

PicoCOM1

First Steps

Version 1.06 Date 15 April 2008

© by F & S Elektronik Systeme GmbH 2007

F & S Elektronik Systeme GmbH
Untere Waldplätze 23
D-70569 Stuttgart
Phone.: (+49)(0)711/1237220 Fax: (+49)(0)711/12372299

Table Of Contents

1	Getting started	2
1.1	Powering-on PicoCOM1	5
2	Updating the PicoCOM1	8
2.1	Using NetDCUUSBLoader.EXE utility	8
2.2	Updating – NBOOT	10
2.3	Updating – EBOOT	10
2.4	Updating – Windows CE Kernel	11
2.4.1	Partitioning the Flash storage	12
3	Configuration	14
3.1	Using NDCUCFG.EXE utility	14
3.2	Network interface configuration	16
3.2.1	Network – General Facts	16
3.2.2	Network – Network address	17
3.2.3	Network – Set up with NDCUCFG.EXE	18
3.2.4	Network – Saving the parameters to registry	19
4	Remote Tools (VS2005 / 2008)	20
4.1	Remote Display	20
5	Software Development	22
5.1	Visual Studio – managed Code	22
5.2	Visual Studio – native Code	23
5.3	Visual Studio – debugging an device application	24

1 Getting started

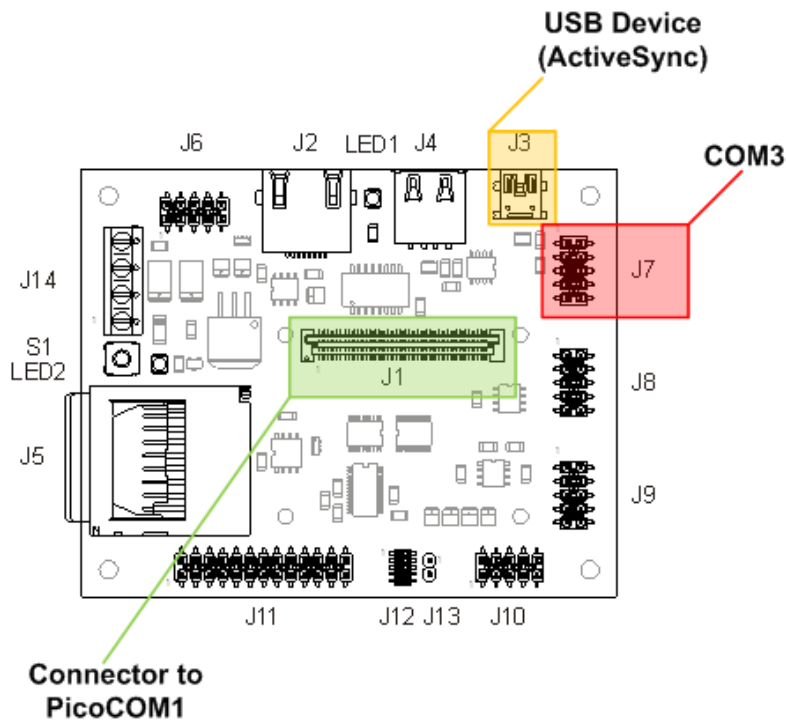
This documentation is a step by step introduction in how to use PicoCOM1 with PicoCOM1-Startintf. The connectors referenced in this documentation are on the PicoCOM1-Startintf. The documentation does NOT explain how to use PicoCOM1 without PicoCOM1-Startintf. The latest version of this document can be found at

<http://www.picocom.de>

Additional support information can be found in our discussion forum at

<http://www.picocom.de>

Next picture shows the PicoCOM1-Startintf for PicoCOM1 with the position and description of connectors.



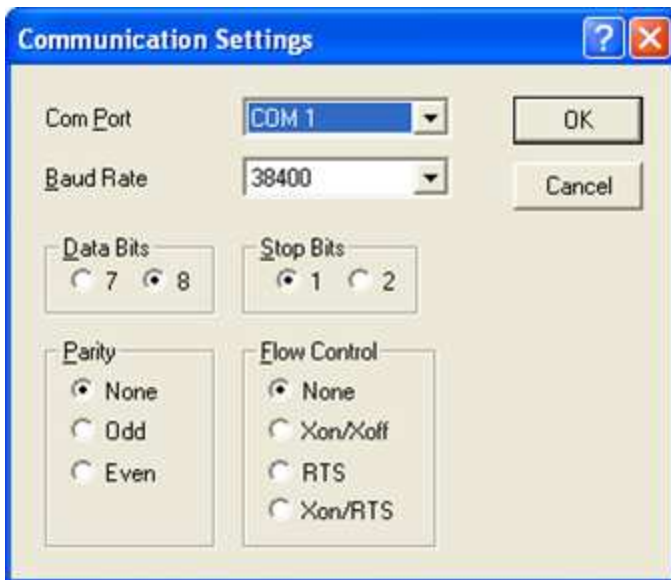
PicoCOM1-Startintf without PicoCOM1.

Important:

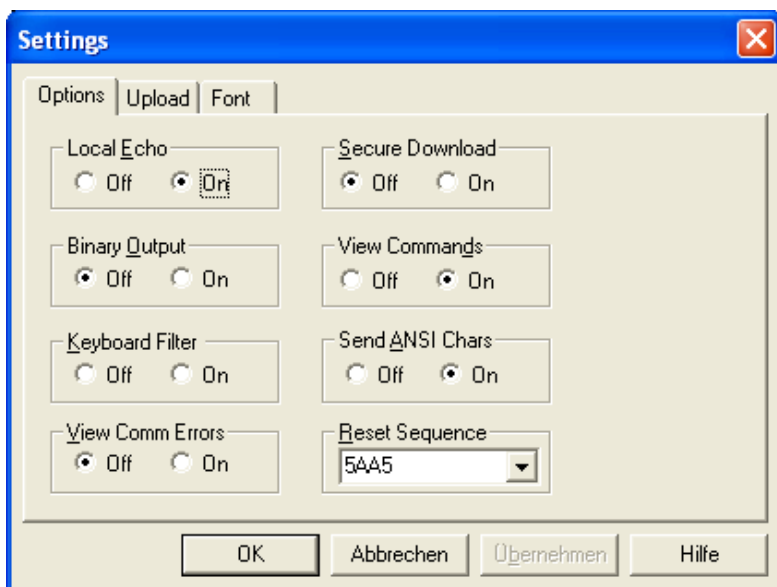
1.1 Powering-on PicoCOM1

Before you power on PicoCOM1, you should make a serial connection between PicoCOM1 and your PC. Please use the cables shipped with the PicoCOM1 Starter Kit. On the PC, you should have installed DCUTerm.exe as terminal program. Follow the steps below, to make a connection:

- Install DCUTerm.exe on your PC
- Configure DCUTerm as shown in the following picture:

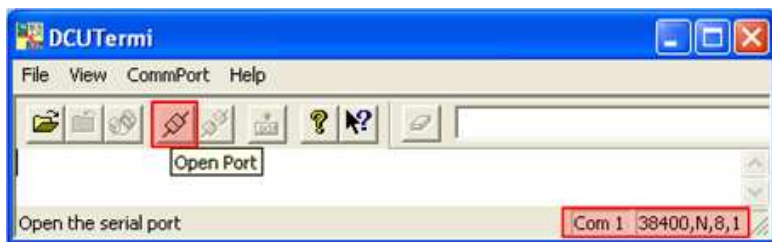


CommPort menue → Settings



View menu → Options

- Press the connect button in DCUTermi.exe



- Use serial cable shipped with PicoCOM1 Starter Kit to make a connection between COM3: and your PC

- Make a cable between power connector (gray) of PicoCOM1-Startintf and your power supply. At this moment you only need to connect ground and +5V (2A).

After these steps the LED 1 should be on and you should see output similar to the following in the terminal window of DCUTerm.exe.

```
Microsoft Windows CE Ethernet Bootloader Common Library  
Version 1.1 Built Feb 20 2008 19:04:33  
Master Clock is 105050715 Hz  
Microsoft Windows CE Bootloader for PICOCOM1 Built Mar 17 2008  
Portions copyright (c) 2007 F&S Elektronik Systeme GmbH  
Boot Loader, Version 0.1  
NBoot, Version C100
```

```
INFO: OEMLaunch: Jumping to Physical Address 0x20101000h  
(Virtual Address 0x80101000h)...
```

```
NetDCU Config Utility Ready  
Version: 033  
Type help for commands
```

```
!>
```

If you don't see text similar to the above one:

- Check serial connection
- Check power supply
- Check settings of DCUTerm.exe

Note, by default the NDCUCFG.EXE runs on COM3: and the processors debug output is disabled. You can enable and disable the debug output by step into the bootloader (press "shift+s" into the terminal program connected to PicoCOM1s COM3: and hold the keys while power on PicoCOM1) and use the command "O". You can also move NDCUCFG.EXE to COMX: by modify registry value "Port" under HKLM\system\ndcucfg ("Port"=string: COMX:).

2 Updating the PicoCOM1

The Startup process of PicoCOM1 is divided into three steps:

- NBOOT (Stepping Stone Bootloader)
 - Loading the Windows CE Bootloader
- EBOOT (Windows CE Bootloader)
 - Loading the Windows CE Kernel
- Windows CE Kernel

You can update all these three parts separately by different processes.

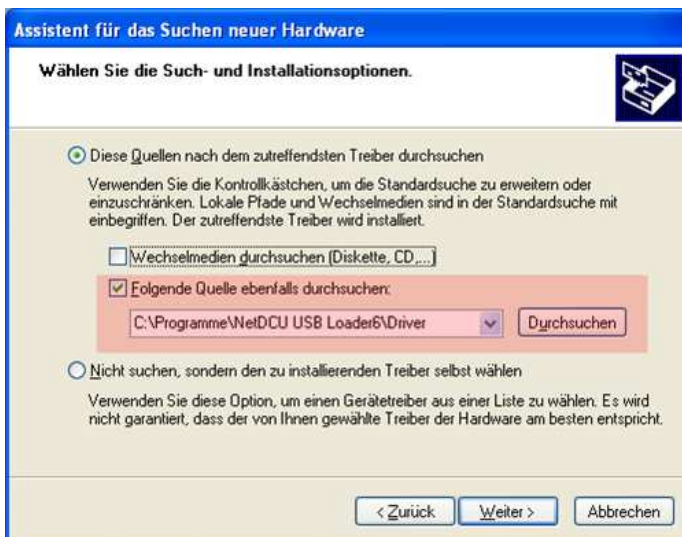
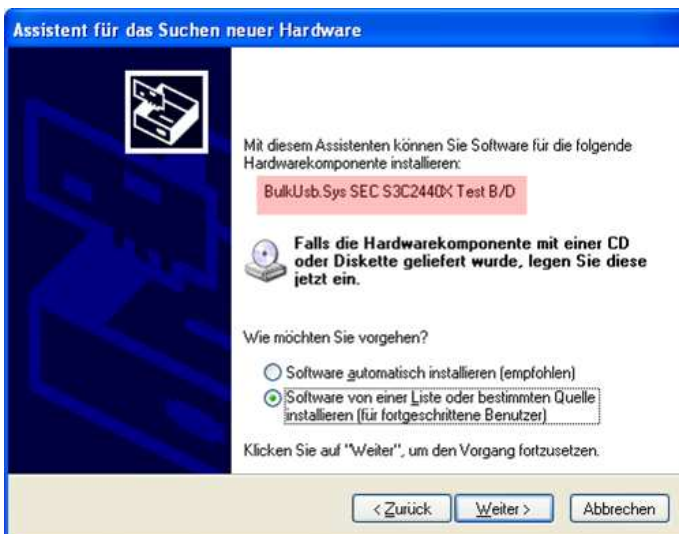
2.1 Using NetDCUUSBLoader.EXE utility

The preferred method to update PicoCOM1 is the utility NetDCUCUSBLoader.EXE which enables you to download the bootloaders and the Windows CE Kernel to PicoCOM1 using USB.

When connecting PicoCOM1 and NetDCUUSBLoader for the very first time you have to install a driver on your development PC. That driver is shipped with NetDCUUSBLoader installer and gets copied to the installation directory of NetDCUUSBLoader. The procedure of downloading a new bootloader or Windows CE Kernel with this utility is described in chapter 2.2 to 2.4.

Installing the driver on your development PC:

When trying to download a bootloader or kernel for the very first time the Windows OS on your development PC asks you for installing a special driver named **bulkusb.sys** which can be found under <InstallationPathOfNetDCUCUSBLoader>\Driver.



2.2 Updating – NBOOT

To update NBOOT press 'SHIFT' + 's' while powering on the PicoCOM1. Doing that you will enter the Windows CE Bootloader EBOOT. To download the new NBOOT (nboot.bin) press 'D' 'U' (press '?' to list the commands for EBOOT) and start the NetDCUUSBLoader.EXE utility on your PC. In NetDCUUSBLoader.EXE choose the respective *.nbi file and click on 'Start'.

In case of updating NBOOT the *.nbi file has the following body:

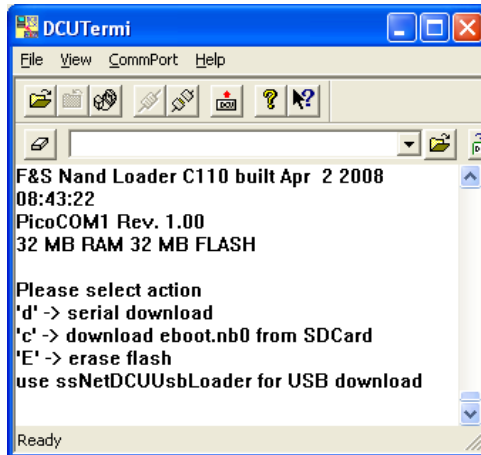
```
author = "F&S EElektronik Systeme GmbH"  
customer="XXX"
```

[Files]

```
StepStone=<PATH>\nboot.bin
```

2.3 Updating – EBOOT

To update EBOOT press 's' while powering on the PicoCOM1. Doing that you will enter the Mini Bootloader NBOOT.



As you can see you can use different processes to download the new EBOOT. The preferred method is **USB download** via NetDCUUSBLoader.EXE. Therefore choose 'u' and start NetDCUUSBLoader.EXE on your PC. In NetDCUUSBLoader select the respective EBOOT.nb0 file and click on 'Start'.

You can also download EBOOT.nb0 by **serial download**. Therefore press 'd' and click on 'Transmit binary file' in the File-Menue of the DCUTerm utility. Select the respective EBOOT.nb0 file and press 'open'. After the download has finished press 'f' to save.

note:

At the moment only serial download is supported.

2.4 Updating – Windows CE Kernel

You can update the Windows CE Kernel by Ethernet or by the preferred method via USB and the NetDCUUSBLoader.EXE utility. Therefore press 'Shift' + 's' while powering on the PicoCOM1. You are then entering the Windows CE Bootloader EBOOT.

Download by Ethernet:

To download the Windows CE Kernel by Ethernet press 'D' 'E' and start the ESHELL.EXE utility on your desktop PC. As soon as PicoCOM1 sends 'Sent BOOTME to 255.255.255.255' you can select the device in the 'Select Device' dialog of ESHELL.EXE. Then choose 'Select image' from the File-Menue of ESHELL.EXE and select the respective NK.bin file. Press 'open' – this starts the download to PicoCOM1.

Download by USB:

To download the Windows CE Kernel by USB press 'D' 'U' and start the NetDCUUSBLoader.EXE utility on your desktop PC. As soon as the connection is established the button in the top right

corner of NetDCUUSBLoader.EXE turns from red to green. Select the respective NK.bin file and click on 'Start'.

note:

The Windows CE Kernel can be downloaded to RAM or to Flash.
EBOOT commands:

- MR - Store kernel in RAM memory
- MF - Store kernel on Flash disk

To start Windows CE Kernel after reboot you have to execute the EBOOT command:

- L1 - Launch previously stored kernel after boot

2.4.1 Partitioning the Flash storage

There are up to 3 partitions possible on PicoCOM1.

- **OS-Image (BINFS):** The Windows CE Kernel is stored here.
- **FFSDISK:** This partitions can be used to store user data and applications. It is available under \FFSDISK on a running Windows CE system.
- **Extended Partition:** The extended partition must be administrated in WindowsCE. There are no partitions available by default so you have to create them using the StorageManager. The size of this partition might be 0 in most cases.

note:

By default BINFS is 12 MB large which will be to small for the currently available kernel image. Possibly we will adapt this default size in future bootloader versions.



To partitioning the flash storage to store the PicoCOM1 Windows CE Kernel please press 'SHIFT'+s' to enter the Windows CE bootloader EBOOT. With the command '?' you get a list of EBOOT commands. If you can see, you can partitioning the storage with command 'P' ("Setup partitioning information").

> P

-----PARTITION CONFIGURATION-----

Current settings:

Total : 33 MB

OS-Image: 15 MB

FFSDISK : 18 MB, Part type: FAT

SECOND : 0 MB, Part type: EXTENDED

Enter maximal size for OS-Image [15]:

At this point you can resize the partions for BINFS, FFSDISK and Extended. Follow the instructions printed in your terminal program DCUTerm. You should see output similar to the printed below.

WindowsCE image and all data in FFSDISK will be erased.

Continue ? (Y/n) Y

FMD: Can't erase block 0x0

FMD: Can't erase block 0x1

.

.

FMD: Can't erase block 0x11

done

Create partition for boot section ...

.

.

Success

Create partition for kernel section ...

Success

Create partition for FFSDISK section ...

Success

You can ignore the FMD messages. Some blocks are reserved for storing the bootloader and other required data. You should not worry about them.



3 Configuration

Configuration of PicoCOM1 device is provided by different means. Though most powerful and acceptable way is running NDCUCFG.EXE software utility. In fact, this is a standard command prompt program allowing you to adjust variety of system parameters.

Most of changes done to PicoCOM1 device through NDCUCFG.EXE utility are stored in persistent system registry, and take effect after next reboot of PicoCOM1 device.

According to device's software architecture, this utility is automatically started on the COM3:. As well, the utility can be remotely executed over TELNET connection, once you have got network access to PicoCOM1 device.

All in all, software components and core of operating system running on PicoCOM1 offer you an easy and effective way to make necessary settings.

In chapter 3.2 comes a description of Network adjustment to make it possible to connect and transfer files via Ethernet. It also demonstrates pretty good how to use NDCUCFG.EXE utility.

3.1 Using NDCUCFG.EXE utility

note:

You can either run debug messages on COM3 or NDCUCFG.

To start NDCUCFG on COM3 please press 'SHIFT'+s' while powering on the PicoCOM1. Doing that you will enter EBOOT. Please press '?' to get a list with all EBOOT commands. With the command 'O' you can set 'Disable serial debug output during boot'. If you choose 'Y' the NDCUCFG utility gets started on COM3.

After booting the PicoCOM1 device wait until following prompt message will appear:

```
NetDCU Config Utility Ready  
Version: 030  
Type help for commands
```

```
!>
```

If you can see it, then you are ready to pass commands to NDCUCFG.EXE utility. Otherwise something went wrong. Check various parameters described in chapter 1.1.

If you have success on starting NDCUCFG.EXE over the serial line of your PC, then you can start passing commands to the utility. It's recommended that first command you issue is the command *help*. This is final part of what you will see on issuing it:

```
        . . . list of messages starts earlier  
backlight off  
start <file name>  
quit  
help  
help <command>  
!>
```

You definitely know how to use such trivial (but important!) commands as *help* and *quit*. For all other commands you can use hint given you in last string of above output. I.e. if you do not know how to issue command *backlight* then you type following and then press Enter:

```
!>help backlight
```

two possible ways of executing this command will be shown you in response. If you still interesting in what command *backlight off* does, just type and finish with Enter the following:

```
!>help backlight off
```

and you will get satisfying answer to you *help*-request.

To save the changes execute the command:

```
!>reg save
```

You have to reboot the device to make the changes effective. Upper examples demonstrate how the NDCUCFG.EXE utility functions in general. Now, let us set up the Network.

3.2 Network interface configuration

PicoCOM1 implements powerful and stable Ethernet interface which allows customer to create on its basis a variety of modern hardware Internet applications highly required by modern market of data processing and transporting appliances.

Ethernet interface implemented in PicoCOM1 meets 802.3 10BaseT specifications by IEEE, and provides safe data transfer on speeds up to 100 Mbit/sec.

3.2.1 Network – General Facts

Being integrated into IP-network, in order to get directly referred by other network devices, every PicoCOM1 device must obtain its own IP-address, unique within entire network segment. Such address along with other necessary parameters generally must be confirmed by network administrator.

Get a preferred IP-address from range of currently available IP-addresses (for example 192.168.5.5) , and mark this address as one currently being assigned to PicoCOM1. Ask your network administrator if you don't know how to obtain unused IP-address or see "Network – Network address".

Hardware layer of communication between network devices assumes every device to have one more address. This another kind of address is a so-called MAC-address, or 'Ethernet address', or 'physical address'. It is formed of six-byte sequence, and, in ac-

cordance to corresponding IEEE's regulations, is unique for every network device across the World.

3.2.2 Network – Network address

Every IP-Address can be split into the network address and station address. It's not part of this documentation to describe all details of this but we want to explain how you can obtain your network address from your PC.

Open command window and type:

```
C:> ipconfig
```

then press Enter. Output you get must be relative to following:

Windows IP configuration:

Ethernet Adapter RTL81391:

IP address: 192.168.5.131
Subnet mask: 255.255.255.0
Standard Gateway:

From this information you can calculate your network address. Interpret the values as hexadecimal values and do a logical and of IP address and subnet mask. The result is the network address.

192	.	168	.	5	.	131
255	.	255	.	255	.	0
192	.	168	.	5	.	0

So, for our example network address is 192.168.5.0 and station address within this network is 192.168.5.131. Only stations that are in the same network can communicate with each other.

3.2.3 Network – Set up with NDCUCFG.EXE

Almost all device settings can be configured by registry. Therefore you use the *reg* commands of the NDCUCFG.EXE utility as described below:

```
!>help reg
reg open
reg open <key>
reg open <key>
reg enum key <#>
reg enum key *
reg enum value <#>
reg enum value *
reg set value <name> dword <value>
reg set value <name> string <value>
reg set value <name> multi <value1>;<value2> ;<valueN>
reg set value <name> hex <value>,<value>,<value>
reg create key <name>
reg del value <name>
reg del key <name>
reg save
reg erase
!>
```

The Network parameters for PicoCOM1 can be found under:
[HKLM\Comm\EMACB1\Parms\TcpIp]

So execute the command:

```
!>reg open \Comm\EMACB1\Parms\TcpIp
OK
```

to access the network parameters. The output *ok* tells you that NDCUCFG.EXE could successfully open the path.

I.e. you can change the value *IpAddress* with the command:

```
!>reg set val IpAddress string "10.0.0.111"  
OK  
OK  
!>reg enum  
OK -> reg enum key \  
OK -> reg enum value \  
00 "IpAddress"=string:10.0.0.111 \  
01 "EnableDHCP"=dword:0 \  
02 "UseZeroBroadcast"=dword:0 \  
03 "DefaultGateway"=string:192.168.0.1 \  
04 "Subnetmask"=string:255.0.0.0 \  
05 "DNS"=string:0.0.0.0 \  
06 "WINS"=string:0.0.0.0 \  
OK
```

3.2.4 Network – Saving the parameters to registry

After changes as for type of network were correctly done, and special checking following it have approved this fact, it's suitable time to save those changes from RAM memory to physical media, so they will take an effect after next reboot of PicoCOM1 device.

Enter following:

```
!>reg save
```

and press Enter. Procedure of physical saving takes about two seconds – do nothing during this period! If you can see "OK" message again then it means that all the changes provided to PicoCOM1 system during current session of working with NDCUCFG.EXE utility are stored in persistent registry.

4 Remote Tools (VS2005 / 2008)

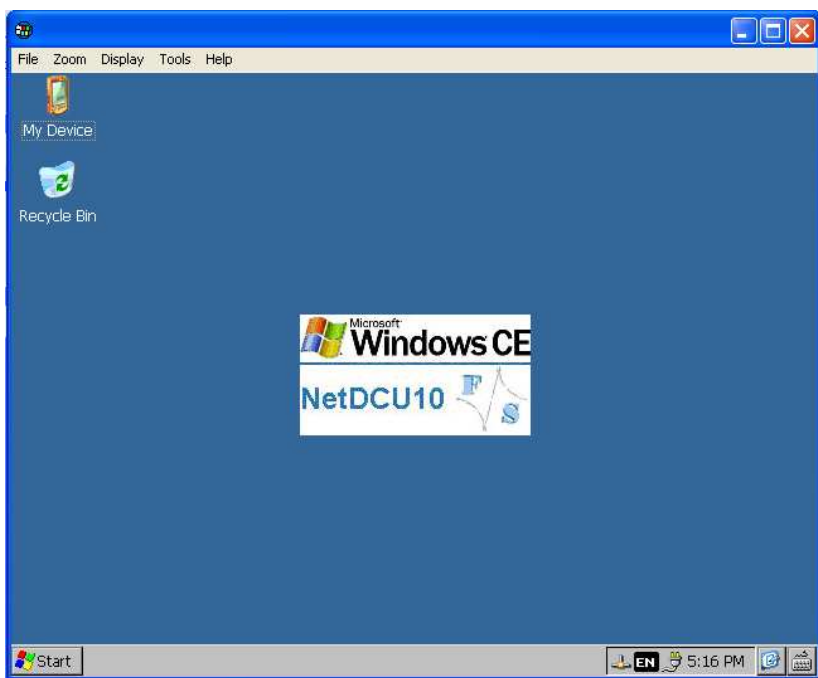
Microsoft Visual Studio 2005 / 2008 is shipped with a couple of useful Remote Tools.

- Remote File Viewer – File Explorer
- Remote Heap Walker – lists Heap per Process
- Remote Process Viewer – Task Manager
- Remote Spy - displays messages received by windows associated with applications running on a target device
- Remote Zoom In - On a development workstation, Remote Zoom-in displays a screen image from a target device
- Remote Display

4.1 Remote Display

Establish an ethernet connection to PicoCOM1 and start Remote Tool ,cerhost.exe' on development PC

- Registry Settings: [HKLM\System\GDI\Drivers]
 - „Display“ REG_SZ set to DDI_CER.dll



5 Software Development

For software development you have to use Visual Studio 2005 or newer. The kernel-image that you can download from the web side www.picocom.de includes already the Microsoft Compact Framework 3.5. This enables the developer to write managed code in C# or VB.NET. It is also possible to develop applications in native code (C++) using the Win32 API or MFC. To use native code you need to install the PicoCOM1 SDK that you also can download from the above mentioned web side.

To connect Visual Studio to PicoCOM1 for software development you can use a USB device connection or an Ethernet connection.

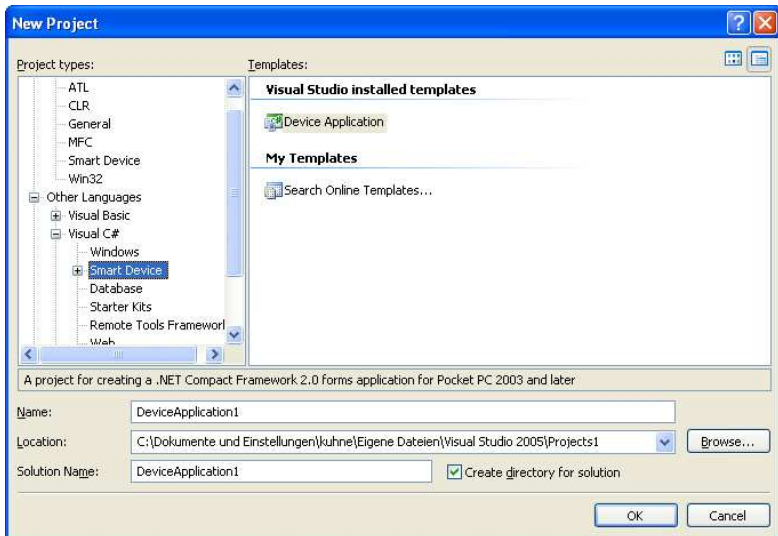
A detailed description how to connect via Ethernet is given by "Connect VS 2005 to NetDCU using LAN" which is analog to PicoCOM1 and you can download it from <http://www.fs-net.de/download/bin/> .

The best way (because easy to handle) is to connect via USB device using Microsoft ActiveSync. For this install the latest version of ActiveSync on your PC (download ActiveSync from <http://www.microsoft.com>) and connect PicoCOM1 and PC using the USB device connector shipped with the PicoCOM1-Startint. The connection is established automatically.

5.1 Visual Studio – managed Code

The application programmer can develop the application in C# or VB.NET using the Compact Framework 3.5 which is part of the Windows CE Kernel for PicoCOM1.

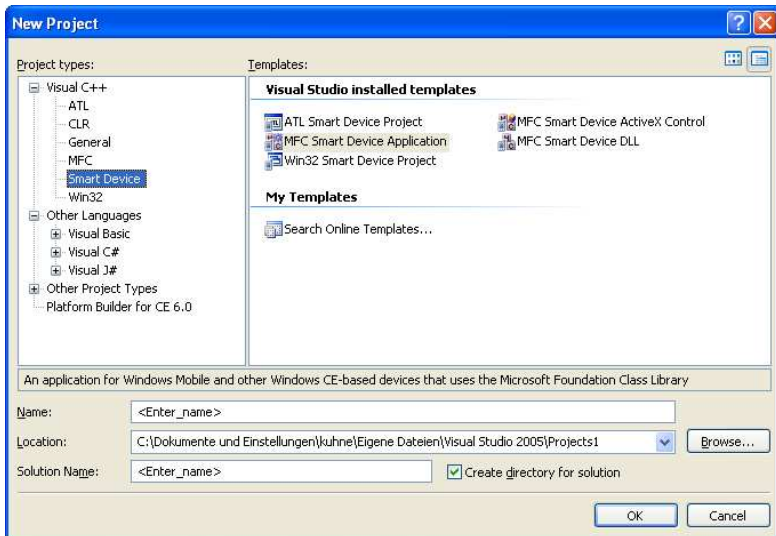
The next figure shows the project settings using managed code.



5.2 Visual Studio – native Code

The application programmer can develop the application in C++ using the PicoCOM1 SDK which can be downloaded from www.picocom.de/download.

The next figure shows the project settings using native code.



5.3 Visual Studio – debugging an device application

The application developer can debug an application via Ethernet or via USB – which is the best way. To debug via USB establish a connection between your development PC and PicoCOM1 with ActiveSync. As soon as the connection is set up you can start debug the application with breakpoints etc. as you know from applications for desktop PCs.

note:

When starting your application in Visual Studio with 'Start debugging' and you are getting memory problems on your PicoCOM1, please disable deploying the latest version of Compact Framework. Therefore select menu Project- Properties- Devices and deselect:

- Deploy the latest version of the .NET Compact Framework (including Service Packs)

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. F&S Elektronik Systeme assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained in this documentation. F&S Elektronik Systeme reserves the right to make changes in its products or product specifications or product documentation with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

F&S Elektronik Systeme makes no warranty or guarantee regarding the suitability of its products for any particular purpose, nor does F&S Elektronik Systeme assume any liability arising out of the documentation or use of any product and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Products are not designed, intended, or authorized for use as components in systems intended for applications intended to support or sustain life, or for any other application in which the failure of the product from F&S Elektronik Systeme could create a situation where personal injury or death may occur. Should the Buyer purchase or use a F&S Elektronik Systeme product for any such unintended or unauthorized application, the Buyer shall indemnify and hold F&S Elektronik Systeme and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that F&S Elektronik Systeme was negligent regarding the design or manufacture of said product.