GOSUNCN

Welink your smart

DRIVER

Wireless Module Driver User Guide for WinCE

Version: V1.5

Date: 2018-05-03

insight WeLink

Website: www.gosuncnwelink.com

E-mail: welink@gosuncn.com

SAFETY INFORMATION

The following safety precautions must be observed during all phases of the operation, such as usage, service or repair of any cellular terminal or mobile incorporating ME3610 module. Manufacturers of the cellular terminal should send the following safety information to users and operating personnel and to incorporate these guidelines into all manuals supplied with the product. If not so, GOSUNCN does not take on any liability for customer failure to comply with these precautions.

	Full attention must be given to driving at all times in order to reduce the risk of an accident. Using a mobile while driving (even with a hands free kit) cause distraction and can lead to an accident. You must comply with laws and regulations restricting the use of wireless devices while driving.
	Switch off the cellular terminal or mobile before boarding an aircraft. Make sure it switched off. The operation of wireless appliances in an aircraft is forbidden to prevent interference with communication systems. Consult the airline staff about the use of wireless devices on boarding the aircraft, if your device offers a Airplane Mode which must be enabled prior to boarding an aircraft.
	Switch off your wireless device when in hospitals or clinics or other health care facilities. These requests are designed to prevent possible interference with sensitive medical equipment.
	GSM cellular terminals or mobiles operate over radio frequency signal and cellular network and cannot be guaranteed to connect in all conditions, for example no mobile fee or an invalid SIM card. While you are in this condition and need emergent help, please remember using emergency call. In order to make or receive call, the cellular terminal or mobile must be switched on and in a service area with adequate cellular signal strength.
	Your cellular terminal or mobile contains a transmitter and receiver. When it is on, it receives and transmits radio frequency energy. RF interference can occur if it is used close to TV set, radio, computer or other electric equipment.
	In locations with potentially explosive atmospheres, obey all posted signs to turn off wireless devices such as your phone or other cellular terminals. Areas with potentially explosive atmospheres including fuelling areas, below decks on boats, fuel or chemical transfer or storage facilities, areas where the air contains chemicals or particles such as grain, dust or metal powders.

APPLICABILITY TABLE

Product
ZM5202
ZM5202B
MF210V2
MF226
MF206A
ME3732_V2
ZM2210
ZM2110
ZM2111
ZM8620
ME3960
ME3620
ZM5330
MW3650
ME3630

REVISION HISTORY

Version	Date	Description
1.0	2010-09-30	1st release
1.1	2013-03-11	1. Increase reversion history 2. Update version, modify some mistake and format
	2013-04-02	3. Modify the logo of cover and page footer 4. Modify Legal Information
	2013-05-31	5. Release as Version 1.1
1.2	2014-03-07	1. Add the Applicability Table 2. Update the format of this document 3. Update the Legal Information 4. Add the Contact Information 5. Update Chapter 1 of Introduction to Driver Program Package 6. Add three figures in Chapter 2 of How to Integrate Binary Driver Programs 7. Update Chapter 3.1 of Querying Component Information in Current System 8. Add 4 figures in Chapter 6 of 6 Troubleshooting 9. Add Chapter 7 of Appendix: Ports of GOSUNCN modems
1.3	2015-08-12	Add chapter 4 of How to realize power management Add the module of ME3620&MW3650
1.4	2016-07-14	Add the module of ME3630 Update the legal information and format of the doc.
1.5	2018-05-02	Update the format of this document

CONTENTS

1 Special Note

The operation of a driver program needs to be supported by necessary system components. Thus, if the driver program is integrated into the existing WinCE system, it might be the lack of necessary system components that causes the driver to fail to work normally. It's strongly recommended to integrate this driver program when the user is creating the mirror of WinCE system.

2 Introduction to Driver Program Package

When releasing the driver package to customers, we have two releasing modes: binary mode and installation mode. So in the driver package, there are two folders: ***Binary*** and ***Cab***. The ***Binary*** folder contains all driver programs released via the binary mode, while the ***Cab*** folder contains all files released via the installation mode.

2.1 The *Binary* folder

Under *Binary* folder, there are *AP* folder, *Drivers* folder and *REG* folder, see Figure 1-1. Among them, the *AP* folder provides a simple example of an application program under WCDMA mode, including ARM version and x86 version. This application program is named as *ZTESampleUI.exe*, performing the functions of dialup connection, simple short message receiving & sending and simple phonebook. The *Drivers* folder contains the binary driver programs, including ARM version and x86 version. There are two driver programs: *ZTEUSBCOM.dll* and *ZTEUSBManager.dll*. The *REG* folder provides the registry information that the driver program requires. As this file is the same under both ARM and x86 versions, there is only one file named *USBManager.reg*.

Note: Refer to Chapter 2 for how to integrate the driver programs by the files under Binary folder.

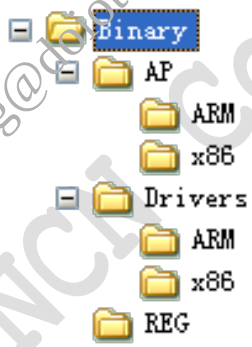

Figure 1-1

2.2 The *Cab* folder

The Cab folder includes two sub-folders: ARM and x86, which store the driver packages in installation mode under the ARM system and the x86 system respectively. See Figure 1-2.

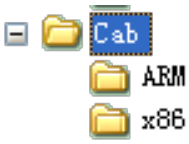

Figure 1-2

Note: Refer to Chapter 3 for how to integrate the driver programs by the Cab package.

3 How to Integrate Binary Driver Programs

This chapter introduces how to manually add the related driver programs under binary folder into WinCE system. When integrating the driver programs, strictly follow the steps below, as the lack of any step might cause the failure of driver programs to work normally.

3.1 Modifying Configuration Files

Notice: There are some differences when perform this step on WinCE5 and WinCE6 system, and the user needs to choose one according to his own system.

3.1.1 Copying Files

Copy "*.reg*" and "*.dll*" files under the *Binary* folder to the related BSP folder in the system, such as the directory "E:\WINCE600\PLATFORM\(TARGETBSP)\FILES\". At the same time, modify *platform.reg* and *platform.bib* under this directory, to add ".reg" and ".dll" files into the final mirror of the operating system. See Figure 2-1.

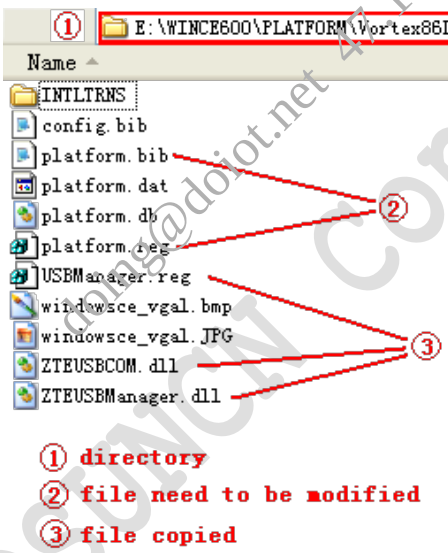

Figure 2-1

3.1.2 Modifying Registry File

In "*platform.reg*", add the content as shown below to the end:

#include "$(_PLATFORMROOT)\(TARGETBSP)\FILES\USBManager.reg"

Note: TARGETBSP is the actual BSP directory. Figure 2-2 shows an example.

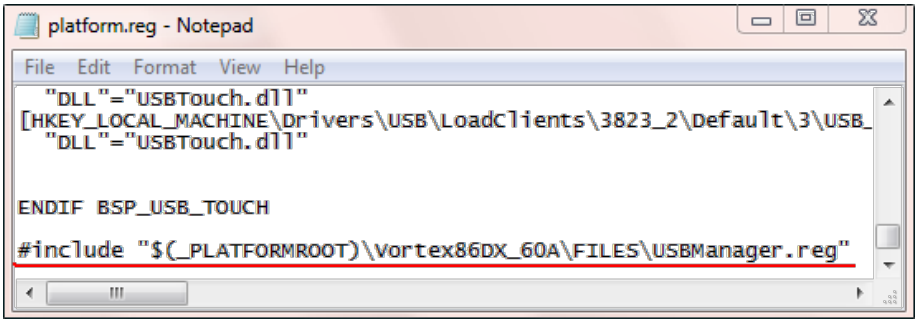

Figure 2-2

3.1.3 Modifying System Configuration Files

Add the following content into "*platform.bib*". As the content is different in WinCE5 and WinCE6, the user can make the corresponding modification according to his own system.

The content must be added into MODULES field, for example, you can add it to the end of the MODULES field, just search FILES field in "*platform.bib*", and then add the following content before it.

1. **Content in CE6 system:**

   ZTEUSBCOM.dll $(_PLATFORMROOT)\(TARGETBSP)\FILES\ZTEUSBCOM.dll NK SHK

   ZTEUSBManager.dll $(_PLATFORMROOT)\(TARGETBSP)\FILES\ZTEUSBManager.dll NK SHK

2. **Content in CE5 system:**

   ZTEUSBCOM.dll $(_PLATFORMROOT)\(TARGETBSP)\FILES\ZTEUSBCOM.dll NK SHC

   ZTEUSBManager.dll $(_PLATFORMROOT)\(TARGETBSP)\FILES\ZTEUSBManager.dll NK SHC

Note: TARGETBSP is the actual BSP directory. Figure 2-3 shows an example in WinCE6 system.

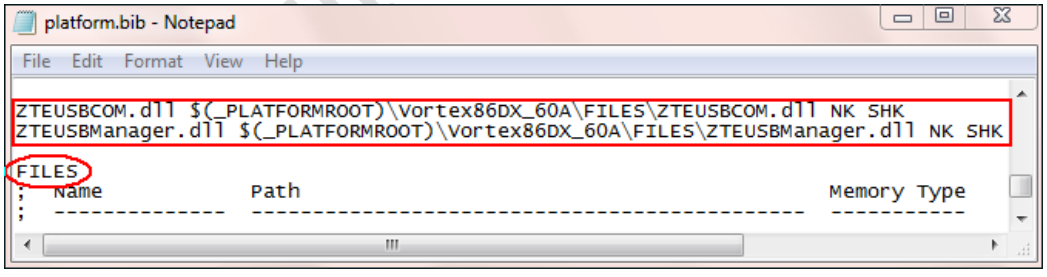

Figure 2-3

3.2 Confirming System Components

As the running of drivers depend on certain related system components, it's necessary to integrate the related system components into the system. The following part is a list of the necessary system components.

PPP Dialup

SYSGEN_PPP=1

SYSGEN_TAPI=1

SYSGEN_PPPOE=1

SYSGEN_ASYNCMAC=1

SYSGEN_MODEM=1

USB Component

SYSGEN_USB=1

SYSGEN_USB_HID=1

SYSGEN_USB_HID_CLIENTS=1

Network-related

SYSGEN_WINSOCK=1

SYSGEN_NDIS=1

SYSGEN_NDISUIO=1

SYSGEN_NETUTILS=1

SYSGEN_AUTORAS=1

SYSGEN_CONNMC=1

IE-related

SYSGEN_IE=1

SYSGEN_IESAMPLE=1

Other Component

SYSGEN_WCELOAD=1

Registry Component

SYSGEN_STATE_NOTIFICATIONS=1 (As this component cannot be added by “catalog view” in VS2005, it can be set via the menu “Project->XXXX Property Pages->Enviroment->NEW”.)

3.3 Rebuilding System & Recreating System Mirror

After adding the system components and configuring the related driver programs, execute the command “*clean sysgen*” to recreate the system mirror.

4 How to Integrate Driver Programs in CAB Package

The driver programs can be integrated by directly installing CAB package in an existing WinCE system. Refer to chapter 3.3 for the installation of the CAB package.

However, as the running of driver programs relies on certain WinCE system components, if the system lacks the necessary components, the driver program will definitely fail to run. So before installation, it is strongly suggested to ensure all the mandatory system components were present. Refer to chapter 3.1 for how to query the system components in your current WinCE system.

Besides, if the system mirror provided by the customer did not have a HIVE-based registry, all the registry information added will be lost after restarting the system. So in such system, each time after the system is restarted, the driver programs can continue to be used only after be re-installed. Refer to chapter 3.2 for how to judge whether the current system mirror has a HIVE-based registry.

4.1 Querying Component Information in Current System

The user can query the system file *ceconfig.h* to acquire the list of components used in the current system. This file is saved under "*windows**ceconfig.h*".

4.1.1 Query WinCE6 System Components

Here is an example of *ceconfig.h* under CE6 system :

```
#ifndef __CECONFIG_H__
#define __CECONFIG_H__
// CEFilter Component List:
#define CE_MODULES_COREDLL 1
#define CE_MODULES_KCOREDLL 1
…
#define CPLMAIN_STYLUS 1
#define CPLMAIN_CERTS 1

// <SYSGENS>
// SYSGEN_ASYNCMAC
```

```
// SYSGEN_ATAPI
…
// SYSGEN_WINSOCK
// SYSGEN_XPSKIN
// </SYSGENS>
#endif
```

As can be seen from the above example file, the last part of the file has many "*SYSGEN_XXXX*" variables. By querying these variables, the user can judge whether the current system includes the mandatory system components for all driver programs. Refer to chapter 2.2 for the list of the mandatory system components.

4.1.2 Query WinCE5 System Components

Here is an example of *ceconfig.h* in CE5 system:

```
#ifndef __CECONFIG_H__
#define __CECONFIG_H__
// CEFilter Component List:
#define CE_MODULES_COREDLL 1
…
#define CE_MODULES_RT_TESTS 1
#define DCOM_MODULES_ATL 1
…
#define DCOM_MODULES_UUID 1
#define GDIEX_MODULES_IMAGING 1
#define IE_MODULES_IEPEERS 1
…
#define IE_MODULES_MLANG 1
#define SCRIPT_MODULES_JSCRIPT 1
#define SERVERS_MODULES_UPNPSVC 1
```

…

#define CPLMAIN_CERTS 1

#endif

In the "*ceconfig.h*" file of WinCE5 system, there were many variables defined to "1" by the line like:

#define XXXX_XXX_XXX 1

By these variables, the user can judge whether the mandatory system components were included in an existing WinCE5 operating system. The following variables must be defined:

Network Dialup-related:

#define CE_MODULES_NDIS 1

#define CE_MODULES_AUTORAS 1

#define CE_MODULES_PPP 1

#define CE_MODULES_TAPI 1

#define CE_MODULES_PPPOE 1

#define CE_MODULES_CONNMC 1

#define CE_MODULES_NDISPWR 1

#define CE_MODULES_NETUI 1

#define CE_MODULES_ASYNCMAC 1

#define CE_MODULES_UNIMODEM 1

#define CE_MODULES_TCPSTK 1

#define CE_MODULES_WINSOCK 1

USB-related:

#define CE_MODULES_USBHOST 1

#define CE_MODULES_USBD 1

IE-related:

#define IE_MODULES_IESAMPLE 1

Others:

#define DATASYNC_MODULES_WCELOAD 1

4.2 Judging Whether the Registry is HIVE-based

For WinCE6 system, the registry type used by the current system can be confirmed by directly querying the "*SYSGEN_XXXX*" variables in file "*ceconfig.h*". If the following variable could be found, then the registry of current WinCE6 system is HIVE-based.

SYSGEN_FSREGHIVE

For WinCE5 system, the registry type can also be judged by the variables defined in "*ceconfig.h*", if the following variable was defined to "1", then the registry of current WinCE5 system is HIVE-based.

FILESYS_FSREGHIVE

4.3 Installing & Using Driver Programs

For the driver installation package, directly double-click it to install. After the driver programs are installed completely, the installation package will be deleted automatically. During the installation process, the driver programs are imported to the *Windows* directory, the registry information related to drivers is imported to the registry file, and if there is UI file, the UI file is imported to the "*program file*" directory.

For WCDMA or GSM, the user can use the attached sample UI to dial up, send short messages or access the phone book. Also user can manually create a dial up connection. But for EVDO, the user can only manually create the dial up connection to test the networking function. Refer to chapter 4 for how to manually create a dial up connection.

5 How to realize power management

5.1 Modify the code of Host Controller Driver (HCD)

5.1.1 Overview

There are three kinds of HCD named OHCI/UHCI/EHCI in WinCE5. OHCI/UHCI is the realization of USB bus specification V1.1 and EHCI is the realization of USB bus specification V2.0. In order to realize the USB suspend and remote wakeup, you need to modify the code of HCD. We take the standard HCD code in WinCE5.0 for example (If the users of module have you own HCD code, the changes are similar, only may be difference is the path of files.)

You need to changes the files list below for OHCI/UHCI:

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\COMMON\cdevice.cpp

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\COMMON\cdevice.hpp

You need to changes the files list below for EHCI:

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\USB20\USB2COM\ cdevice.cpp

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\USB20\USB2COM\ cdevice.hpp

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\USB20\EHCI\chw.cpp

Note: we need to modify cdevice.hpp and cdevice.cpp for both EHCI and OHCI/UHCI. But when the protocol is EHCI, the files path is difference from OHCI/UHCI. If you don't know which OCHI/UHCI or EHCI is used, you can modify all the files list above as the follows.

5.1.2 Modify the files for OHCI/UHCI

5.1.2.1 Modify cdevice.hpp

Find the line as below:

virtual BOOL ResumeNotification() { return FALSE; };

This is the declaration and definition for the member function of CDevice, modify it as shown in the follows:

virtual BOOL ResumeNotification();

5.1.2.2 Modify cdevice.cpp

Add the content as below in the end of file:

```
void TriggerResumeEvent_ZTE(TCHAR *eventName)
{
```

```
    HANDLE hRemoteWakeUpEvent = CreateEvent(NULL, FALSE, FALSE, eventName);
    if (GetLastError() == ERROR_ALREADY_EXISTS)
    {
        SetEvent(hRemoteWakeUpEvent);
    }
    CloseHandle(hRemoteWakeUpEvent);

}
BOOL CDevice::ResumeNotification()
{
    TriggerResumeEvent_ZTE(TEXT("ZTE_MF627_DEVICE_RWAKEUP_EV\0"));
    return FALSE;
}
```

5.1.3 Modify the files for EHCI

5.1.3.1 Modify cdevice.hpp

Find the line as below:

virtual BOOL ResumeNotification() { return FALSE; };

This is the declaration and definition for the member function (ResumeNotification) of CDevice, modify it as shown in the follows:

virtual BOOL ResumeNotification();

5.1.3.2 Modify cdevice.cpp

Add the content as below in the end of file:

```
void TriggerResumeEvent_ZTE(TCHAR *eventName)
{
    HANDLE hRemoteWakeUpEvent = CreateEvent(NULL, FALSE, FALSE, eventName);
    if (GetLastError() == ERROR_ALREADY_EXISTS)
    {
        SetEvent(hRemoteWakeUpEvent);
    }
    CloseHandle(hRemoteWakeUpEvent);

}
BOOL CDevice::ResumeNotification()
```

```
{
    TriggerResumeEvent_ZTE(TEXT("ZTE_MF627_DEVICE_RWAKEUP_EV\0"));
    return FALSE;
}
```

Modify CRootHub::WaitForPortStatusChange function

Find the function named CRootHub::WaitForPortStatusChange, and define a global variable in front of the function:

BOOL g_AutoResume = FALSE;

Then find the code as the shown below, add the lines in the red square to the function of CRootHub::WaitForPortStatusChange

```
for ( UCHAR port = 1; !fSuccess && port <= m_usbHubDescriptor.bNumberOfPorts; port++ ) {
  if (m_pCHcd- >DidPortStatusChange( port )) {
    // port status changed on this port
    rPort = port;
    // we could just call CHW::GetPortStatus, but it is
    // better to call CRootHub::GetStatus which is
    // designed to implement the virtual CHub::GetStatus function.
    fSuccess = GetStatus( port, rStatus );
    if( (rStatus.status.port.PortSuspended == 0) && (g_AutoResume == TRUE)){
      rStatus.change.port.SuspendChange = 1;
      g_AutoResume = FALSE;
    }
  }
}
```

The content in red square need be added.

5.1.3.3 Modify chw.cpp

Modify CHW::RootHubFeature function

Find the line as below in the function of CHW::RootHubFeature:

portSC.bit.ForcePortResume=0;

add the lines in the red square as shown below:

```
switch (feature) {
 case USB_HUB_FEATURE_PORT_ENABLE:        portSC.bit.Enabled=0; break;
 case USB_HUB_FEATURE_PORT_SUSPEND:        // EHCI 2.3.9
  if (portSC.bit.Suspend !=0 ) {
    portSC.bit.ForcePortResume=1;
    Write_PORTSC( port, portSC );
    Sleep(20);
    portSC.bit.ForcePortResume=0;
    portSC.bit.Suspend=0;
  }
  break;
```

红框为新加的内容

The content in red square need be added.

Modify CHW::DidPortStatusChange function

find the code as shown below, add the lines in the red square to this function :

```
extern BOOL g_AutoResume;
BOOL CHW::DidPortStatusChange( IN const UCHAR port )
//
// Purpose: Determine whether the status of root hub port # "port" changed
//
// Parameters: port -  0 for the hub itself, otherwise the hub port number
//
// Returns: TRUE if status changed, else FALSE
//
// Notes:
// ************************************************************************
{
   USB_HUB_AND_PORT_STATUS s;
   CHW::GetPortStatus(port, s);
   if(s.change.port.SuspendChange){
     g_AutoResume = TRUE;
   }
   return s.change.word ? TRUE : FALSE;
}
```

The content in red square need be added.

两处红框中为新增内容

5.2 Modify the code of USBD

5.2.1 File list

\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\USB\USBD\usbddrv.cpp

We take the standard USBD code in WinCE5.0 for example (If the users of module have you own USBD code, the changes are similar, only may be difference is the path of files.)

5.2.2 Modify usbddrv.cpp

Find the function named HcdDeviceAttached, and locate the red square as shown in the figure below:

```
LPCUSB_INTERFACE pInterface =pDev->pDeviceInfo->lpActiveConfig->lpInterfaces;
for (iIndex =0 ; iIndex < pDev->pDeviceInfo->lpActiveConfig->Descriptor.bNumInterfaces && iIndex < gcMaxI
   pDev->rgbInterfaceIndex[iIndex] = pInterface->Descriptor.iInterface;
   ASSERT(pInterface->Descriptor.iInterface != INVALID_INTERFACE_INDEX);
   pInterface ++;
}
while (fRet && !fLoaded) {
  // Attempt to load client driver based on registry settings
  fRet = LoadDeviceDrivers(pDev, &fLoaded);
  if (fRet && !fLoaded) {
```

红框中为错误代码

Note: Replace the code in red square with the line below:

pDev->rgbInterfaceIndex[iIndex] = pInterface->Descriptor.bInterfaceNumber;

5.3 Rebuild the system

After all the modifications above has finished, rebuild the system.

6 How to Establish System Dialup

In the current WinCE system, there is no program officially developed for dialing up , but only a simple sample dialing up program in WCDMA mode. If do not want to use the sample program, users can manually create the dialing up connection.

For EVDO mode, GSM mode and WCDMA mode, there is a difference between dialing up account and APN. The following section introduces EVDO mode and WCDMA mode respectively. For GSM mode, refer to the configuration of WCDMA mode, and the only difference is APN.

6.1 How to Make Dialup in EVDO System

1. Enter the control panel, and double-click the "***Network and Dial-up Connections***" icon.

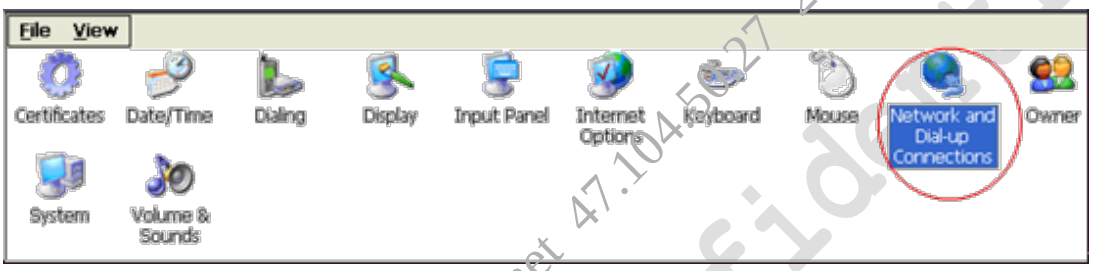

Figure 4-1

2. Double-click the "***Make New Connection***" icon.

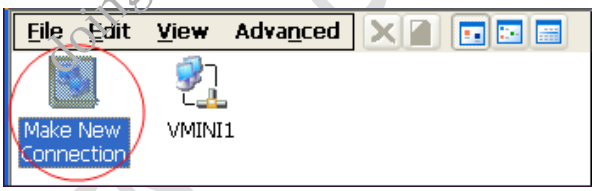

Figure 4-2

3. Click the "***Next***" button.

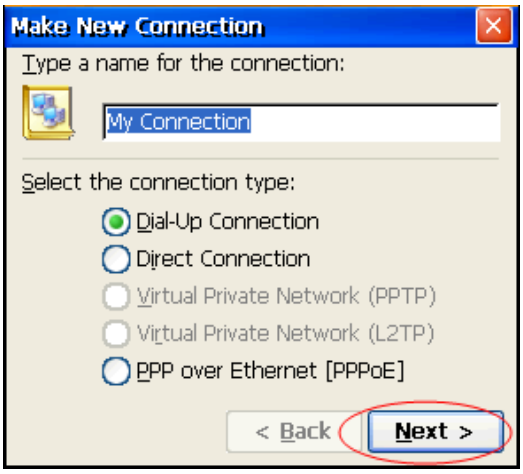

Figure 4-3

4. Click the drop-down list.

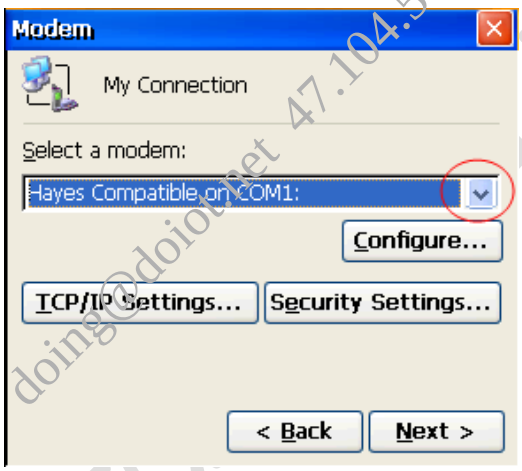

Figure 4-4

5. Choose "***ZTE MF627 Modem***", as shown below

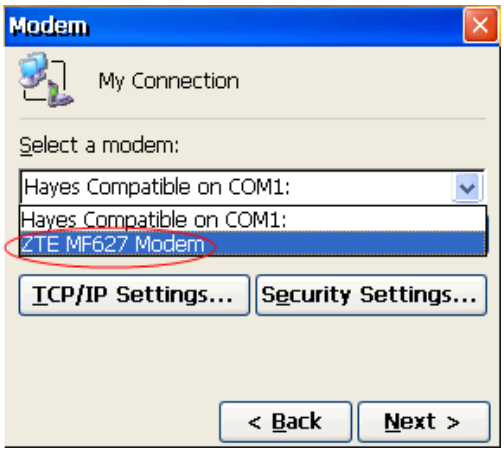

Figure 4-5

6. Click the "***Next***" button.

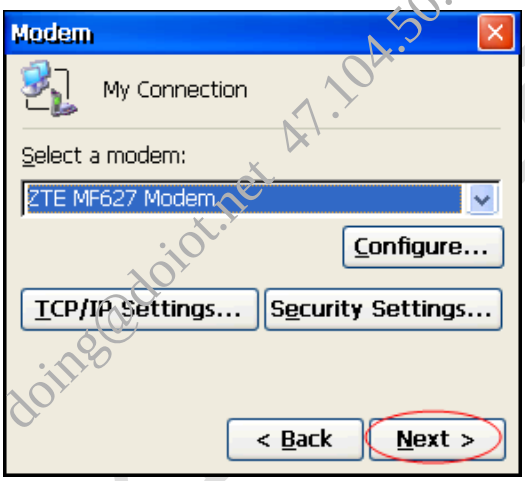

Figure 4-6

7. In the input box "***Phone number***", input the dialup command "#777", and then click the "***Finish***" button.

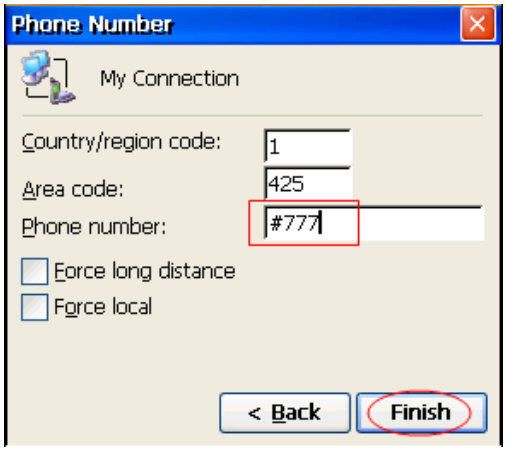

Figure 4-7

8. After completing the above steps, the shortcut of "***My Connection***" dialup connection is created.

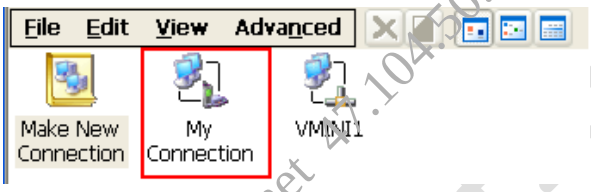

Figure 4-8

9. Double-click the "***My Connection***" icon in Figure 4-8, to enter the dialup interface as shown below.

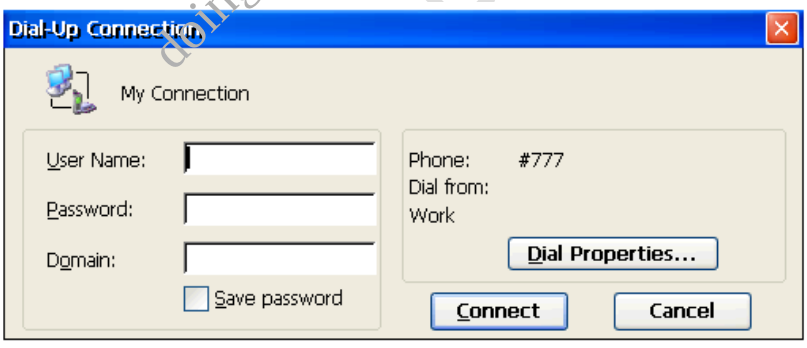

Figure 4-9

10. In the "***User Name***" input box, input the account (such as "card"; in the "***Password***" input box, input the password. The "***Save Password***" option can be checked or not. Click the "***Connect***" button to create the dialup connection.

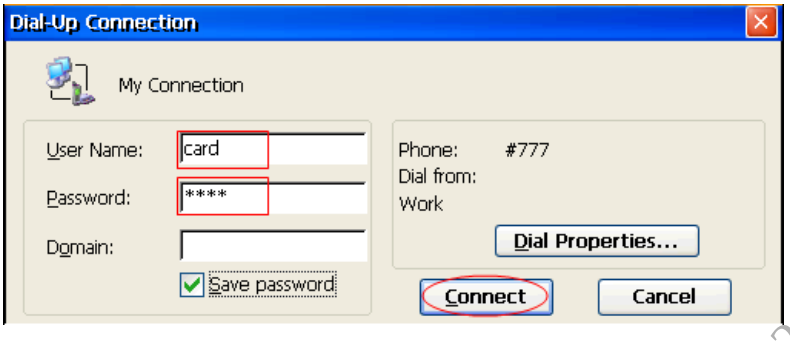

Figure 4-10

11. When creating the dialup connection, it first enters the "Opening Port" phase, as shown below.

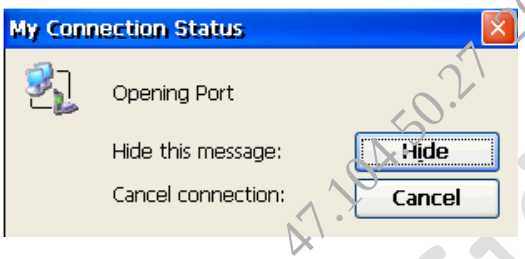

Figure 4-11

12. Then enter the dialup phase, as shown below.

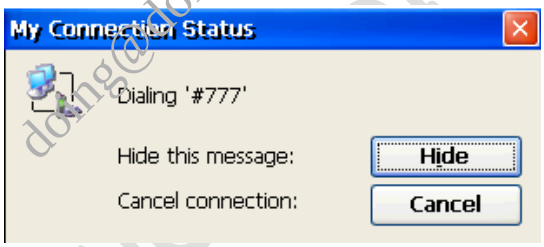

Figure 4-12

13. If the dialup is made successfully, the interface of successful connection appears, as shown below.

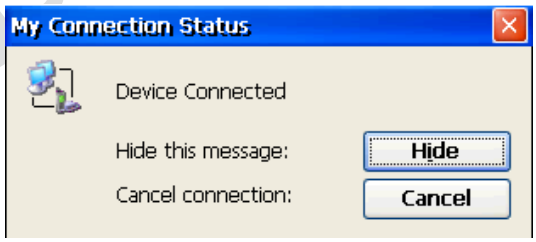

Figure 4-13

14. Due to the environment signal, the following interface of dialup failure might appear. It's recommended to attempt the dialup for several times.

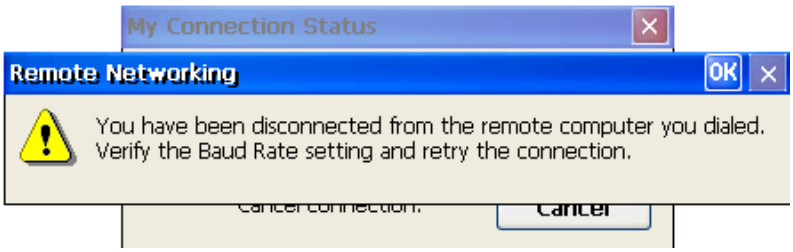

Figure 4-14

6.2 How to Make Dialup in WCDMA (GSM) System

The dialup connection in WCDMA mode can be established according to steps below. The only difference between WCDMA mode and GSM mode is the setting of APN. The detailed steps are as below.

15. Enter the control panel, double-click the "***Network and Dial-up Connections***" icon.

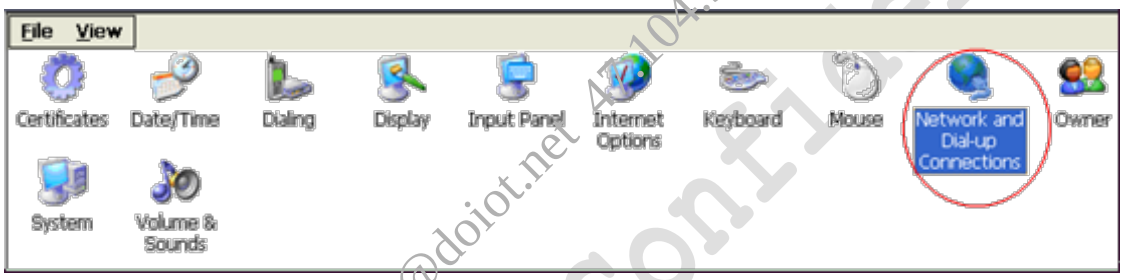

Figure 4-15

16. Double-click the "***Make New Connection***" icon.

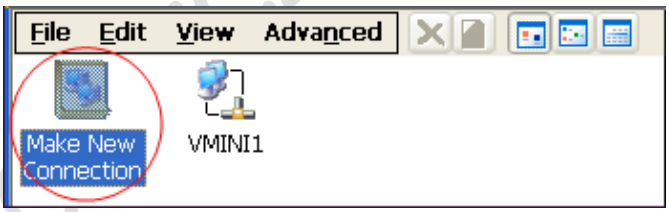

Figure 4-16

17. Click the "***Next***" button.

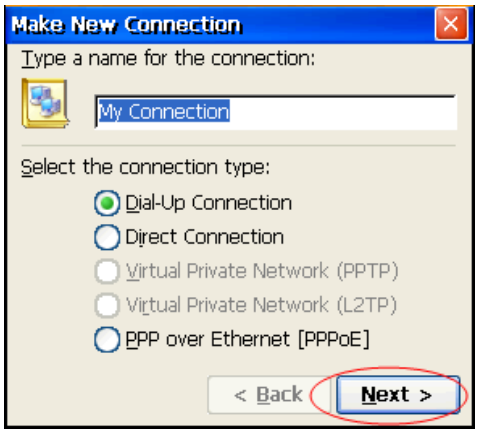

Figure 4-17

18. Click the drop-down list.

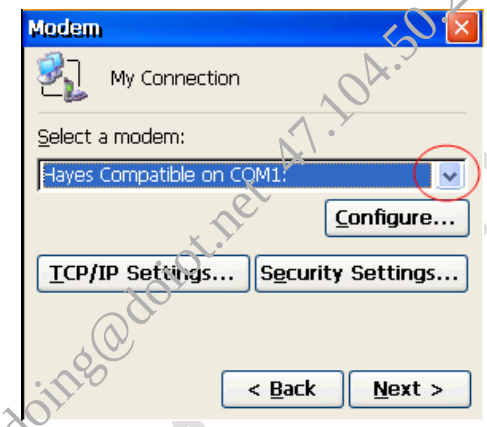

Figure 4-18

19. Choose "***ZTE MF627 Modem***", as shown below.

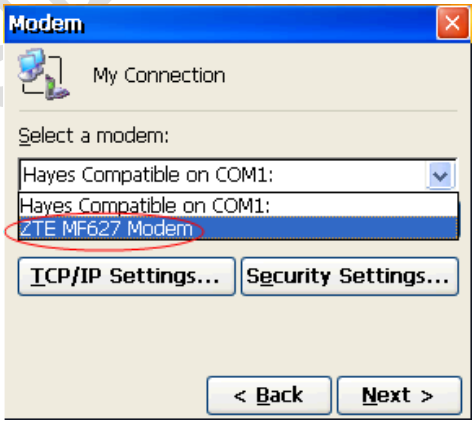

Figure 4-19

20. Click the "***Configure***" button.

Figure 4-20

21. Then choose the "***Call Options***" tab to enter the following interface.

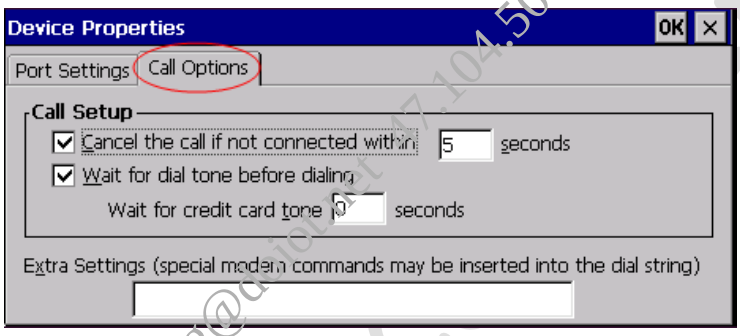

Figure 4-21

22. In "***Extra Settings***" edit box, input the following dialup command. As shown below, "CMNET" indicates the dialup APN, which is modified according to the local APN. After the information is input completely, click ***OK*** to return.

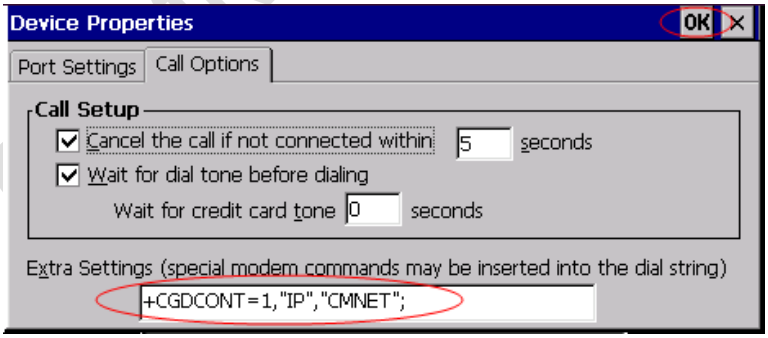

Figure 4-22

23. Click "***Next***" button.

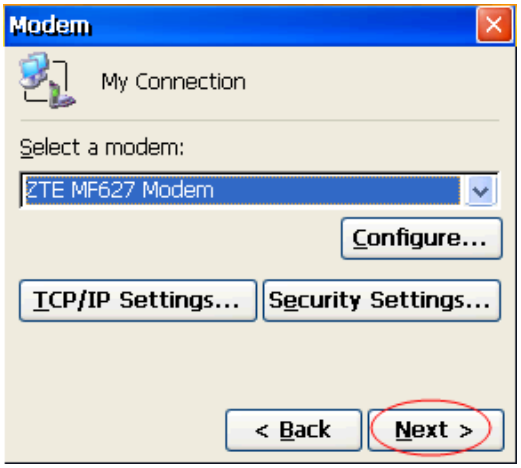

Figure 4-23

24. In the input box "***Phone number***", input the dialup command "***99#**", and then click the "***Finish***" button.

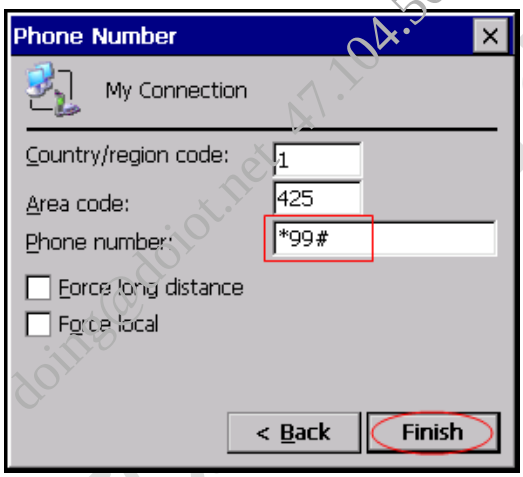

Figure 4-24

25. After completing the above steps, the shortcut of ***My Connection*** dialup connection is created.

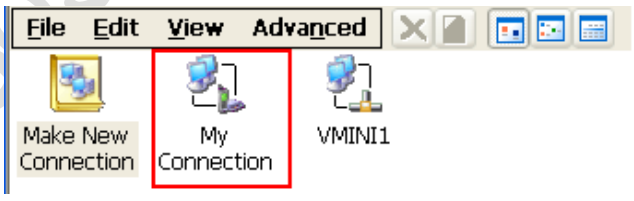

Figure 4-25

26. Double-click the "***My Connection***" icon in Figure 4-25, to enter the dialup interface as shown below. Then click the "***Connect***" button.

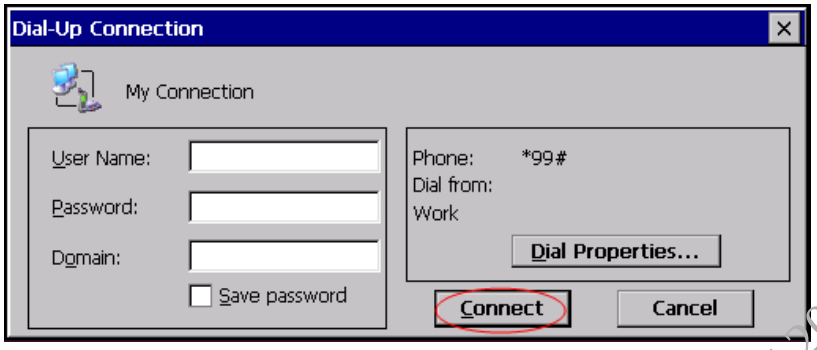

Figure 4-26

27. When creating the dialup connection, it first enters the "Opening Port" phase, as shown below.

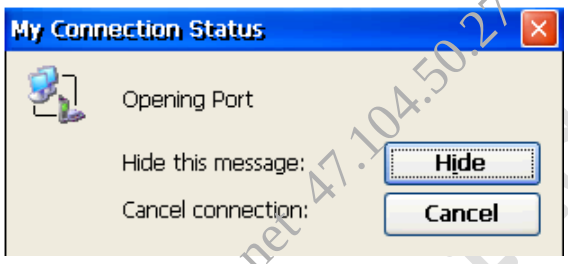

Figure 4-27

28. Then enter the dialup phase, as shown below.

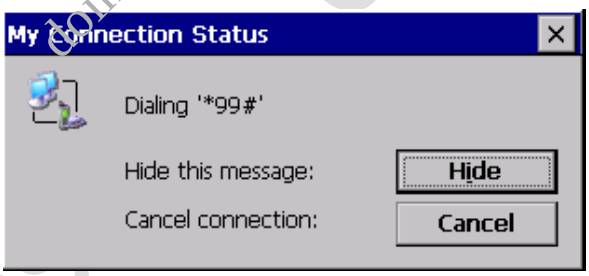

Figure 4-28

29. If the dialup is made successfully, the interface of successful connection appears, as shown below.

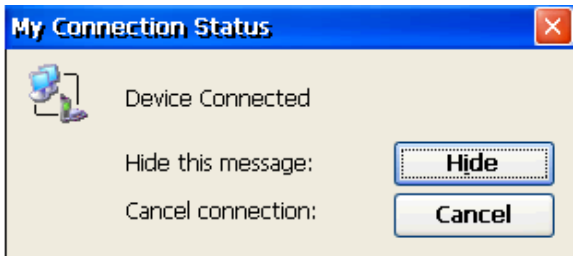

Figure 4-29

30. Due to the environment signal, the following interface of dialup failure might appear. It's recommended to attempt the dialup for several times.

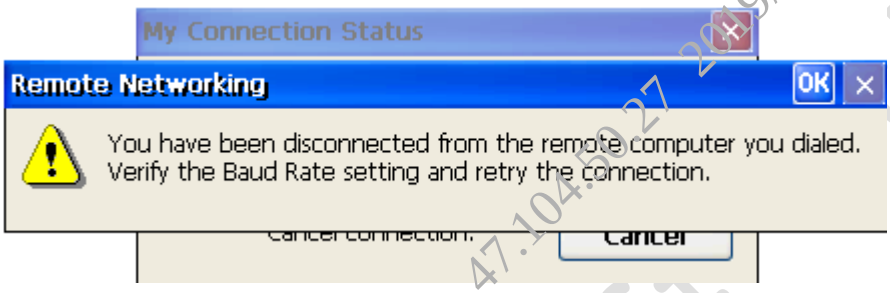

Figure 4-30

7 Description of Related Registry Key

7.1 Function Switches

These switches are related to power management, including:

- The selective suspending function switch (*SuspendEnable*)
- The maximum idle time before suspending(*SuspendTime*)
- The automatic wakeup interval of the equipment(*AutoWakeTime*).

The selective suspending function switch (*SuspendEnable*) indicates whether the selective suspending function is enabled. If enabled, it means that when the idle time of the equipment is longer than the maximum idle time (*SuspendTime*), the system will automatically enter the power-saving mode. In the power-saving mode, if there is any new equipment operation from the host, the equipment will automatically wake up from the power-saving mode and enter the normal working mode to work. If there is no operation after another period, the equipment enters the power-saving mode again.

The selective suspending function is developed to save power. In the desktop system, after the system has enabled the selective suspending function, if there is a remote call or short message, the equipment will be waken up and enter the normal working mode. However, there are some bugs in the realization of USB stack in WinCE system, so even the equipment has enabled the selective suspending function, it still cannot be waken up by the remote call or short message. With our driver programs, this problem can be fixed by the following two methods:

1. Modify the code of WinCE system. Refer to the document “Enable remote wakeup function in WinCE or Windows Mobile.pdf” for the detailed methods.
2. Add the registry key for automatic driver wakeup (*AutoWakeTime*). After adding this registry key, if the driver has enabled the selective suspending function, the system will automatically wake up after a certain period to detect whether there is any waiting short message.

7.1.1 SuspendEnable

Location:

[HKEY_LOCAL_MACHINE\SOFTWARE\ZTE\Setting]

Default Value:

1. By default, this key is initialized as FALSE.

2. If the user has enabled the selective suspending function, this key is set as TRUE.

Special note: To enable the selective suspending function, this key should be set as TRUE, and the value of *SuspendTime* should be set as well. Refer to *SuspendTime* for related explanations.

Note: The key SuspendEnable is the switch for the selective suspending function. By default, the system does not support the selective suspending function.

If *SuspendEnable* is set to TRUE and the value of *SuspendTime* is set as well, the system checks the system running status. If there is no data processed during the period of *SuspendTime*, the equipment automatically enters the suspending status. Under this status, if there is data need processing, the equipment is automatically waken up to process the data.

7.1.2 SuspendTime

Location:

[HKEY_LOCAL_MACHINE\SOFTWARE\ZTE\Setting]

Default Value:

The default value of this key is **48** in the unit of second.

Note: This key is used together with SuspendEnable. If SuspendEnable is set to FALSE, the selective suspending function is not enabled, so this key will not take effect. This key takes effect only when SuspendEnable is TRUE.

7.1.3 AutoWakeTime

Location:

[HKEY_LOCAL_MACHINE\SOFTWARE\ZTE\Setting]

Default Value:

1. By default, this key is not added into the system.

2. If the system needs to support the automatic wakeup function, this key is added and input with the interval of automatic wakeup, in the unit of second.

Note: As the WinCE system does not support the remote wakeup function, normally, if the USB equipment enters the suspending status, short messages will not be received in time, and can only be received after the equipment wakes up.

This is a bug of WinCE system itself, but can be fixed by modifying the related system codes. If the system codes cannot be modified, users can add this key to solve this problem. After adding this key, if the driver enters the suspending status, the system will automatically wake up after the set period to check whether there are waiting short messages.

Special note: The system performance is affected by adding this key, so it is suggested to modify the related system codes to solve the remote wakeup problem. For detailed, refer to document "Enable remote wakeup function in WinCE or Windows Mobile.pdf".

7.2 Debugging Switches

7.2.1 MdmLogFile

Location:

[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\ZTEUSBMODEM_CLASS\Unimodem\Settings]

Default Value:

By default, the value of this key is **0**. Set this key to **1** to display the log information of dialup.

Note:

1. This key determines whether to output the related LOG information for the system dialup commands.

2. If there is any problem with establishing the dialup connection by the driver, set this key to 1 to output the log information. The information is saved in the system root directory with the name of "mdmlog*.txt", where symbol "*" indicates the number.

7.2.2 RetailZone

Location:

[HKEY_LOCAL_MACHINE\SOFTWARE\ZTE\Setting]

Default Value:

By default, the value of this key is **0**. If there is any problem with the driver, open this log switch to output the information of driver usage.

Note: This registry key decides whether to output the debugging information during the running process of the driver. By default, the debugging information is not generated.

If there is any problem with the driver, the user can change the key value to "***0x80000000***" to output the log information of the driver. The log file is saved in the system root directory with the name of "***USBLog.txt***"

8 Troubleshooting

8.1 Special Note

When integrating the driver programs, it is mandatory to strictly follow the driver integration methods. If any step is missing, the loading of driver will probably fail!

8.2 Driver Loading Phase

8.2.1 Problem 1

Problem Description:

After the equipment is accesses, the system prompts to input the driver name. As shown in Figure 6-1.

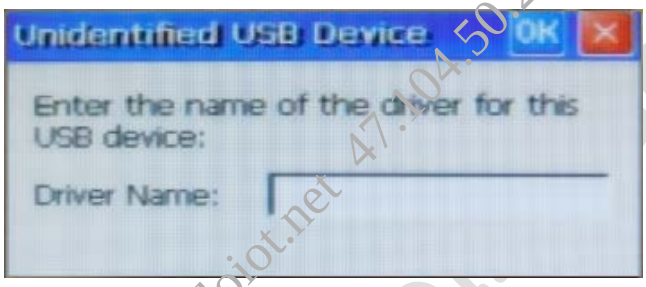

Figure 6-1

Problem Analysis:

This problem occurs because the inserted equipment has not installed the driver program.

As WinCE system loads the equipment via the equipment PID, if the PID of the accessed equipment is not in the list of supported PID, the above problem occurs when inserting the equipment.

Solution:

1. Make sure that the ".dll" files have all been correctly loaded into the system mirror.

There are two driver files: ***ZTEUSBCOM.dll*** and ***ZTEUSBManager.dll***. After installing the driver, the files are saved under the "***Windows directory***". By default, they are hidden files.

2. Via the other system, query the PID of the current equipment via the following methods:

Insert the equipment into a computer with Windows system. In Device Manager -> Properties -> Details-> Hardware IDs, check the field in Figure 6-2, "0117" is the equipment PID in hexadecimal format.

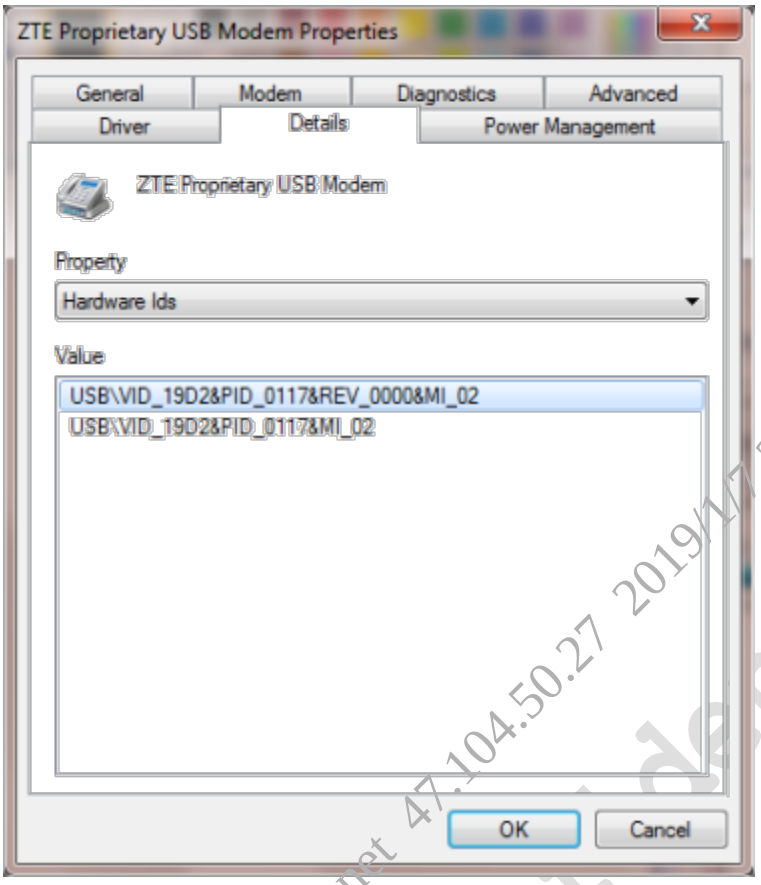

Figure 6-2

3. Make sure your WinCE system supports the PID of current equipment

(1) In the *USBManager.reg* file of the driver released package, check "PID_XXX" variables. The hexadecimal format integers in these variables are the supported equipment PIDs . Make sure that the equipment PID exists. See Figure 6-3.

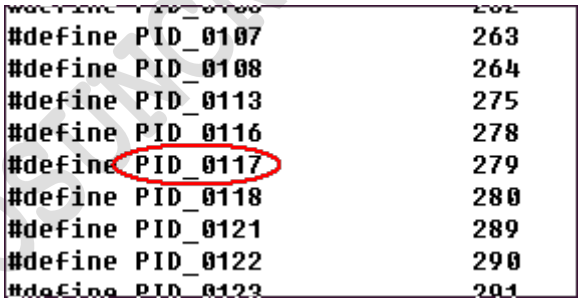

Figure 6-3

(2) In WinCE system, open the registry and find the following location:

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients]

There were many keys with the name like "*6610_XXXX*", the decimal format integers in these variables were the PIDs supported by the current driver, Make sure that the equipment PID exists. See Figure 6-4.

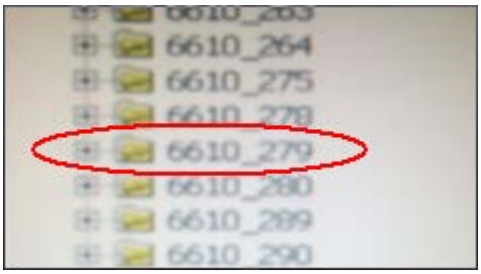

Figure 6-4

4. As long as the current equipment PID exists among the supported PID list of WinCE system, the above problem will not occur when accessing the equipment.

8.2.2 Problem 2

Problem Description:

After inserting the equipment, the driver loading fails, and the UI interface displays that no equipment is inserted (the equipment connection icon displays no connected equipment).

Problem Analysis:

There are many causes that might lead to the failure of driver loading:

Cause 1: When the equipment is inserted into the system, it is reported as CD-ROM equipment with PID as 0x2000, and the CD-ROM driver of WinCE system can identify this equipment. So it is identified as the CD-ROM driver.

Cause 2: Check whether the value of related sysgen key (SYSGEN_STATE_NOTIFICATIONS) has been added when making the mirror of WinCE system. If this key is not added, the related registry operations cannot be realized, and the equipment driver quits the loading process due to the failure of registry operations.

Solution:

For Cause 1:

Disable the "*autorun*" function via the steps below:

By using the super terminal in Windows XP system, open the AT interface of data card, input the "AT+ZCDRUN=8" command.

By the "AT+ZCDRUN=4" command, query whether the autorun function has been disabled.

For Cause 2:

On WinCE6 system: As this component cannot be added via catalog view of VS2005, set the value via the menu Project->XXXX Property Pages->Environment->NEW.

On WinCE5 system: As this component cannot be added via catalog view of PB5, set the value via the menu Platform->Settings->Environment->NEW.

8.2.3 Problem 3

Problem Description:

When the loading of driver fails, by opening the driver log information, it is found out that COM_Deinit is directly invoked after loading the driver.

Problem Analysis:

1. Open the log switch to display the driver loading process, and check which step fails during the process.

The location of log switch is as below:

[HKEY_LOCAL_MACHINE\SOFTWARE\ZTE\Setting]

RetailZone=0x80000000

The default value of the above key is 0. Set its value to 0x80000000 to open the log switch. The log is saved as "***USBLog.txt***" under the root directory.

2. The printed log information for the above problem is as below:

271406 PID:e3bed026 TID:e3b8072e [2]/[3]:

271412 PID:e3bed026 TID:e3b8072e interface[2] buffer=ZTEUSBMODEM_CLASS

271421 PID:e3bed026 TID:e3b8072e ZTEUSBModemDevice::Attach

271428 PID:e3bed026 TID:e3b8072e ZTEUSBModemDevice::Attach(InterfaceOnly): Found Supported Interface (bInterfaceNumber=2,bAlternateSetting=0

271444 PID:e3bed026 TID:e3b8072e +COM_Init

271450 PID:e3bed026 TID:e3b8072e Try to open Drivers\Active\36

271458 PID:e3bed026 TID:e3b8072e DevIndex 0

271464 PID:e3bed026 TID:e3b8072e USBSer: Retail Trace Setting: 0x0

271479 PID:e3bed026 TID:e3b8072e +COM_Deinit

271485 PID:e3bed026 TID:e3b8072e interface 0, flag is 0x1

3. The final error cause is:

TDQueue::TDQueue - no memory for TD list (line285, D:\WINCE600\PUBLIC\COMMON\OAK\DRIVERS\USB\HCD\OHCD2\Transfer.cpp)

Currently, code 6410 adopts the default OHCI code of Microsoft. The memory size ***PhysicalPageSize*** is set as below:

The default size is 64K.

static const DWORD gcTotalAvailablePhysicalMemory = 64*1024; // 64K

Solution:

Set the value (gcTotalAvailablePhysicalMemory) to be greater, so as to support more USB equipment.

At present, if the codes are not modified, the value of this key can modified in the registry (PhysicalPageSize under the item OHCI driver).

8.3 Dialup Interconnection Phase

8.3.1 Problem 1

Problem Description:

The dialup program cannot run on the target equipment.

Problem Analysis:

1. The version of WinCE system does not match that of the dialup programs.
2. The CPU type of the target equipment does not match the version of the dialup programs.

Solution:

1. Check whether it's caused by the mismatch of WinCE version or CPU type.
2. Find out the corresponding WinCE version and SDK of the corresponding CPU type, and re-compile the dialup programs.

8.3.2 Problem 2

Problem Description:

After the dialup UI clicks Connect, it directly returns failure. During the dialup process, the dialup UI does not pass the authentication phase (User Authentication).

Problem Analysis:

Open the registry and find the following key:

[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\ZTEUSBMODEM_CLASS\Unimodem\Settings]

"MdmLogFile"=dword:1

Check the dialup operation log "mdmlog6.txt" (the number is any number generated by the system in sequence) under the root directory. The correct dialup command should be: ATDT + dialup command in each mode (For example, ATDT*98*1# for TD mode). If the dialup command in log file is not the same as above, the failure of dialup occurs.

Solution:

Generally, the wrong dialup command is caused by the wrong information of system dialup. The correct settings should be:

[HKEY_CURRENT_USER\ControlPanel\Dial\Locations]

"0"=multi_sz:"Work","*99#","9,1FG","9,011,EFG","425","","1","0","","",""

"1"=multi_sz:"Home","*99#","9,1FG","9,011,EFG","425","","1","0","","",""

If the registry information is not the same as them, make related modifications.

8.3.3 Problem 3

Problem Description:

When manually creating the WCDMA dialup mode to dial up, the dialup operation fails, and no error prompt appears.

Problem Analysis:

As APN needs to be set during the dialup process of WCDMA, the user will fail to dial up if the APN has not been set. In this case, there is no error prompt as seen from the log file of modem dialup.

Solution:

During the process of establishing the dialup connection, as shown in Figure 4-22, manually add the following commands to the AT command sequence:

+CGDCONT=1,"IP","APN";

Among the above, APN is the specific APN command of the customer. Refer to the section of "How to Manually Establish WCDMA Dialup Connection" for details.

8.4 Other Problems

8.4.1 Problem 1

Problem Description:

After loading the equipment, the com port number cannot be specified.

Problem Analysis:

No port number of the equipment has been specified in the previous drivers, as the port number was automatically allocated by the system when loading the equipment.

Solution:

Under the following location in registry, add the related key of Index, and specify it to the corresponding com port No.

[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\ZTEUSBMODEM_CLASS]

"Index"=dword:X (X is the com port number that needs to be generated.)

9 Appendix: Ports of GOSUNCN modems

After the driver has been installed, if a GOSUNCN modem is connected, several ports will appear in the operating system. Refer to the following table for more details.

Note: PID in the first column is the product ID of your modem.

Modules	PID	Interface number	Port
MF206A MG3732_V2/B &V2/C ZM5202E	0x0117	0	Diag
		1	NMEA
		2	Modem
ZM5330	0x1432 (NDIS) 0x1433(ECM) for Linux	0	Diag
		1	AT
		2	Modem
		3	NDIS(ECM)
		4	ADB
ZM5202	0x0144	0	Diag
		1	GPS
		2	NMEA
		3	NMEAext
		4	Modem
	0x0117	0	Diag
		1	NMEA
		2	Modem
ZM2210	0x0152	0	Modem
		1	Diag
		2	AT
		3	Vousb
		4	NMEA
ZM8620	0x0396	0	Diag
		1	AT
		2	Modem
		3	NDIS(ECM)
		4	ADB
MW3650 MG3732_V2/A	0xFFEB	0	Diag
		1	VoUSB
		2	Service/AT
		3	Modem
MC8635 MC8630	0xFFEE	0	Modem
		1	Service/AT
		2	Vousb
		3	Diag
MC2716	0xFFED	0	Modem
		1	Service/AT
		2	Vousb
		3	Diag
MC8332 MC8618	0xFFEA	0	Diag
		1	Modem

Modules	PID	Interface number	Port
ME3620_C1A	0x1432(NDIS)	0	Diag
	0x1433(ECM)	1	AT
	0x1476(ECM)	2	Modem
		3	NDIS(ECM)
		4	ADB
MF226	0x0144(PRC, Japan)	0	Diag
		1	GPS
		2	AT
		3	ATExt
		4	Modem
	0x0117(South Korea)	0	Diag
		1	NMEA
		2	Modem
	0x2003(Malaysia)	0	Diag
		1	NMEA
		2	MSUSB
		3	Modem
ME3630	0x1432	0	DIAG
		1	AT
		2	MODEM
		3	NDIS
		4	ADB
	0x1476	0	DIAG
		1	AT
		2	MODEM
		3	ECM
		4	ECM
		5	ADB
	0x1509	0	DIAG
		1	AT
		2	MODEM
		3	RESERVED
		4	ECM
		5	ECM