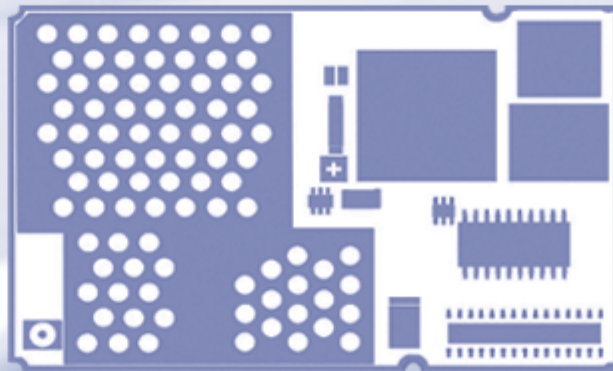


SIEMENS



Integrating USB into GSM Applications

Siemens Cellular Engine

Version: 02
DocId: WM_AN32_USB_v02
Products: MC75, TC65, TC63, AC65, AC75, XT65, XT75

Application Note 32

Application Note 32: **Integrating USB into GSM Applications**Version: **02**Date: **2006-9-29**DocId: **WM_AN32_USB_v02**Status **Confidential / Released**Supported Products: **MC75, TC65, TC63, AC65, AC75, XT65, XT75****General Notes**

Product is deemed accepted by recipient and is provided without interface to recipient's products. The documentation and/or product are provided for testing, evaluation, integration and information purposes. The documentation and/or product are provided on an "as is" basis only and may contain deficiencies or inadequacies. The documentation and/or product are provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Siemens further disclaims all warranties, including without limitation any implied warranties of merchantability, completeness, fitness for a particular purpose and non-infringement of third-party rights. The entire risk arising out of the use or performance of the product and documentation remains with recipient. This product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results. Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones must be followed. Siemens or its suppliers shall, regardless of any legal theory upon which the claim is based, not be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the documentation and/or product, even if Siemens has been advised of the possibility of such damages. The foregoing limitations of liability shall not apply in case of mandatory liability, e.g. under the German Product Liability Act, in case of intent, gross negligence, injury of life, body or health, or breach of a condition which goes to the root of the contract. However, claims for damages arising from a breach of a condition, which goes to the root of the contract, shall be limited to the foreseeable damage, which is intrinsic to the contract, unless caused by intent or gross negligence or based on liability for injury of life, body or health. The above provision does not imply a change on the burden of proof to the detriment of the recipient. Subject to change without notice at any time. The interpretation of this general note shall be governed and construed according to German law without reference to any other substantive law.

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 created by patent grant or registration of a utility model or design patent are reserved.

Copyright © Siemens AG 2006

Contents

0	Document History	6
1	Introduction	7
1.1	Supported Products.....	7
1.2	Related Documents.....	7
1.3	Abbreviations.....	8
2	General Features of the USB interface	9
2.1	Technical Specification.....	9
2.2	USB Modem Profile.....	9
2.3	USB Interface Connections.....	9
3	USB Modem in Windows XP/2000 Environment	10
3.1	Installation.....	10
3.1.1	Default Installation.....	10
3.1.2	Customer-Specific-Installation.....	13
3.1.2.1	Firmware Updates.....	14
3.2	Operation.....	15
3.2.1	USB VCOM Port Emulation.....	15
3.2.2	Turning the Module on and off.....	15
3.2.3	Different Usage of AT Commands on USB and ASC0/ASC1.....	16
3.2.3.1	Bit Rate.....	16
3.2.3.2	Modem Control and Status Lines.....	16
3.2.3.3	Flow Control, Character Framing.....	17
3.2.3.4	USB and I ² C/SPI Usage.....	17
3.2.4	USB and Multiplex Mode.....	17
3.2.5	Power Management.....	18
4	USB Modem in Windows CE Environment	19
4.1	Driver Integration.....	19

Tables

Table 1: Abbreviations 8

Figures

Figure 1:	USB modem connections	9
Figure 2:	Phone and Modem Options	11
Figure 3:	Advanced COM port settings.....	12
Figure 4:	Driver Signing Options.....	13
Figure 5:	Bit Rates	16
Figure 6:	Serial USB Driver Integration System.....	19

0 Document History

New document: AN32: Integrating USB into GSM Applications, Version **02**

Preceding document: Integrating USB into MC75, TC6x, AC75 Applications, Version 01

Chapter	What is new
--	Changed document title. Added further supported products.
3.1.1	Modified description for driver signing option in section Installation Issues.

1 Introduction

Your wireless module integrates a self-powered USB 2.0 Full Speed interface (12 Mbit/s) compliant with the "Universal Serial Bus Specification Revision 2.0".

The purpose of this Application Note is to provide a detailed technical specification for the module's USB interface and to assist application engineers in designing applications with full support of USB by describing the performance scope USB interface.

1.1 Supported Products

This Application Note applies to the following Siemens products:

- MC75 Module
- TC63 Module
- TC65 Module
- AC65 Module
- AC75 Module
- XT65 Module
- XT75 Module

1.2 Related Documents

- [1] Hardware Interface Description related to your Siemens product
- [2] AT Command Set related to your Siemens product
- [3] DSB75 Support Box - Evaluation Kit for Siemens Cellular Engines
- [4] Universal Serial Bus Class Definitions for Communication Devices, Version 1.1 January 19, 1999
- [5] Application Note 16: Upgrading Firmware
- [6] Multiplexer Driver Installation Guide
- [7] CE-USB-Driver User Guide
- [8] USB.ORG: www.usb.org/developers/docs/USB_LANGIDs.pdf

To visit the Siemens Website you can use the following link:

<http://www.siemens.com/wm>

1.3 Abbreviations

Table 1: Abbreviations

Abbreviation	Description
ACM	Abstract Control Model
ACPI	Advanced Configuration and Power Interface
ASC0 / ASC1	Asynchronous Controller. Abbreviations used for the first and second serial interface of MC75/TC6x products.
B2B	Short for board-to-board connector
COM port	Communication port, serial port
CDC	Communication Device Class
CRC	Circular Redundancy Check
DSB	Development Support Box
DTR	Data Terminal Ready
IGT	Ignition
ME	Mobile Equipment
TA	Terminal Adapter
TE	Terminal Equipment
TXD	Transmit direction
RXD	Receive direction
USB	Universal Serial Bus
VCOM	Virtual COM port
VCU	Virtual COM port USB

A complete list of abbreviations is provided in [\[1\]](#).

2 General Features of the USB interface

2.1 Technical Specification

- USB 2.0 Full Speed compatible (12 Mbit/s)
- Compliant with "Universal Serial Bus Specification Revision 2.0"
- Self-powered device, control via USB host only (for details see [1])
- Supports Multiplex mode, i.e. the USB interface can be partitioned into virtual channels
- Supported configurable IDs: Vendor ID, Product ID and Serial number (MC75, AC65/AC75, XT65/XT75only)

The USB interface is primarily intended for use as command and data interface and for downloading firmware. The USB I/O-pins are capable of driving the signal at min 3.0V. They are 5V I/O compliant.

2.2 USB Modem Profile

The module's USB stack implementation is based on the Communication Device Class Abstract Control Model specification (CDC ACM v1.1) as defined for the Universal Serial Bus by the USB Implementer Forum. For details see [4]. By supporting ACM the USB modem is integrated as a USB modem device with a virtual COM port and understands standard V.25ter (AT) commands.

2.3 USB Interface Connections

The module supports a USB 2.0 Full Speed (12Mbit/s) device interface. It can be operated on a USB 2.0 root hub, i.e. a PC host working as Full Speed hub or High Speed hub in Full Speed mode. The module does not support a generic USB 2.0 High Speed hub which translates High Speed (480 Mbit/s) to Full Speed (12 Mbit/s). Therefore, the module's USB interface should be directly linked to a PC as shown in the figure below:

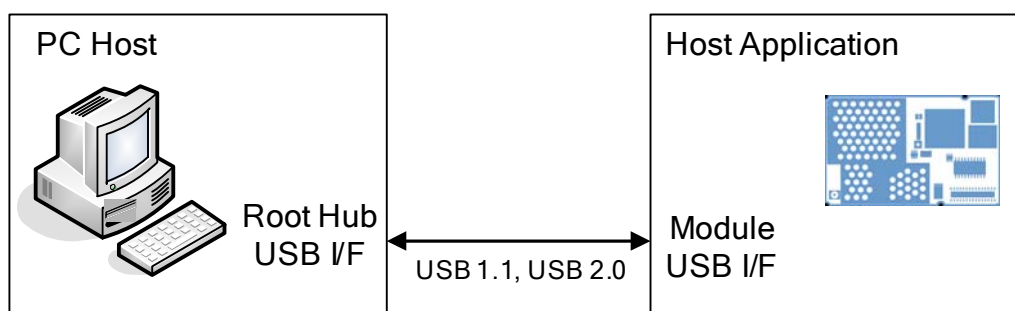


Figure 1: USB modem connections

For details on how to properly connect the module's USB interface to the host application see [1].

3 USB Modem in Windows XP/2000 Environment

3.1 Installation

This section assumes that you are familiar with installing and configuring a modem under Windows XP/2000. As Windows XP/2000 uses multiple methods to access modem settings this section provides only a brief summary of the most important steps.

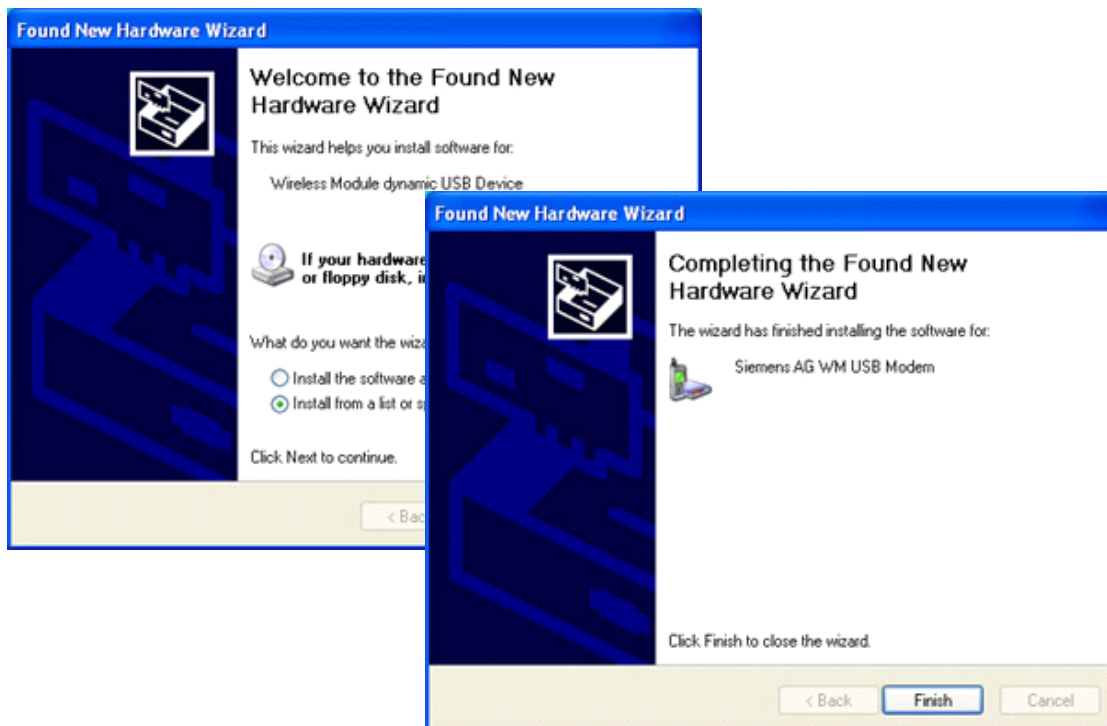
3.1.1 Default Installation

Take care that the modem setup file `usbmodem.inf` delivered with the module is at hand. Connect the USB cable to the module's host application (for example the evaluation board DSB75) and the PC. Windows detects the module as a new USB modem and opens the *Found New Hardware Wizard*. Follow the instructions on the screen and specify the path to the modem setup file `usbmodem.inf`. The modem setup file includes the Windows specific USB driver `usbser.sys`. The USB driver supports a virtual COM port emulation via USB and implements a USB modem based on the communication device class CDC ACM v1.1 (see [4]).

Notes for *Windows 2000* only:

- During the installation procedure you will be prompted for the `usbser.sys` driver. Make sure the file is present before you start the above inf file. The `usbser.sys` file is not delivered as a single file, but must be extracted from a Windows 2000 cabinet file. This is either the file `driver.cab` located in the I386 folder of the original Windows 2000 CD or a later cabinet file inside the Service Pack. SP4 for example includes the `sp4.cab` file that can be found in its I386 folder. The `usbser.sys` driver from the Service Pack has priority over one provided with the standard Windows 2000 install CD.
- It is necessary to restart Windows 2000 to make the changes take effect.

Windows copies the required software to your computer and configures the modem by assigning an available (virtual) COM port to the physical USB port. If you are already using more than one COM port then the next available one will be allocated. Click *Finish* to complete the installation.



3.1 Installation

After installation the “Siemens AG WM USB Modem” is listed under *Control Panel | Phone and Modem Options | Modems*

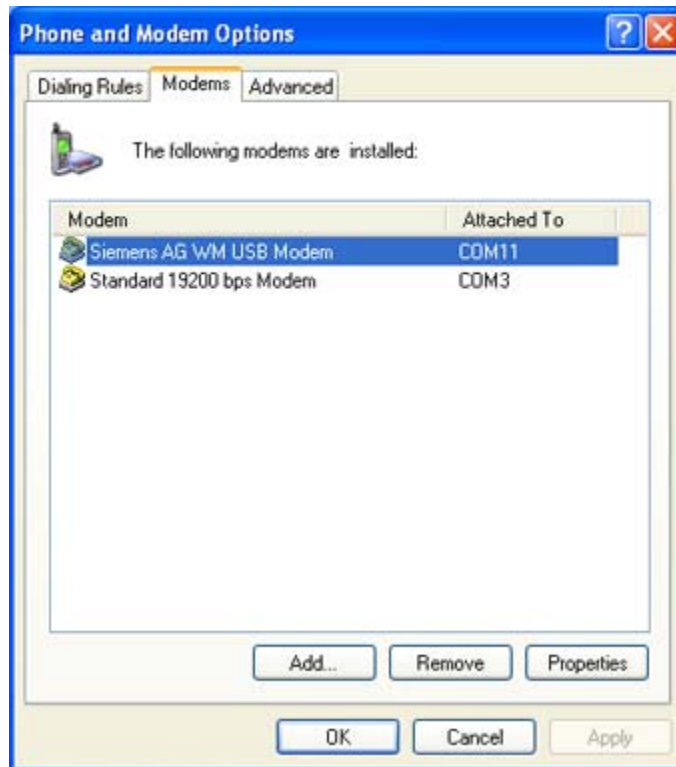


Figure 2: Phone and Modem Options

Installation Issues

- If Windows fails to assign the next free COM port to the USB modem and e.g., allocates a COM port already used by another modem you can manually select a free port as follows:

Open the Windows *Device Manager*, select the installed “Siemens AG WM USB Modem”, click *Properties*, select the *Advanced* tab and click *Advanced Port settings*. From the listbox *COM Port Number* choose a free port. To make the changes take effect disconnect and re-connect the USB cable. If not yet successful, also restart Windows.

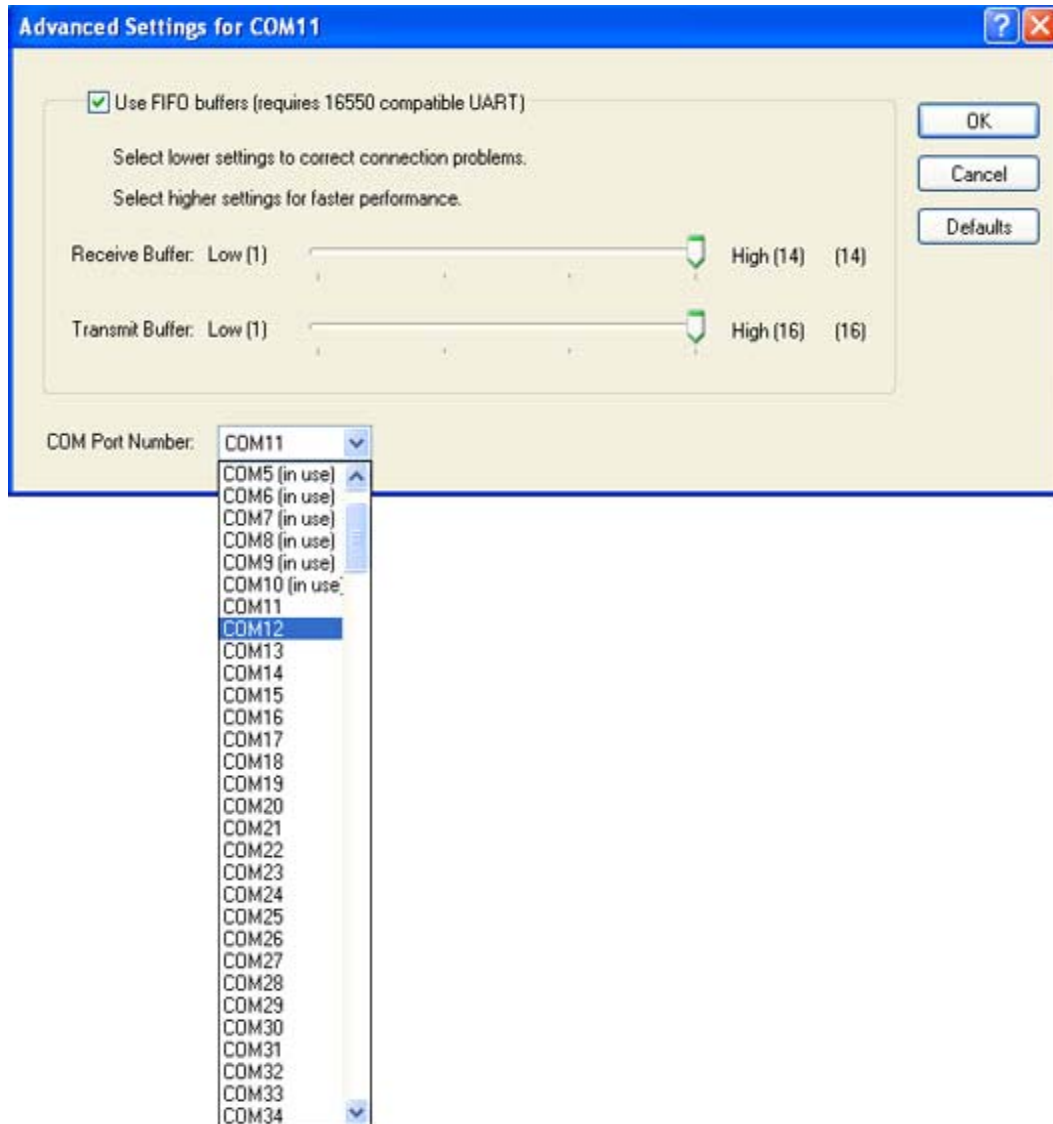


Figure 3: Advanced COM port settings

3.1 Installation

- During installation the message “Driver is not digitally signed, driver will not be installed” may pop up. In this case you should abort the installation, select *Ignore* as *Driver Signing Option* under Control Panel | System | Hardware | Driver Signing and restart the installation.

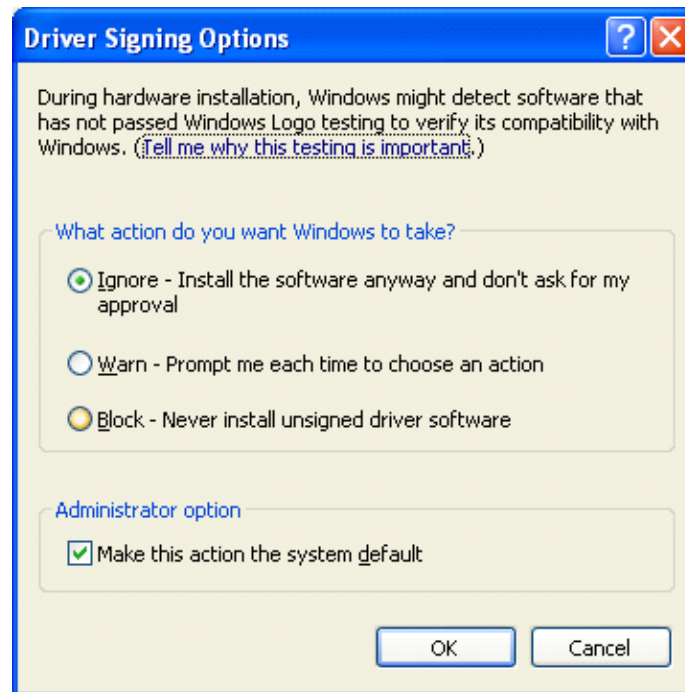


Figure 4: Driver Signing Options

- After installation the input echo may be disabled, i.e. a terminal program might not display the entered AT commands. To enable the input echo, please enter the AT command ATE1.

3.1.2 Customer-Specific-Installation

For **MC75**, **AC65/75** and **XT65/75** modules it is possible to install the USB modem with customer-specific device settings, e.g., a particular vendor ID or a manufacturer string to be displayed as hardware device name. These settings will have to be configured prior to the actual USB modem installation with the AT[^]SCFG parameter "Serial/USB/DDD".

The parameter setting <deviceDescr> determines whether to use the Siemens standard USB modem device descriptor configuration (as for the default installation) or a customized device descriptor configuration:

- If <deviceDescr>=0 is set, it is not possible to change any other parameter settings related to "Serial/USB/DDD".
- If <deviceDescr>=1 is set, all other parameter settings related to "Serial/USB/DDD" can be changed. These other parameter settings include a customer-specific language, vendor and product ID as well as specific manufacturer, product and serial number strings. For more information on the AT[^]SCFG parameter "Serial/USB/DDD" see [2].

Note: Customizing the USB device descriptor configuration requires a unique vendor ID obtained from the USB Implementers Forum. For more information please refer to <http://www.usb.org>.

The AT[^]SCFG read command returns amongst others the device descriptor configuration (in this case the Siemens default settings for the MC75 module):

```
AT^SCFG?
```

```
...
```

```
^SCFG: "Serial/USB/DDD","0","0","0409","0681","0034","Siemens AG Wireless Modules", "MC75", ""
```

```
...
```

3.1 Installation

A customer specific device descriptor configuration would look as follows:

```
AT^SCFG=
  "Serial/USB/DDD"=1,    // customer setting
  0,                    // for future use, to set more than one descriptor
  0409,                // Language Code
  0123,                // Vendor ID
  4567,                // Product ID
  MECOM Ltd.,          // Manufacturer String Descriptor
  ME10A,                // Product String Descriptor
  1230456              // Serial Number String
```

Apart from modifying the USB device descriptor configuration the USB modem setup file `usbmodem.inf` needs to be changed as well, usually at least the vendor ID, product ID and, optionally, the manufacturer string. This is because the strings specified as customized "Serial/USB/DDD" parameter settings must be identical with their counterparts in the modem setup file. An easy way to adapt the modem setup file is copying the default Siemens `usbmodem.inf` file and overwriting the appropriate values.

The following extract from an `usbmodem.inf` file highlights the sections that will have to be adapted as a minimum. VID and PID set the vendor and product ID, Siemens and Siemens01 specify the manufacturer and modem strings:

```
...
[Models]
%Siemens01% = Modem1, USB\VID_0681&PID_0034 ;Identifies Vendor = Siemens WM, Product = Modem
...
[Strings]
MS1 = "Microsoft"
USBFilterString ="USB Modem Driver"
Siemens = "Siemens AG"
Siemens01 = "Siemens AG WM USB Modem"
...
```

After having changed the USB device descriptor configuration and the modem setup file, the actual USB modem installation can be completed as described for the default installation described in [Section 3.1.1](#).

3.1.2.1 Firmware Updates

If the module's firmware is updated via USB modem, the Siemens default device descriptor configuration must be present. Therefore, before updating the firmware be sure that the `<deviceDescr>=0` is activated. This can be done by entering the AT command:

```
AT^SCFG="Serial/USB/DDD",0
```

The corresponding result code lists the Siemens default configuration for the module, in this case MC75:

```
^SCFG: "Serial/USB/DDD", "0", "0", "0409", "0681", "0034", "Siemens AG Wireless Modules", "MC75", ""
```

Remember to restart the ME after reverting to `<deviceDescr>=0` and to re-install the default USB driver (i.e., to implement the default `usbmodem.inf` file) before starting the firmware download via `gWinSwup`. For more information on how to download firmware see [\[5\]](#).

3.2 Operation

3.2.1 USB VCOM Port Emulation

The virtual COM (VCOM) port emulation under Windows XP is involved with the usage of a USB serial number provided by the device: If a serial number is assigned this number identifies the USB device within the operating system. If no serial number is assigned the USB device is identified by the allocated USB port number.

MC75, TC65, TC63, AC65/AC75 and XT65/XT75 modules have no default USB serial number and are therefore identified by their physical USB port number. This means that when the USB port of the module is detached and replugged to another USB root hub port the USB driver must be set up again for the new physical USB port. If there are several USB devices without serial number each must have a specific physical USB port number depending on the USB root hub port it is connected to.

However, for MC75, AC65/AC75 and XT65/XT75 modules it is possible to use varying serial numbers. For more information on how to configure varying serial numbers see [Section 3.1.2](#) and [2]. The handling of a single module becomes independent of the USB root hub port it is plugged in. The module is always assigned the same USB VCOM port and no additional explicit driver installation is necessary. For multiple modules however, each module is assigned a defined USB VCOM port. The USB VCOM port no longer depends on the physical USB root hub port the module is plugged to. It always remains the same. There is an initial explicit installation required for every new module attached to the USB host.

It is also possible to use a fixed serial number for MC75, AC65/75 and XT65/XT75 modules. However, fixed serial numbers are not supported by Microsoft and therefore not recommended.

3.2.2 Turning the Module on and off

In a Windows environment, the USB COM port emulation causes the USB port of the module to appear as a virtual COM port (VCOM port). The VCOM port emulation is only present when Windows can communicate with the module, and is lost when the module shuts down. Furthermore, the operating system maps the VCOM port number to an internal ID that changes each time the USB device is restarted, i.e. the VCOM port number is permanent, but the internal ID changes. Therefore, the host application or Terminal program must be disconnected from the USB VCOM port each time the module is restarted.

Restart after shutdown with AT^SMSO:

After entering the power down command AT^SMSO on one of the interfaces (ASC0, ASC1, USB) the PC application or Terminal program used on the USB VCOM port must be closed before the module can be restarted by activating the IGT line, or by replugging the USB cable between the module's host application (e.g., the DSB75 Support Board) and the PC.

Software reset with AT+CFUN=x,1:

Likewise, when using the reset command AT+CFUN=x,1 ensure that the PC application or Terminal program on the USB VCOM port is closed down before the module restarts. The best approach is to quit this program first, and then enter the AT+CFUN=x,1 command from ASC0 or ASC1. Otherwise, if AT+CFUN=x,1 is entered on the USB interface then the PC application or Terminal program on the USB VCOM port must be closed down immediately after sending the command and receiving the OK response.

Shutdown from Charge-only mode and restart:

As specified in [1], the AT^SMSO command cannot be used to power down the module when in Charge-only mode. Instead, it is necessary to disconnect the charger.

If the USB port is involved, the PC application or Terminal program connected to the VCOM port must be closed down before disconnecting the charger.

3.2.3 Different Usage of AT Commands on USB and ASC0/ASC1

The following sections discuss AT commands that have different effects if used on the USB interface or the serial interfaces ASC0 and ASC1. For further detail see also [2]. Please note that there is no ASC1 interface available for XT65/XT75 modules.

3.2.3.1 Bit Rate

As specified in Section 2 the MC75/TC6x module supports a bit rate of 12 Mbit/s on the USB. This is the maximum bit rate shared by all USB devices connected to the same application platform. Consequently, if more than one USB device is connected the bit rate may be lower.

The bit rate settings known from the serial interfaces ASC0/ASC1 are not relevant on the USB. Therefore, all bit rate settings made by the following AT command are ignored on the USB port:

AT+ILRR (sets TE-TA local rate reporting)

AT+IPR (sets fixed local rate)

The same applies to the bit rates selected on the Modem Property pages of the Windows Control Panel. Although the settings are visible and can be changed, none of the selected bit rates is of use for USB.

Despite that bear in mind that the bit rate selected on the Modem Property pages will be shown in the status display window of the Dial-Up network connections. This may be confusing to the user.

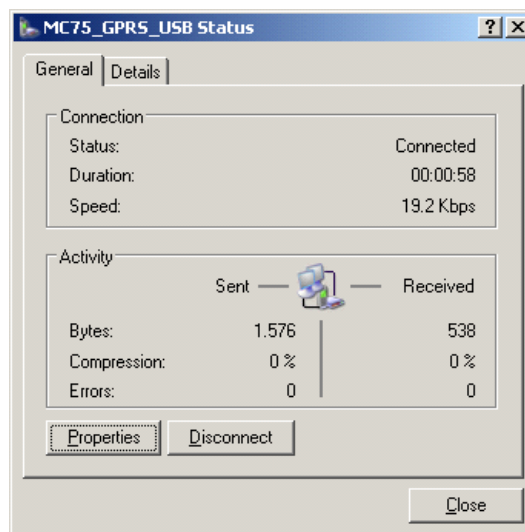


Figure 5: Bit Rates

3.2.3.2 Modem Control and Status Lines

The VCOM USB driver (usbser.sys) uses the DTR control line for the signaling between DTE and DCE to indicate that the VCOM port is open and ready to receive data. This is independent of the selected hardware handshake mechanism, but can influence the communication for the used virtual COM port if the signaling state is changed explicitly.

Besides, the signals RTS, DCD, DSR and RING can be used on the USB port for signaling purposes.

DTE -> DCE

DTR: DTR signals that the host is ready to send or receive data on the VCOM port. Behavior depends on configuration command AT&D too.

RTS: RTS signals will only be transmitted if DTR changes too.

Unlike legacy serial interfaces the DTR and RTS signals cannot be used to start the module.

DCE -> DTE (Fastest response 128ms after triggering)

DCD: Signaling depends on configuration command AT&C.

DSR: Signaling depends on the configuration command AT&S.

RING: Indicates an incoming call or URC (as configured by AT^SCFG; see [\[2\]](#))

CTS: This signal is not available inside CDC ACM.

3.2.3.3 Flow Control, Character Framing

The direct flow control mechanism driven by hardware (RTS/CTS) or software (XON/XOFF) typically used for a legacy COM ports is not used by the USB VCOM port emulation, simply because it is not needed. USB already supports the native flow control via protocol (ACK/NAK technique). Therefore, the settings of the following AT commands are ignored if used on the USB port:

AT\Qn (sets type of flow control)

AT+ICF (sets serial interface character framing)

AT+IFC (sets flow control separately for data directions)

AT^STPB (transmits Parity Bit (for 7E1 and 7O1 only))

3.2.3.4 USB and I²C/SPI Usage

On TC65 and TC63 modules it is possible to operate the I²C and SPI interfaces in parallel with the USB interface.

This is not the case for MC75, AC65/AC75 and XT65/XT75 modules however. The USB interface will be disabled as soon as the I²C or SPI channel is opened using the AT^SSPI command. To activate the USB interface again after closing the I²C or SPI channel it is necessary to restart the MC75, AC65/AC75 or XT65/XT75 module, either by switching the ignition line or entering the command AT+CFUN=1,1. The terminal program or host application connected with the USB interface must be closed before activating the USB (see [Section 3.2.2](#)).

3.2.4 USB and Multiplex Mode

To configure the USB interface for use with the WinMux2k driver the virtual COM port assigned to the module's USB interface shall be set to max. 115200 bps. For more details see [\[6\]](#). The setting is only needed to open the virtual multiplex channels on the virtual COM port assigned to the USB interface. There is no loss of performance because the virtual baud rate on the USB interface has no influence on the data transfer rate (see [Section 3.2.3.1](#)).

ASC0 and ASC1 are disabled when the multiplexer is enabled on the USB interface. The USB interface and the multiplex channel 3 are using the same parameters, and thus, the same user defined profile (if any). As a result, non-volatile settings and a user profile stored on multiplex channel 3 will take effect on the USB interface after closing the multiplexer and starting up USB. Likewise, non-volatile settings and a user profile stored on the USB interface will be loaded on multiplex channel 3.

3.2.5 Power Management

The modules support various power saving modes. These can be configured with the AT+CFUN command alone as long as there is no active USB connection. For details on how to configure power saving with AT+CFUN see [\[2\]](#).

However, if the PC host application employs the USB interface, power saving must be configured both on the module and on the PC host, i.e. the PC host must be set to suspend mode (usually Selective or Global Suspend) by the PC host's power management system. The PC host's power management system (e.g., ACPI = Advanced Configuration and Power Interface management system) has to be configured to support power saving for the USB port the modem device is located at.

If you modify the AT+CFUN parameters only, the selected <fun> level is accepted, but the active USB connection keeps the module alive and thus prevents power saving until the USB plug is disconnected. Also, if the module is in SLEEP mode, reopening the USB connection will cause the module to stop power saving although the configured CFUN level did not change.

A major feature of the module's USB functionality in connection with the PC host's power management system is **USB Remote Wake-up**. With this feature it is possible for the USB modem device to wake up the USB host controller from any sleeping state. Reasons for a remote wake up could be: An incoming call (RING) indication or some other important message, i.e., Unsolicited Result Code (URC). In these cases the USB modem device wakes up the USB PC host and the signal or message may for instance be displayed on a terminal application.

4 USB Modem in Windows CE Environment

To integrate a USB modem into a Windows CE 5.0 environment with USB host controller, Siemens WM provides the source code required to generate the `usbser.dll`, i.e., the serial USB driver. This driver realizes a virtual COM port for the operating system in order to access the module on its USB interface by means of this COM port. For details on the integration and operation of the “serial USB driver” please refer to [7].

4.1 Driver Integration

As is common for Windows CE device drivers, the serial USB driver is built as a dll. The dll exports the standard USB interface accessed by the operating system in order to install or uninstall the driver and to attach or detach the corresponding USB device, if its hardware has been located on the USB bus by the USB host driver. In addition, the dll exports the device context interface of the virtual COM port and presents it to the system with the device name "VCU". It is also possible to apply a multiplex driver to that device.

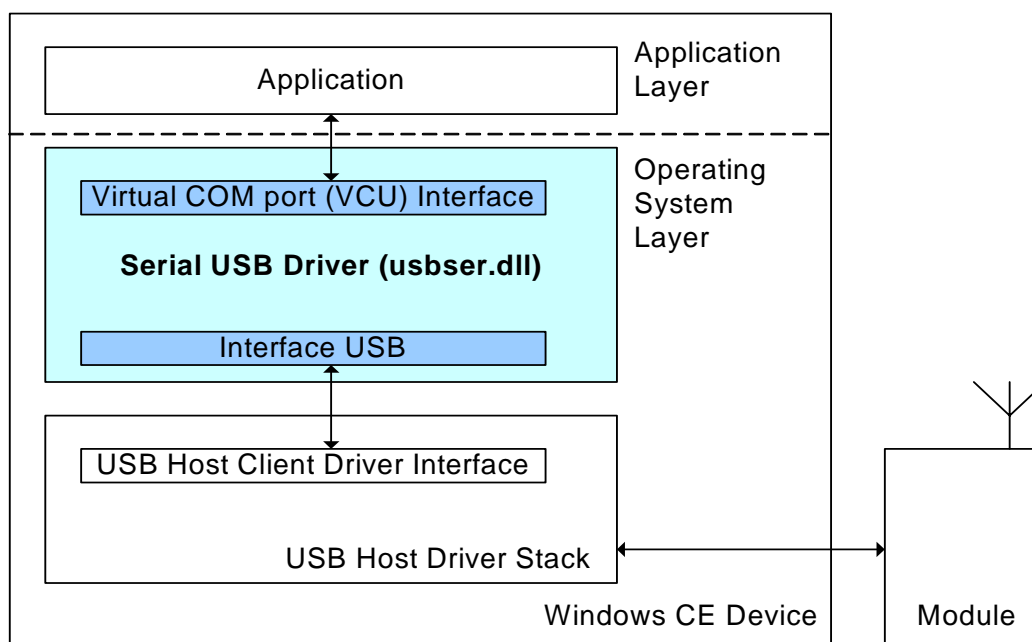


Figure 6: Serial USB Driver Integration System

For more information on how to integrate and operate the Siemens serial USB driver see [7].