

PicoMOD

First Steps

Version 1.03 Date: 2011-05-16

© by F & S Elektronik Systeme GmbH 2011

F&S Elektronik Systeme GmbH
Untere Waldplätze 23
D-70569 Stuttgart
Fon: +49(0)711-123722-0
Fax: +49(0)711-123722-99



History

Date	V	Platform	A,M,R	Chapter	Description	Au
2009-02-05	1.01		M	8	Info to use CF3.5 (VS2008)	DK
2010-05-14	1.02	PicoMOD6	A		PicoMOD6 added	HF
2011-05-16	1.03		M	1.1	Info about supply voltage of SKIT	KU

V Version
A,M,R Added, Modified, Removed
Au Author: CZ, DK, HF, HK, MK



Content

1	Getting started	1
1.1	Connecting basic peripheral devices	2
1.1.1	Connecting the Display	4
2	Powering-on PicoMOD	5
3	Bootup sequence	7
4	Bootloader	9
4.1	NBoot	9
4.2	Eboot	9
5	Firmware update on the PicoMOD	10
5.1	The NetDCU-USBLoader utility	10
5.2	Updating – Windows CE kernel image	11
5.3	Partitioning the Flash storage	12
5.4	Updating – EBOOT	13
5.5	Updating – NBOOT	13
6	Configuration	15
6.1	Using NDCUCFG utility	15
6.2	Network interface configuration	16
6.2.1	Network – General Facts	16
6.2.2	Network – Network address	16
6.2.3	Network – Set up with NDCUCFG	17
6.2.4	Network – Saving the parameters to registry	18
6.3	Touch panel calibration	18
7	Remote Tools (VS2005 / 2008)	20
8	Software Development	21
8.1	Visual Studio – managed Code	21
8.2	Visual Studio – native Code	22
8.2.1	Visual Studio – debugging an device application	22
9	Appendix	23
	Important Notice	23
	Listings	24
	Figures	24



1 Getting started

This documentation is a step by step introduction in how to use PicoMOD3,PicoMOD4 and PicoMOD6 with PicoMOD-Startinterface. The connectors referenced in this documentation are on the PicoMOD-Startintface. The documentation does NOT explain how to use PicoMODx without PicoMOD-Startintf. The latest version of this document can be found at:

<http://www.fs-net.de>

Additional support information can be found in our discussion forum at:

<http://forum.fs-net.de>



1.1 Connecting basic peripheral devices

Next picture shows the PicoMOD-Starterkitf with the position and description of connectors.

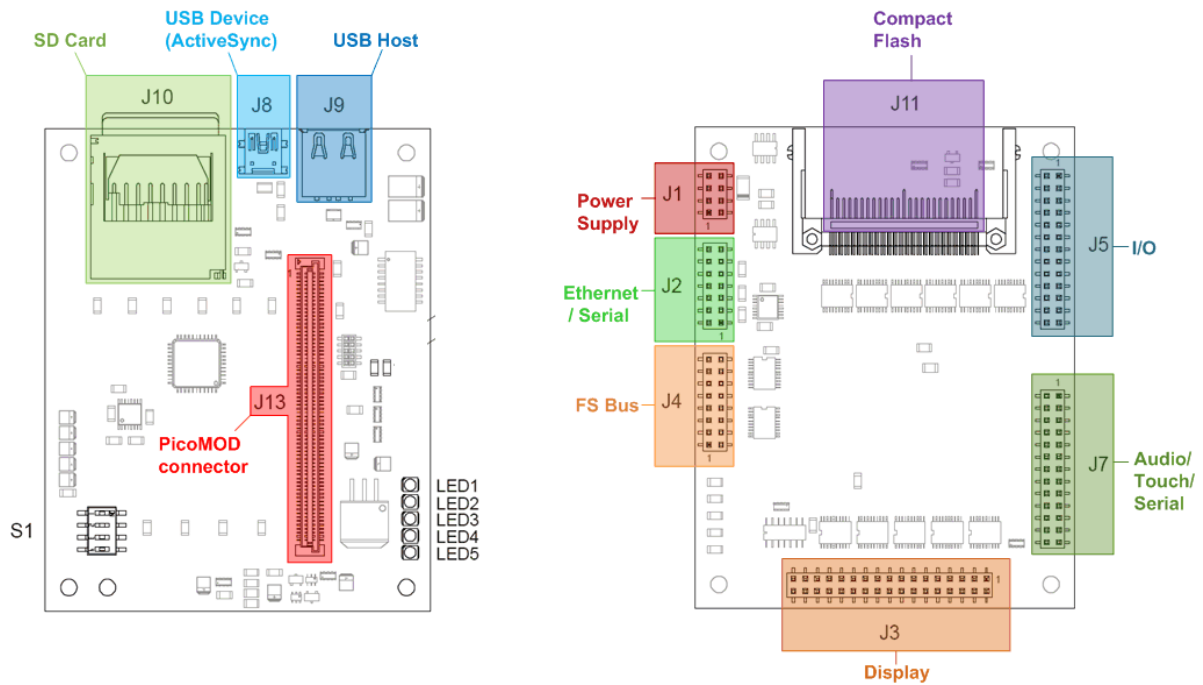


Figure 1: PicoMOD-Starterkitf without PicoMOD module

All required cables and adapters are arranged to the Starterkit package. When connecting these cables **please take account of the pin1 marker on the cable and the connector.**

For power supply one single 5V source is required. The VCFL source is required for LC Displays with CFL Backlight.

More information about the PicoMOD carrier board can be found in the PicoMOD-Starterkitf documentation on our website.



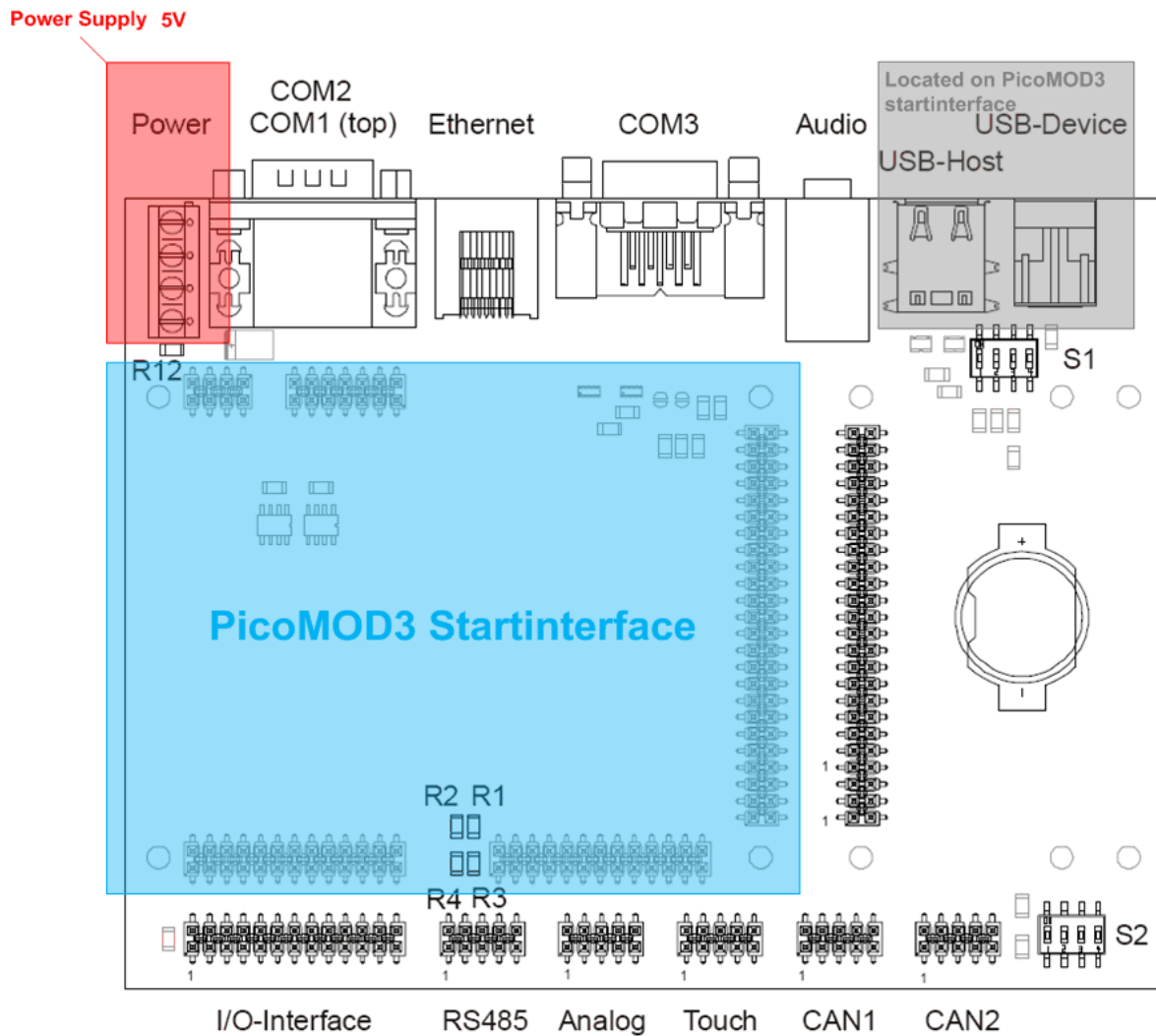


Figure 2: NetDCU-Startintf without PicoMOD-Startintf

The PicoMOD Starterkit is shipped with the NetDCU-Startintf. The PicoMOD can connect to this baseboard via the PicoMOD3-Startintf. Figure 2 shows the different interface-ports provided by the NetDCU-Startintf. Please note that the USB ports on the NetDCU-Startintf are not used. The USB ports (Host and Device) are located on the PicoMOD-Startintf.



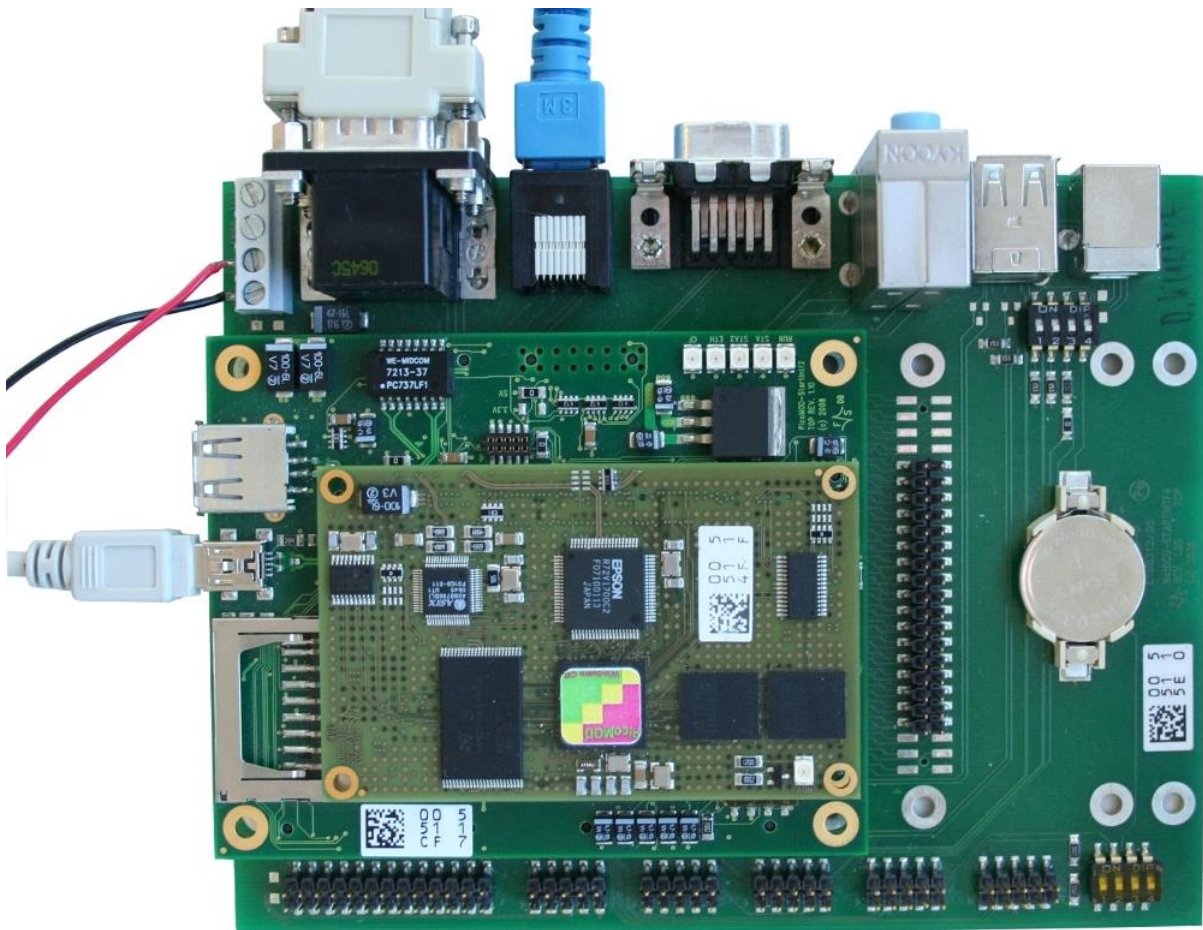


Figure 3: PicoMOD-Startintf connected to NetDCU-Startintf

1.1.1 Connecting the Display

To connect a display in most cases an adapter board is required. You can find the pin assignment for the display interface J3A in the PicoMOD-Startintf documentation.

2 Powering-on PicoMOD

Before you power on PicoMOD, you should make a serial connection between PicoMOD and your PC. Please use the cables shipped with the PicoMOD Starterkit to connect your PC to the Debug-Port of the PicoMOD (COM1, the upper port of the dual port serial connector). On the PC, you should have installed DCUTerm.exe as terminal program, which is available in the password protected download area of the PicoMOD.

Follow the steps below, to make a connection:

- Install DCUTerm.exe on your PC
- Configure DCUTerm.exe as shown in the following picture (the COM-Port may differ on your computer):

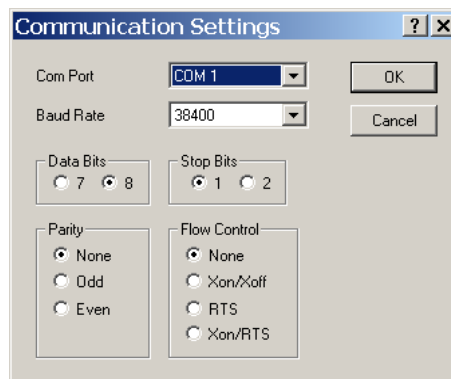


Figure 4: DCUTerm.exe configuration: CommPort → Settings

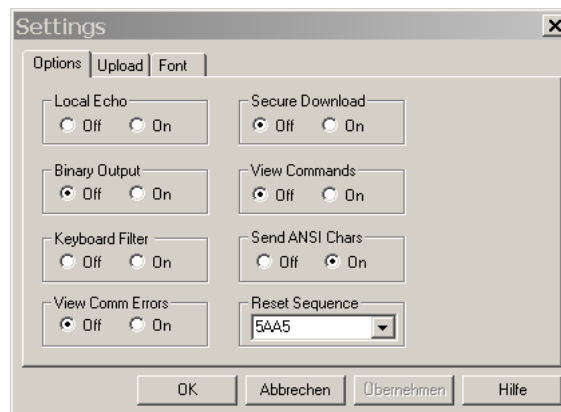


Figure 5: DCUTerm.exe configuration: View → Options



- Press the connect button in DCUTerm.exe

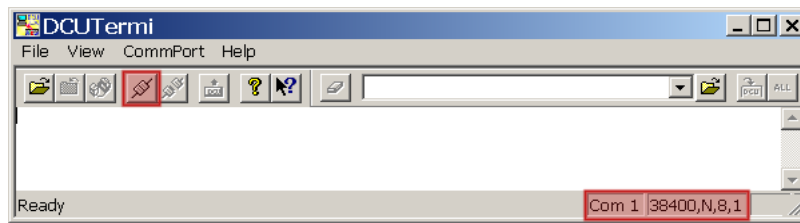


Figure 6: Opening the serial connection in DCUTerm.exe

- Use serial cable shipped with PicoMOD Starterkit to make a connection between COM1 and your PC.
- Make a cable between power connector of PicoMOD Startintf and your power supply. Usually you need to connect ground and +5V (2A). Please refer R16 / R17 (Figure 1) for 3.3V or 5V power supply.



3 Bootup sequence

After you performed the steps from Chapter 2 the LED #5 (RUN) should be illuminated and you should see output similar to the following in the terminal window of DCUTerm.exe:

Bootloader:

```
Microsoft Windows CE Ethernet Bootloader Common Library Version 1.1 Built Sep 5 2008
16:56:11
Microsoft Windows CE Bootloader for PicoMOD Built Sep 11 2008
Portions copyright (c) 2007 F&S Elektronik Systeme GmbH
Boot Loader, Version 1.38
StepStone Loader, Version P314
```

Listing 1: Bootup: Bootloader

Flash memory initialization:

```
FMD: NAND ID:0x2076 --> ST: 512MBit (64MB)
FMD: Block 0 to 18 are locked for boot loader
Create partition for boot section ... Success
System ready!
Preparing for download...
Press >S< to step into monitor...
```

Listing 2: Bootup: Flash initialization

Display initialization:

```
InitLCD...
Display-Mode: 0, Name VGA standard display, 640x480, bpp=16, Voltage=3.3V
Width: 640 Height: 480 Bpp: 16
InitializeDisplayHardware Complete
InitLCD done
```

Listing 3: Bootup: Display initialization

Starting Windows CE Kernel:

```
AUTO BOOT enabled
Kernel read from flash disk started finished
INFO: OEMLaunch: Jumping to Physical Address 0x30141E2Ch (Virtual Address 0x0h)...

Windows CE Kernel for ARM (Thumb Enabled) Built on Apr 16 2008 at 14:31:59

PicoMOD3 V0.12 - Firmware Init
Copyright (c) 2008 F&S Elektronik Systeme GmbH
Build: Sep 2 2008/18:26:09
[...]
```

Listing 4: Bootup: Starting WinCE Kernel

Loading Windows CE Drivers:

```
DIO: Version 1.5, ActiveKey = Drivers\Active\29
USBFN: Version 1.1, ActiveKey = Drivers\Active\30
[...]
```

Listing 5: Bootup: Loading drivers

Starting NDCUCFG.EXE:

```
NDCUCFG V: 036 started. Platform: PicoMOD3
CreateFile() failed -> ERROR COM1:
```

Listing 6: Bootup: Starting ndcucfg.exe

At this point the Windows CE Kernel has loaded all device drivers and is running. As you can see at the last two lines of the bootup debug output, NDCUCFG.exe is getting autostarted and ends with an



error message. This is because the debug output on COM1: is enabled and NDCUCFG.exe could not open that serial port. You can change the settings for debug output (chapter 4.2) and the settings for NDCUCFG.exe (chapter 6) for your needs.

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

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



4 Bootloader

The start up process of PicoMOD is divided into three steps:

- **Nboot** (Stepping-Stone bootloader)
 - Responsible for low level initialization tasks.
 - Loads the WindowsCE bootloader
- **Eboot** (WindowsCE bootloader)
 - Loads the WindowsCE kernel image
- **WindowsCE kernel image**
 - WindowsCE operating system and all drivers for the PicoMOD.
 - Offers you the possibility to develop and debug custom applications.

Both bootloaders (Nboot and Eboot) are equipped with a small configuration menu, that is accessible via serial debug port (COM1).

To open one of these boot menus the following characters must be entered **while** booting the PicoCOM.

- Nboot: 's'
- Eboot: <SHIFT>+'s' (capital 's')

4.1 NBoot

The Nboot bootloader should only be used in emergencies. For that reason it does not offer a huge amount of commands and will not be described in more detail in this documentation.

4.2 Eboot

By contrast the Eboot bootloader supports a wide range of commands that are used for example to setup the partitioning of flash memory, erase the Registry, enable or disable the debug output, A detailed overview of these features will not be part of this documentation, but the command '?' will give you more information about all available commands.

Note:

Operating with the Nboot and Eboot menu will give you the possibility to erase the complete flash. Please always keep in mind that all data stored in flash and even the Eboot itself could get lost.



5 Firmware update on the PicoMOD

All three firmware components of the PicoMOD, described in the chapter before, can be updated separately. The following chapter will describe these operations in more detail.

5.1 The NetDCU-USBLoader utility

The preferred method to update PicoMOD is the using the NetDCU-USBLoader which offers the possibility to download the bootloaders and the WindowsCE kernel to PicoMOD via USB. The NetDCU-USBLoader can be found in the PicoMOD download area on our website.

When connecting PicoMOD and NetDCU-USBLoader for the very first time (see chapter 3.2) you have to install an USB driver on your development PC. That driver is shipped with NetDCU-USBLoader installer and gets copied to its installation directory. The procedure of downloading a new bootloader or a Windows CE kernel with this utility is described in chapter 5.

Installing the driver on your development PC:

When trying to download a bootloader or kernel image 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`.

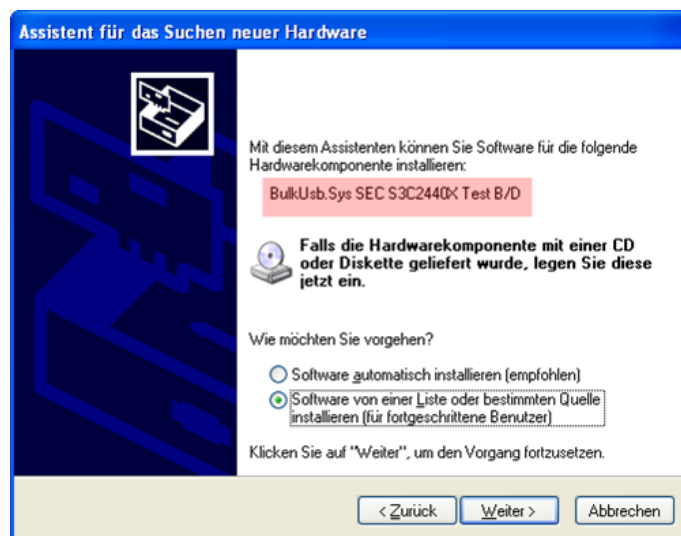


Figure 7: NetDCU-USBLoader driver installation (1)

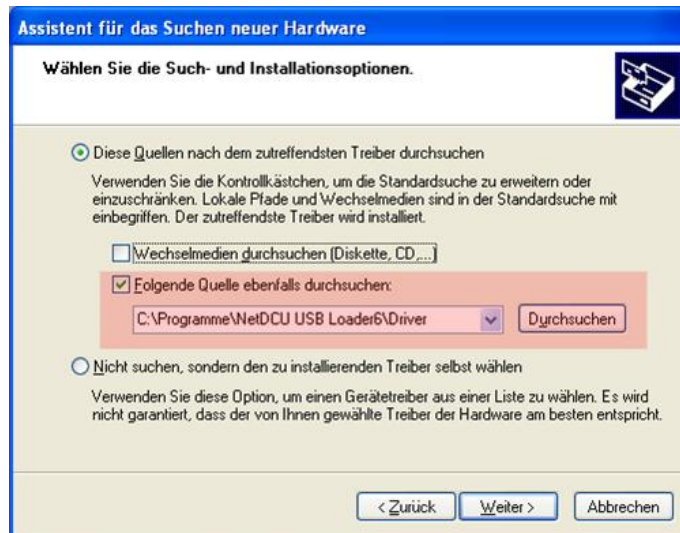


Figure 8: NetDCU-USBLoader driver installation (2)

5.2 Updating – Windows CE kernel image

You can update the Windows CE kernel via Ethernet or by using the NetDCU-USBLoader utility via USB which is the preferred method. Therefore you must enter the WindowsCE Bootloader (Eboot) first by pressing **<SHIFT> + 's'** while powering on the PicoMOD.

Download via USB:

To download the WindowsCE Kernel by USB use the command **'DU'** and start the NetDCU-USBLoader utility on your desktop PC. As soon as the connection is established the button in the top right corner of NetDCU-USBLoader turns from red to green. Select the respective **'<NK-kernel-image>.bin'** file and click on **'Start'**.

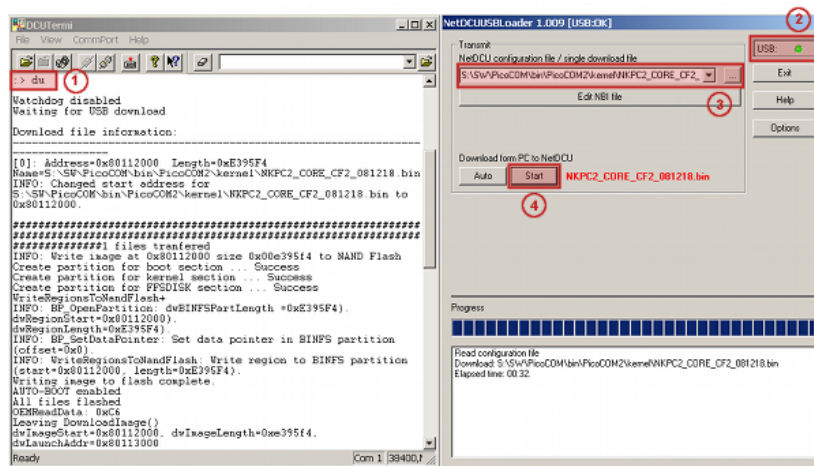


Figure 9: Using the NetDCU-USBLoader utility

After the download of the Windows CE kernel image is finished and the kernel is started, the Windows desktop should appear on the connected display (and ActiveSync should open a connection- if the PicoMOD is connected via the USB device port to your development PC).



Note:

The Windows CE kernel can be downloaded to RAM or to Flash.

This can be configured by the Eboot commands:

'MR' - Store kernel in RAM memory

'MF' - Store kernel on Flash disk

If the Windows CE kernel does not start automatically after reboot you have to execute the following Eboot command:

'L1' - Launch previously stored kernel after boot

5.3 Partitioning the Flash storage

There are up to 3 partitions possible on PicoMOD.

- **OS-Image (BINFS):** The WindowsCE kernel is stored in this partition.
- **FFSDISK:** This partitions can be used to store user data and applications. It is available under '\FFSDISK' on a running WindowsCE 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 Storage-Manager. The size of this partition might be 0 in most cases.

Partitioning the flash memory must be performed within the Eboot menu. With the command '?' you will get a list of all available Eboot commands. Partitioning is setup with the with command 'P'.

```
:> P
-----PARTITION CONFIGURATION-----
Current settings:
Total      : 64 MB
OS-Image: 35 MB
FFSDISK   : 29 MB, Part type: FAT
SECOND    : 0 MB, Part type: EXTENDED
Enter maximal size for OS-Image [35]:
```

Listing 7: Partitioning the Flash memory

At this point you may resize the partitions for BINFS, FFSDISK and the EXTENDED partition. Follow the instructions printed in the terminal program. After confirming the updated partition settings you should see output similar to the printed below.

```
WindowsCE image and all data in FFSDISK will be erased.
Continue ? (Y/n) YFMD: block 0 is locked !
FMD: Can't erase block 0x0
FMD: block 1 is locked !
FMD: Can't erase block 0x1
FMD: block 2 is locked !
FMD: Can't erase block 0x2
[...]
Success
Create partition for kernel section ... FindFreeSector: FreeSector is: 0x12b after processing
part 0x20.
Success
Create partition for FFSDISK section ... FindFreeSector: FreeSector is: 0x12b after
processing part 0x20.
FindFreeSector: FreeSector is: 0x12141 after processing part 0x21.
CreatePartition: Num sectors set to 0xdae0 to allow for compaction blocks.
Success
```



```
WARNING: Boot configuration signature invalid - choosing defaults...
:>
```

Listing 8: Flash partitioning process

You can ignore the FMD messages. Some blocks are reserved for storing the bootloader. You should not worry about these messages.

Download by Ethernet:

In addition to using the NetDCU-USBLoader to update the WindowsCE kernel image via USB the Microsoft utility called Eshell can be used to download the kernel via Ethernet. For this purpose press 'DE' within the Eboot and start the Eshell utility on your desktop PC. As soon as PicoMOD sends a network broadcast ('Sent BOOTME to 255.255.255.255') you should be able to select the device in the 'Select Device' dialog of Eshell. Then choose 'Select image' from the File-Menu of Eshell and select the respective '<NK-kernel-image>.bin' file. Press 'open' – this starts the download to PicoMOD.

5.4 Updating – EBOOT

Updating Eboot is done similar to updating the WindowsCE Kernel with the NetDCU-USBLoader utility. Enter the menu of the currently installed WindowsCE bootloader by pressing **<SHIFT>+s'** while powering on the PicoMOD. To download the new Eboot (eboot.nb0) press **'DU'** and start the NetDCU-USBLoader utility on your desktop PC. In NetDCU-USBLoader select the respective <eboot>.nb0 file and click on 'Start'.

Updating Eboot over the serial debug line

You can also download the Eboot bootloader via the serial debug port. Therefore enter the Nboot menu ('s') and press 'd' to start serial download. In File-Menu of the DCUTermi utility choose 'Transmit binary file', select the respective <Eboot>.nb0 file and press 'open'.

After the download has finished press 'f' to save.

Note:

Serial downloads are very error-prone. For this reason please try avoiding this method.

To make this update variant more robust a checksum will be calculated and displayed after downloading has finished. You can verify this checksum with a small utility (bootloader-checksum) that can be downloaded from the PicoCOM download area.

5.5 Updating – NBOOT

Updating the Nboot bootloader can also be accomplished with the NetDCU-USBLoader. The only difference is that the file type of the Nboot image may not be recognized by the USBLoader automatically. For this reason the NBI file must be edited manually after selecting the Nboot file (button 'Edit NBI file' in NetDCU-USBLoader). In case of updating Nboot the NBI file must have the following body:

```
author = "F&S Elektronik Systeme GmbH"
[Files]
StepStone=<PATH>\<NBoot>.bin
```

Listing 9: Updating NBOOT



By default the Nboot file will be detected as a Windows CE kernel image. After changing that definition in the file section from 'kernel' to 'StepStone' click on 'Start' button in the NetDCU-USBLoader to perform the update.

Note:

A detailed description of the purpose and additional features offered by a NBI file can be found in the documentation of the NetDCU-USBLoader utility



6 Configuration

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

Most of changes to PicoMOD device is done through NDCUCFG utility and stored in persistent system registry, taking effect after next reboot of PicoMOD device.

According to device's software architecture, this utility is automatically started on COM1. As well, the utility can be remotely executed over a Telnet connection, once you have got network access to PicoMOD device. DHCP is enabled by default.

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

In chapter 6.2 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 utility.

6.1 Using NDCUCFG utility

You can either enable debug messages or the NDCUCFG utility on COM1. By default debug messages will run on COM1. To select the usage of this serial port you must enter the Eboot menu. With the command 'O' you can enable or disable the serial debug output during boot'. If you choose 'Y' the NDCUCFG utility gets started on COM1. Powering on the PicoMOD will then output on COM1:

```
[...]  
INFO: OEMLaunch: Jumping to Physical Address 0x30141E2Ch (Virtual Address 0x0h)...  
  
NetDCU Config Utility Ready  
Version: 036  
Type help for commands  
  
!>
```

Listing 10: Starting NDCUCFG on debug port

If this command prompt (!>) appears in the terminal program you are ready to pass commands to NDCUCFG utility. Otherwise something went wrong. Please check various parameters described in chapter 2 Powering-on PicoMOD.

If NDCUCFG is running successfully over the serial line 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:

```
[...]  
backlight off  
start <file name>  
quit  
help  
help <command>  
!>
```

Listing 11: NDCUCFG: Listing commands

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 command and then press Enter:



```
!>help backlight
```

Listing 12: NDCUCFG: using commands

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
```

Listing 13: NDCUCFG: using commands #2

and you will get satisfying answer to you *help*-request.
To save the changes execute the command:

```
!>reg save
```

Listing 14: NDCUCFG: Saving changes permanently

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

6.2 Network interface configuration

PicoMOD 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 PicoMOD meets 802.3 10BaseT specifications by IEEE, and provides safe data transfer on speeds up to 100 Mbit/sec.

6.2.1 Network – General Facts

Being integrated into IP-network, in order to get directly referred by other network devices, every PicoMOD 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 PicoMOD. 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 accordance to corresponding IEEE's regulations, is unique for every network device across the World.

6.2.2 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
```

Listing 15: IP-configuration from command line



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

Windows IP configuration:

Ethernet Adapter AX887961:
IP address: 192.168.0.100
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.

6.2.3 Network – Set up with NDCUCFG

Almost all device settings can be configured by registry.

Therefore you use the *reg* commands of the NDCUCFG utility as described below:

```
!>help reg
reg open
reg open <key>
reg opencu <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
!>
```

Listing 16: NDCUCFG: Registry commands

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

So execute the command:

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

Listing 17: NDCUCFG: Opening TcpIp settings

to access the network parameters. The output *ok* tells you that NDCUCFG 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
!>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

```

Listing 18: NDCUCFG: Changing TcpIp settings

6.2.4 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 PicoMOD device.

Enter following:

```
!>reg save
```

Listing 19: NDCUCFG: Save modifications permanently

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 PicoMOD system during current session of working with NDCUCFG utility are stored in persistent registry.

6.3 Touch panel calibration

The TX09 display available with the starterkit is equipped with a 4 wire touch panel. At delivery status this touch panel should already be calibrated satisfyingly. But when using another display or the touch panel is not working properly it might be required to calibrate the touch panel driver again. This is performed using NDCUCFG.

The desired command is “touch calibrate”. After executing this command the touch panel calibrating routine should be displayed on the LCD screen.

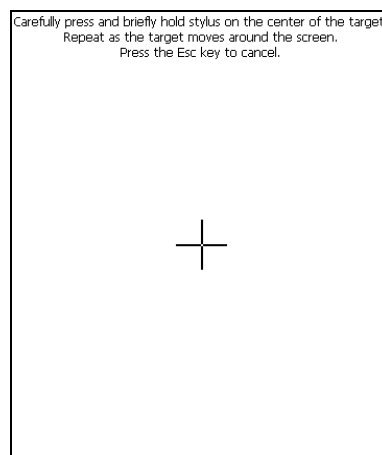


Figure 10: Initial screen of the touch panel calibration program



After holding the stylus on the cross in the center of the screen the cross will move around to each corner where it must be “touched” again. When calibration is finished the cross disappears and you are requested to press the Enter key. If there is no keyboard available you can simply hold the stylus anywhere on the screen to close the calibration program. The regular Windows desktop should appear again and calibration is finished. As the calibration settings are stored in registry the registry must be saved to accept the new settings permanently (“reg save” in NDCUCFG). Please note that calibration settings will be reset every time the registry is reset.



7 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



8 Software Development

For software development you have to use Visual Studio 2005 or newer. When programming for .NET Compact Framework 3.5 you need to have Visual Studio 2008 installed. The kernel-image that you can download from the web side www.pico.com.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 (you can use VS2005 for native code). To use native code you need to install the PicoMOD SDK that you also can download from the above mentioned web side.

To connect Visual Studio to PicoMOD for software development you can use a USB device connection and Ethernet (or only Ethernet; check the Support-Forum for details) connection.

A detailed description how to connect via Ethernet is given by “Connect VS 2005 to NetDCU using LAN” which is analog to PicoMOD 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 using Microsoft ActiveSync. For this install the latest version of ActiveSync on your PC (download ActiveSync from <http://www.microsoft.com>) and connect PicoMOD and PC using the USB device connector shipped with the PicoMOD-Startint. The connection is established automatically.

8.1 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 WindowsCE kernel for PicoMOD.

Note:

To write code for / with the CF3.5 you need VS2008 be installed on your development pc.

The next figure shows the project settings using managed code.

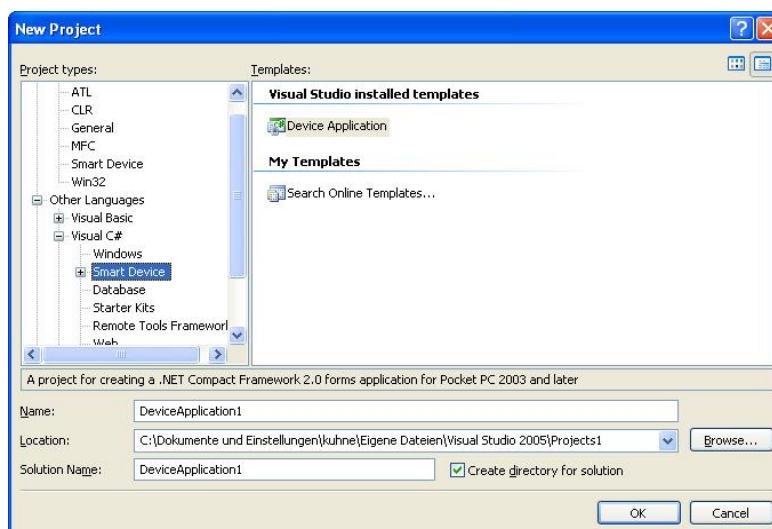


Figure 11: Project settings for managed code



8.2 Visual Studio – native Code

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

The next figure shows the project settings using native code.

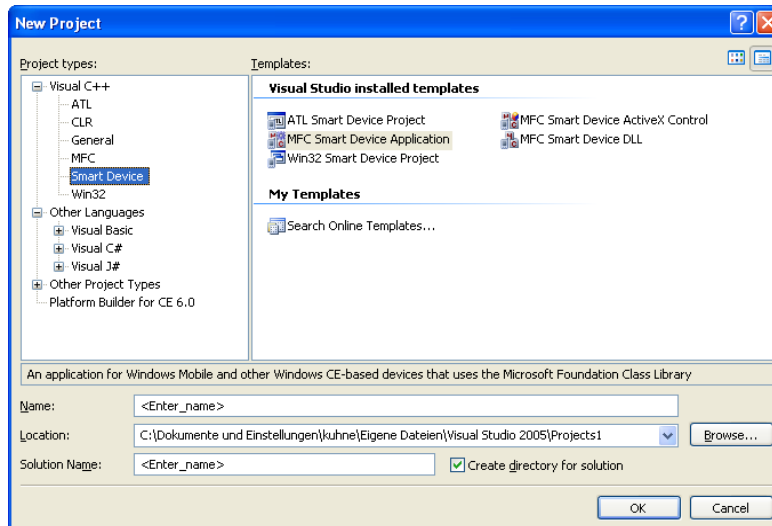


Figure 12: Project settings for native code

8.2.1 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 PicoMOD 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 PicoMOD, 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)



9 Appendix

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.



Listings

Listing 1: Bootup: Bootloader.....	7
Listing 2: Bootup: Flash initialization.....	7
Listing 3: Bootup: Display initialization.....	7
Listing 4: Bootup: Starting WinCE Kernel.....	7
Listing 5: Bootup: Loading drivers	7
Listing 6: Bootup: Starting ndcucfg.exe	7
Listing 7: Partitioning the Flash memory.....	12
Listing 8: Flash partitioning process	13
Listing 9: Updating NBOOT	13
Listing 10: Starting NDCUCFG on debug port.....	15
Listing 11: NDCUCFG: Listing commands	15
Listing 12: NDCUCFG: using commands	16
Listing 13: NDCUCFG: using commands #2.....	16
Listing 14: NDCUCFG: Saving changes permanently	16
Listing 15: IP-configuration from command line	16
Listing 16: NDCUCFG: Registry commands	17
Listing 17: NDCUCFG: Opening Tcplp settings	17
Listing 18: NDCUCFG: Changing Tcplp settings	18
Listing 19: NDCUCFG: Save modifications permanently	18

Figures

Figure 1: PicoMOD-Startintf without PicoMOD module.....	2
Figure 2: NetDCU-Startintf without PicoMOD-Startintf	3
Figure 3: PicoMOD-Startintf connected to NetDCU-Startintf.....	4
Figure 4: DCUTerm.exe configuration: CommPort → Settings.....	5
Figure 5: DCUTerm.exe configuration: View → Options	5
Figure 6: Opening the serial connection in DCUTerm.exe	6
Figure 7: NetDCU-USBLoader driver installation (1)	10
Figure 8: NetDCU-USBLoader driver installation (2)	11
Figure 9: Using the NetDCU-USBLoader utility	11
Figure 10: Initial screen of the touch panel calibration program	18
Figure 11: Project settings for managed code.....	21
Figure 12: Project settings for native code	22

