# First Steps documentation

Windows CE/Compact 7

*for armStoneA8*

Version 1.0
Date: 2012-04-10

armStoneA8

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

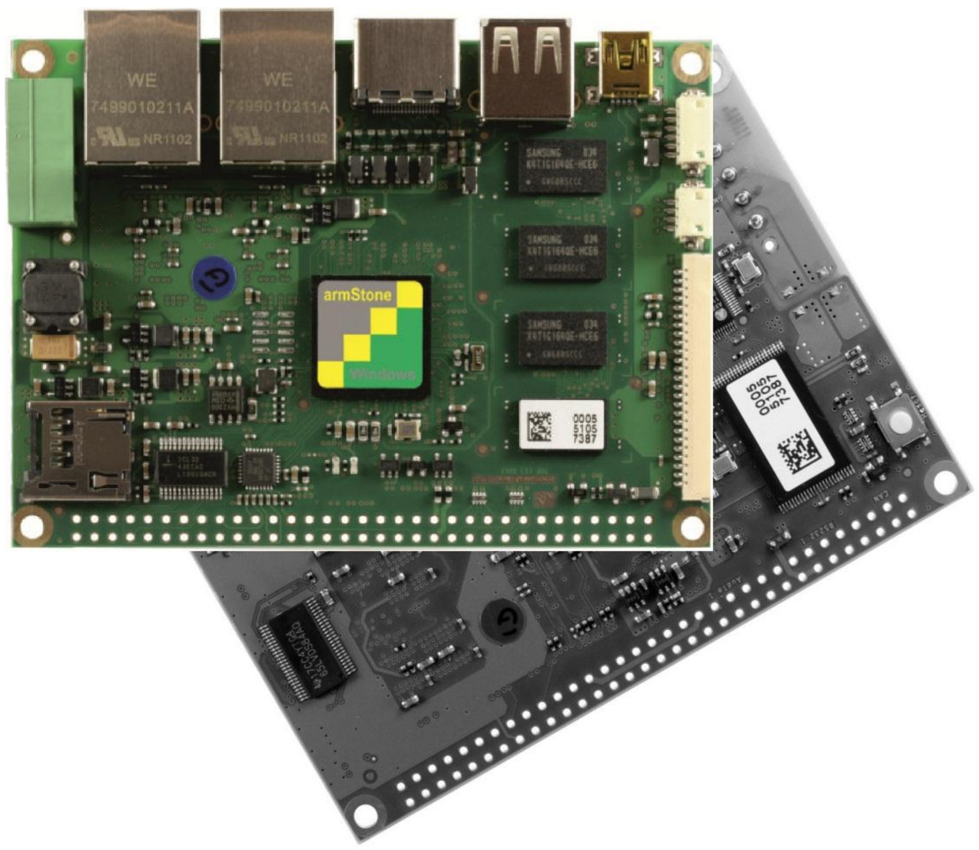
# History

| Date | V | Platform | A,M,R | Chapter | Description | Au |
|---|---|---|---|---|---|---|
| 2012-04-10 | 1.0 | armStoneA8 | A | * | First Version | HF |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| | |
|---|---|
| V | Version |
| A,M,R | Added, Modified, Removed |
| Au | Author |

## About this document

The following document describes the usage and handling of armStoneA8. armStoneA8 is a single board computer and therefore don't need a carrier board. You will learn how to power on the device, make a connection via serial, Ethernet or USB and how to update boot loader or Windows CE/Windows Compact image.



# Table of Contents

# 1 Getting started

This documentation is a step by step introduction in how to use armStoneA8 pico-ITX single board computer. 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 armStoneA8 with the position and description of connectors. The single board computer armStoneA8 adopts VIA's pico-ITX format. The board measures 10 x 7,2 cm (3,94 x 2,84 inches). You can find schematic for the carrier board in the download area.

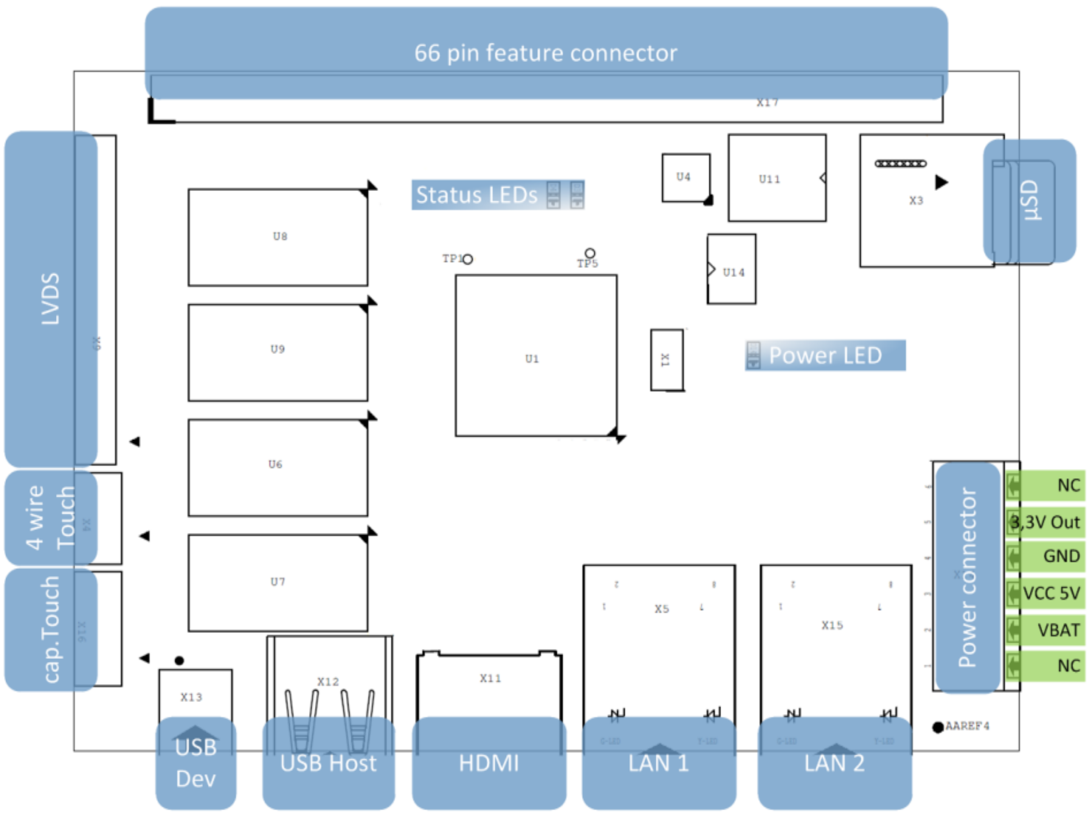
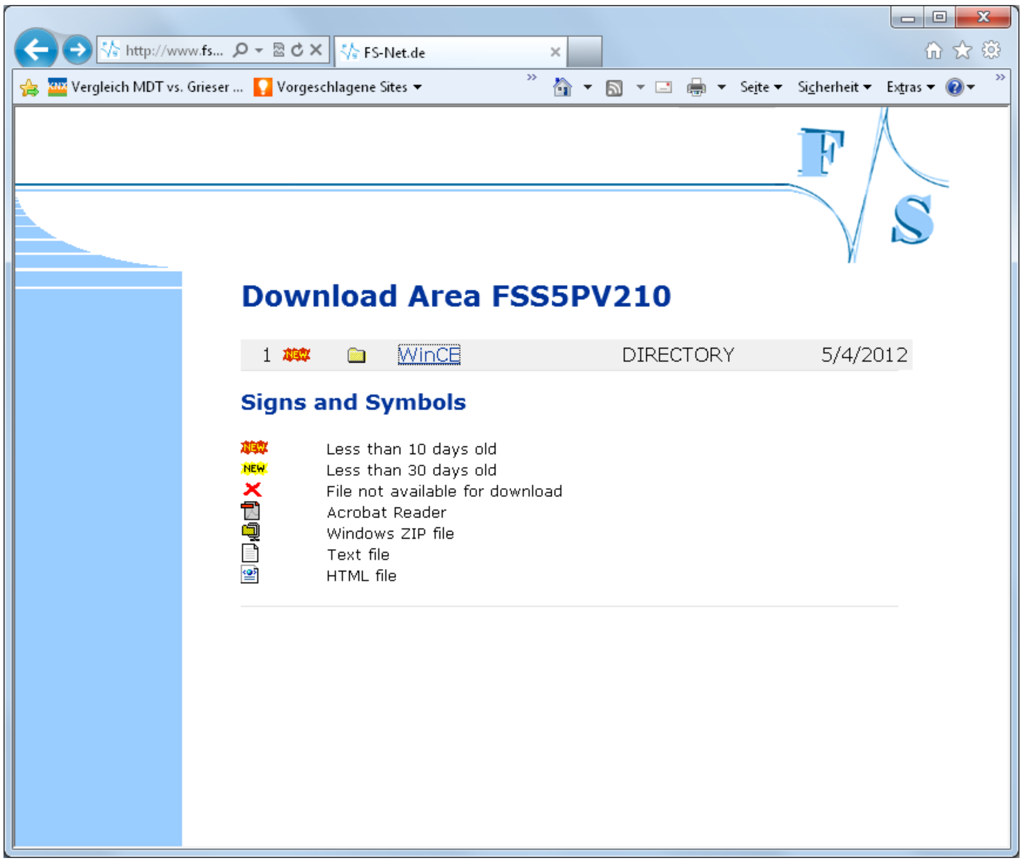


*Figure 1:armStoneA8 pico-ITX interfaces (top side)*

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**.

More information about the armStoneA8 can be found in the armStoneA8 hardware documentation on our website.

# 2 Download Area



## Download Area FSS5PV210

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 1 | NEW | 📁 | WinCE | DIRECTORY | 5/4/2012 |

### Signs and Symbols

|  |  |
|---|---|
| NEW | Less than 10 days old |
| NEW | Less than 30 days old |
| ✕ | File not available for download |
| 📄 | Acrobat Reader |
| 📄 | Windows ZIP file |
| 📄 | Text file |
| 📄 | HTML file |

*Figure 2: Download area*

# 3 Powering-on armStoneA8

Before you power on armStoneA8, you should make a serial connection between armStoneA8 and your PC. Please use the cables shipped with the armStoneA8 Starterkit to connect your PC to the Debug-Port of the armStoneA8. The Debug-Port resides on the 66 pin feature connector. (see Figure 1). In hardware revision 1.10 the pins are located at a 5 pin single row connector which is mounted at pin 53, 55, 57 ,59 and 61. That allows attaching a standard 9pin to DSUB9 adapter cable for debug output of boot loader and kernel with TX and RX to a terminal. Using RTS and CTS too needs a special cable adapter.
Pin 1 of this adapter should connect on pin 62 of the 66 pin connector. The second 4 pin row of the adapter cable is not in use.
**To make using of these signals easier this pin out will be changed in next hardware revision.**

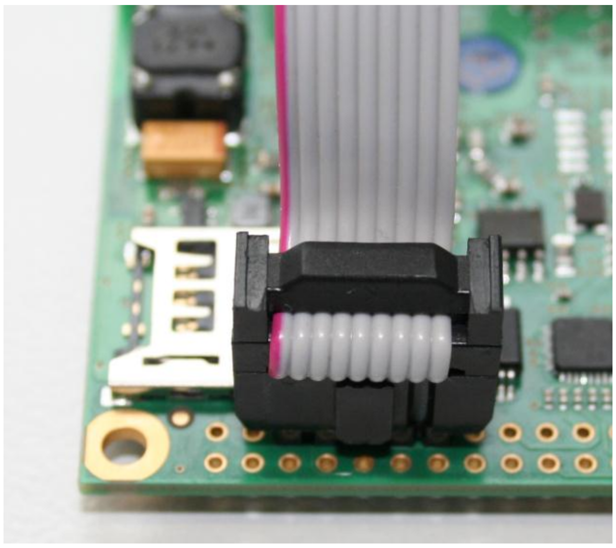
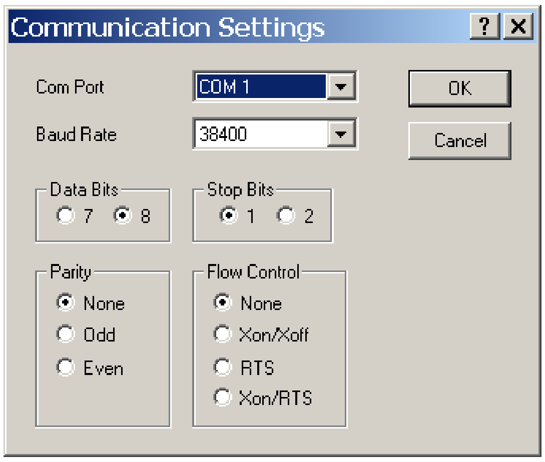


*Figure 3: RS232 adapter mounting*

On the PC, you should have installed DCUTerm.exe as terminal program, which is included in the SDK. The SDK is available in the password protected download area of the armStoneA8.

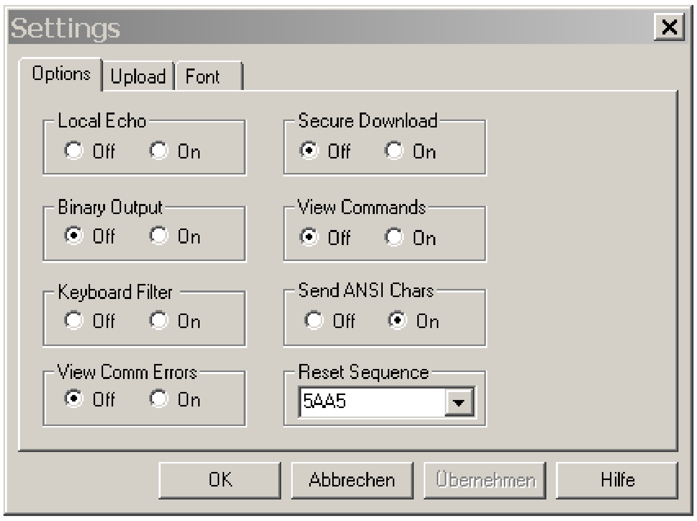
Follow the steps below, to make a connection:

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



Com Port: COM 1
Baud Rate: 38400
Data Bits: 7 [✓ 8]
Stop Bits: [✓ 1] 2
Parity: [✓ None] Odd Even
Flow Control: [✓ None] Xon/Xoff RTS Xon/RTS

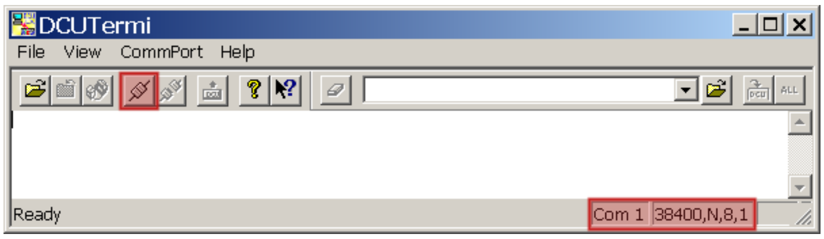
*Figure 4: DCUTerm.exe configuration: CommPort → Settings*



Local Echo: Off On
Secure Download: [✓ Off] On
Binary Output: [✓ Off] On
View Commands: [✓ Off] On
Keyboard Filter: Off On
Send ANSI Chars: Off [✓ On]
View Comm Errors: [✓ Off] On
Reset Sequence: 5AA5

*Figure 5: DCUTerm.exe configuration: View → Options*

- Press the connect button in DCUTerm.exe



Ready
Com 1 38400,N,8,1

*Figure 6: Opening the serial connection in DCUTerm.exe*

- Use serial cable shipped with armStoneA8 Starterkit to make a connection between COM1 and your PC.
- Make a cable between power connector of armStoneA8 (see Figure 1) and your power supply. Usually you need to connect ground and +5V (2A).
- Plug Ethernet connector (LAN1) and a serial cable to Debug Port connector.

# 4 Bootup sequence

As F&S delivers armStoneA8 with pre-installed bootloaders and kernel image you should see debug output on COM1 like here:

Bootloader:

```
Microsoft Windows CE Ethernet Bootloader Common Library Version 1.1
Built Mar 21 2012 14:02:58
Microsoft Windows CE Bootloader for armStoneA8 Built Mar 29 2012
Portions copyright (c) 2012 F&S Elektronik Systeme GmbH
Boot Loader, Version 1.6
NBoot, Version VN12
```

*Listing 1: Bootup: Bootloader*

Read kernel image from NAND flash:

```
Kernel (39089kB) read from flash disk started  finished in 4027
milliseconds
INFO: OEMLaunch: Jumping to Physical Address 0x4002B050h (Virtual
Address 0x0h)...
```

*Listing 2: Bootup: Read image from NAND flash*

Starting kernel image:

```
Windows CE Kernel for ARM (Thumb Enabled) Built on Sep 14 2011 at
17:32:03

armStoneA8 V0.01 - Firmware Init
Copyright (c) 2012 F&S Elektronik Systeme GmbH
Build: Mar 29 2012/15:19:48
```

*Listing 3: Bootup: Start Windows CE*

Loading device drivers:

```
I2C: Version 3.2, ActiveKey = Drivers\Active\04
PSS: Version 1.0, ActiveKey = Drivers\Active\13
AX88796: Version 01.05, ActiveKey =
BE2: Version 1.1, ActiveKey = Drivers\Active\22
I2C: Version 3.2, ActiveKey = Drivers\Active\23
WAV: Version 2.1, ActiveKey = Drivers\Active\25
UART: Version 1.3, Key = Drivers\Active\26
UART: Version 1.3, Key = Drivers\Active\27
CMM: Version 1.2, ActiveKey = Drivers\Active\28
SDMMCCh0: Version 1.5, ActiveKey = Drivers\Active\29
[…]
```

*Listing 4: Bootup: Loading drivers*

Start NDCUCFG application (read chapter *Using NDCUCFG utility* for detail information):

```
NDCUCFG V: 045 started. Platform: armStoneA8
```

*Listing 5: Bootup: Flash initialization*

*Note:*

Debug output can be enabled/disabled by EBoot command ‘O‘.

# 5 Bootloader

The startup process of armStoneA8 is divided into three steps:

- **NBoot** (Stepping-Stone bootloader)
    - Responsible for low level initialization tasks.
    - Loads the Windows CE bootloader

- **EBoot** (Windows CE bootloader)
    - Loads the Windows CE kernel image

- **Windows CE kernel image**
    - Windows CE operating system and all drivers for the armStoneA8.
    - 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 device.

- NBoot: **‘s’**
- EBoot: **<SHIFT>**+**‘s’** (capital „s“)

*Note:*

Details on updating NBoot, EBoot and WINCE kernel image can be found in chapter *Updating Firmware*.

# 6 Configure Windows Embedded Compact 7 Image

Configuration of the armStone device is provided by different means. The 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.

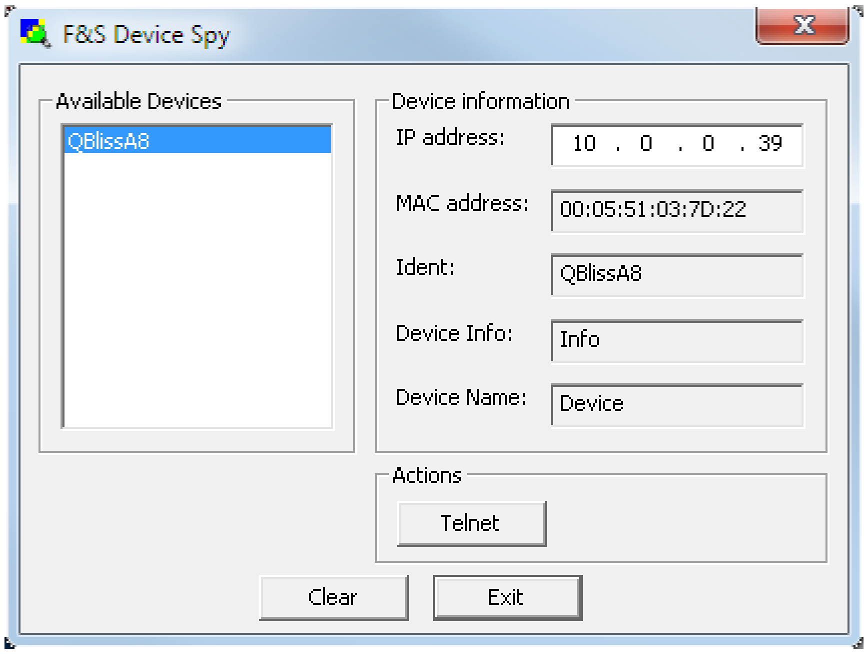
Most of changes to armStone device is done through NDCUCFG utility and stored in persistent system registry, taking effect after next reboot of the device.

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 the device. DHCP is enabled by default.

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

## 6.1 The FSDeviceSpy Utility

There are different ways to make a connection between your PC and armStone. One of them is a Telnet connection using Ethernet as physical transport. To make this as easy as possible, F&S has developed the utility FSDeviceSpy. FS DeviceSpy is included in SDK. After boot, armStone sends broadcast packet with some special information. FSDeviceSpy is waiting for this packet and adds the recognized device in the list of devices. After selecting the device from the device-list just press the Telnet button to make a connection.

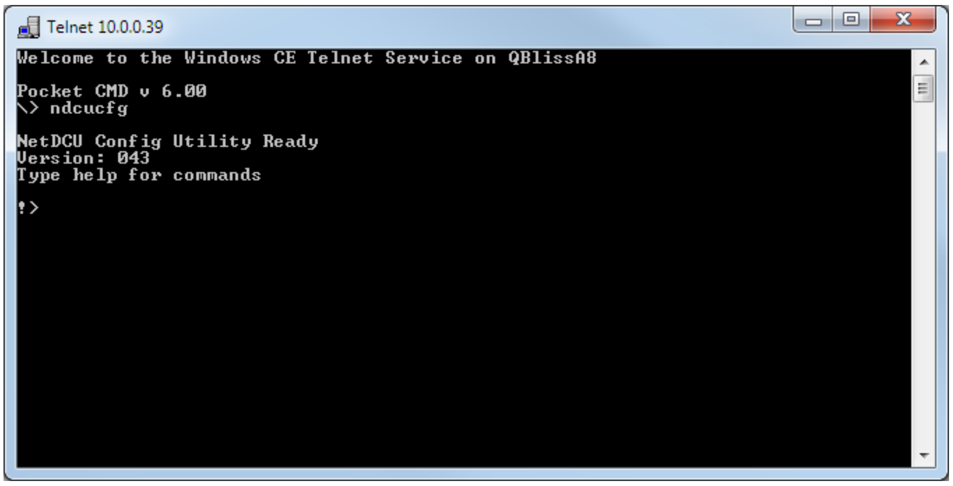


F&S Device Spy

Available Devices
QBlissA8

Device information
IP address: 10 . 0 . 0 . 39
MAC address: 00:05:51:03:7D:22
Ident: QBlissA8
Device Info: Info
Device Name: Device

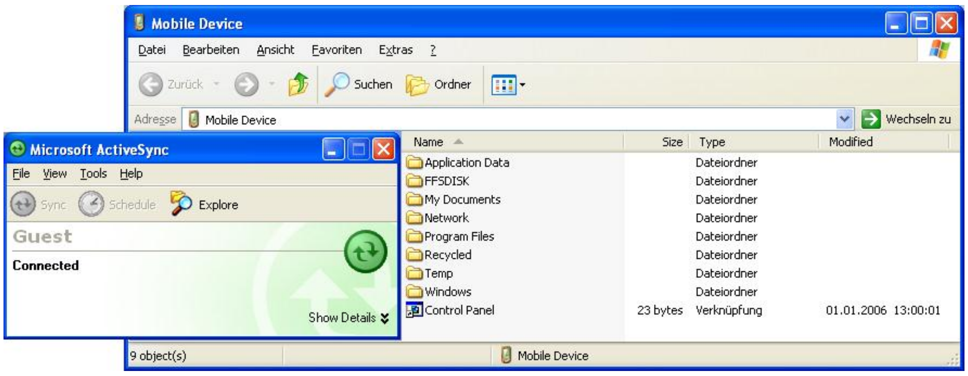
*Figure 7: FSDeviceSpy*



```
Welcome to the Windows CE Telnet Service on QBlissA8

Pocket CMD v 6.00
\> ndcucfg

NetDCU Config Utility Ready
Version: 043
Type help for commands

!>
```

*Figure 8: Telnet connection*

## 6.2 Active Sync



| Name | Size | Type | Modified |
|---|---|---|---|
| Application Data | | Dateiordner | |
| FFSDISK | | Dateiordner | |
| My Documents | | Dateiordner | |
| Network | | Dateiordner | |
| Program Files | | Dateiordner | |
| Recycled | | Dateiordner | |
| Temp | | Dateiordner | |
| Windows | | Dateiordner | |
| Control Panel | 23 bytes | Verknüpfung | 01.01.2006 13:00:01 |

*Figure 9: Browsing the device' filesystem with Active Sync*

## 6.3 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`.

*Note:*

NDCUCFG only starts on COM1 if debug message output is disabled.

```
:> O
Disable serial debug ouput during boot [N] (Y/N) ? :Y
 > Debug output disabled !!!
 > Press S during boot to step into bootloader.
```

Powering on the armStone with debug output disabled will output on COM1:

```
[…]
INFO: OEMLaunch: Jumping to Physical Address 0x30141E2Ch (Virtual
Address 0x0h)...
NetDCU Config Utility Ready
Version: 045
Type help for commands
!>
```

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 Powering-on .

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>
!>
```

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
```

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 any changes execute the command:

```
!>reg save
```

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

## 6.4 Network interface configuration

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

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

### 6.4.1 Network – General Facts

Being integrated into IP-network, in order to get directly referred by other network devices, every armStone 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 armStone. 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 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.

### 6.4.2 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
```

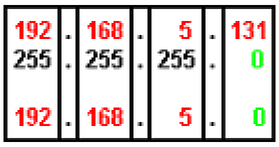
*Listing 6: IP-configuration from command line*

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.

### 6.4.3 Network interface configuration with NDCUCFG

Almost all device settings can be configured by registry.
Therefore you use the *reg* commands of the NDUCFG 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 7: NDCUCFG: Registry commands*

The Network parameters for armStone can be found under:
*[HKLM\Comm\ETHNETA1\Parms\TcpIp]*

Execute the command:

```
!>reg open \Comm\ETHNETA1\Parms\TcpIp
OK
```

*Listing 8: 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 9: NDCUCFG: Changing TcpIp settings*

The Network parameters for second Ethernet controller (optional) can be found under:
*[HKLM\Comm\ETHNETB1\Parms\TcpIp]*

### 6.4.4 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 armStone device.

Enter:

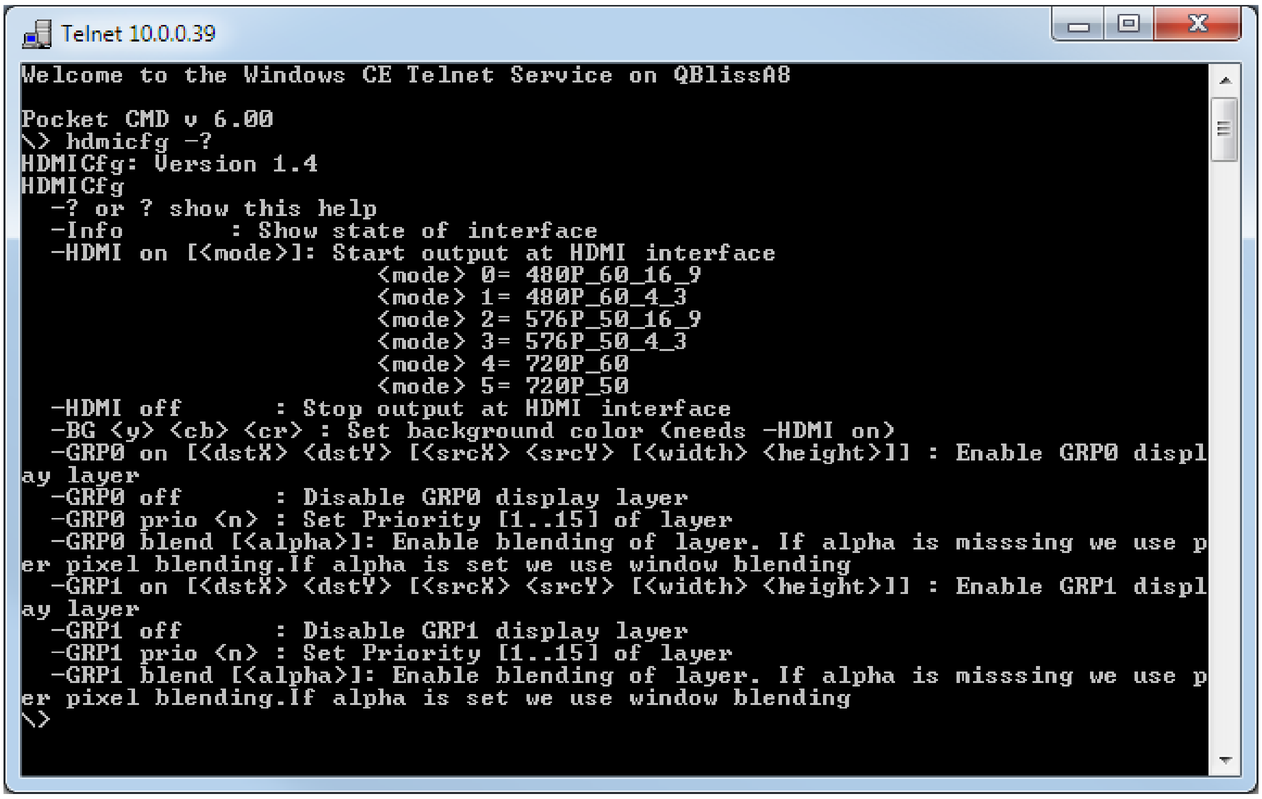
```
!>reg save
```

*Listing 10: NDCUCFG: Save modifications permanently*

and press *return*. 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 armStone system during current session of working with NDCUCFG utility are stored in persistent registry.

## 6.5 HDMICFG utility

armStoneA8 offers beside the LVDS output a HDMI interface. Before you see some output on HDMI you have to run hdmicfg.exe.



```
Welcome to the Windows CE Telnet Service on QBlissA8

Pocket CMD v 6.00
\> hdmicfg -?
HDMICfg: Version 1.4
HDMICfg
  -? or ? show this help
  -Info        : Show state of interface
  -HDMI on [<mode>]: Start output at HDMI interface
                        <mode> 0= 480P_60_16_9
                        <mode> 1= 480P_60_4_3
                        <mode> 2= 576P_50_16_9
                        <mode> 3= 576P_50_4_3
                        <mode> 4= 720P_60
                        <mode> 5= 720P_50
  -HDMI off       : Stop output at HDMI interface
  -BG <y> <cb> <cr> : Set background color (needs -HDMI on)
  -GRP0 on [<dstX> <dstY> [<srcX> <srcY> [<width> <height>]] : Enable GRP0 displ
ay layer
  -GRP0 off       : Disable GRP0 display layer
  -GRP0 prio <n> : Set Priority [1..15] of layer
  -GRP0 blend [<alpha>]: Enable blending of layer. If alpha is misssing we use p
er pixel blending.If alpha is set we use window blending
  -GRP1 on [<dstX> <dstY> [<srcX> <srcY> [<width> <height>]] : Enable GRP1 displ
ay layer
  -GRP1 off       : Disable GRP1 display layer
  -GRP1 prio <n> : Set Priority [1..15] of layer
  -GRP1 blend [<alpha>]: Enable blending of layer. If alpha is misssing we use p
er pixel blending.If alpha is set we use window blending
\>
```

*Figure 10: Running HDMICfg over Telnet*

HDMI driver of armStoneA8 can combine three different inputs to one output. Following picture shows how this works:

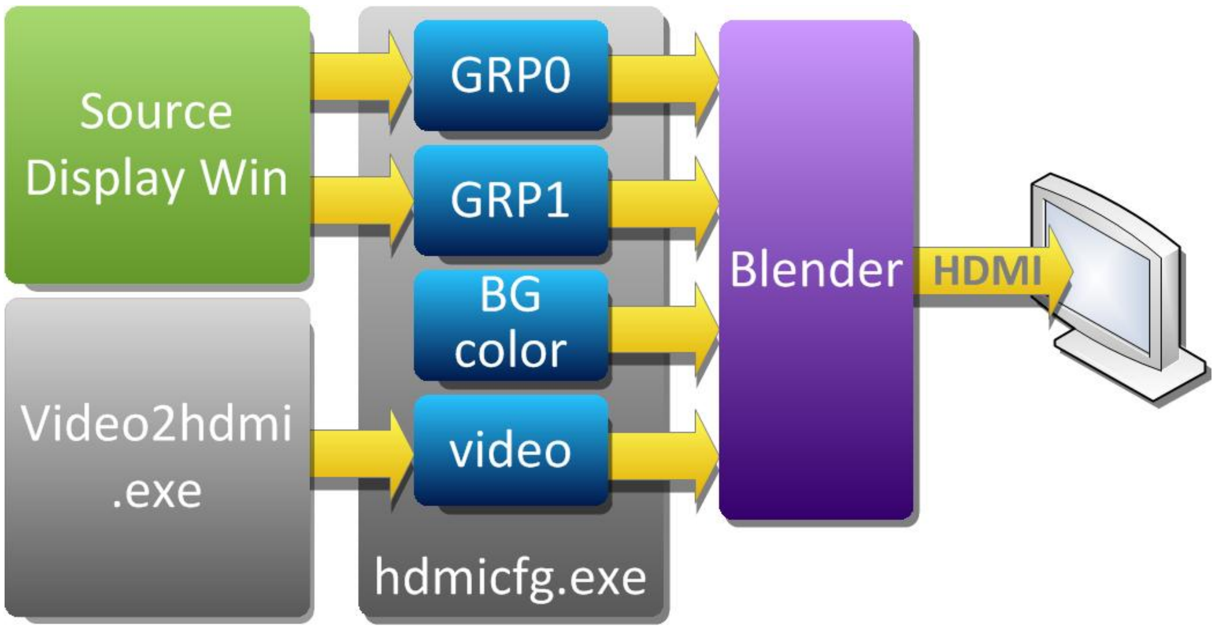


*Figure 11: Block diagram of HDMI driver*

With the tool HDMIcfg.exe you tell the driver how to combine the three layers. The easiest way is to output the main display layer to HDMI.

```
Welcome to the Windows CE Telnet Services on armStoneA8

Pocket CMD v 6.00

\> hdmicfg -HDMI on -GRP0 on
\> hdmicfg -info
```

*Listing 11: hdmicgf.exe: Output display window 0 to HDMI*

In case there is no HDMI monitor connected, you will see the following output:

```
Welcome to the Windows CE Telnet Services on armStoneA8

Pocket CMD v 6.00

\> hdmicfg -info
HDMICfg: Version 1.4
HDMI Device: not connected
HDMI Available Modes:
HDMI: OFF
VideoLayer: OFF
GraphicLayer0: OFF
GraphicLayer1: OFF
```

*Listing 12: hdmicgf.exe: No HDMI monitor detected*

# 7 Remote Tools (VS2005/2008)

Microsoft Visual Studio 2005 / 2008 are 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

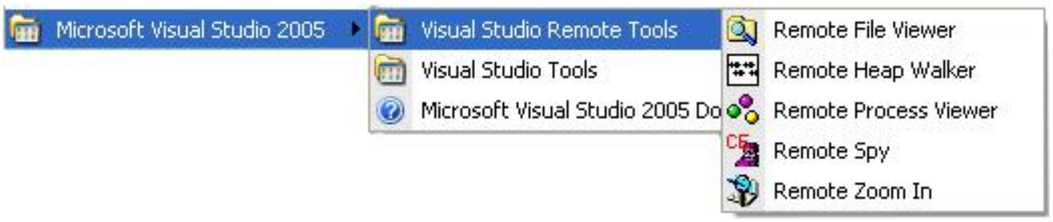


*Figure 12: Visual Studio Remote Tools*

# 8 Software Development

For software development you have to use Visual Studio 2005/2008. When programming for .NET Compact Framework 3.5 you need to have Visual Studio 2008 installed. The kernel-image that you can download from our download area includes already the Microsoft Compact Framework 3.5. This enables developer to write managed code in C# or VB.NET. It is also possible to develop applications in native code (C/C++) using the Win32 API or MFC. To use native code you need to install the armStone SDK that you also find in the download area.

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

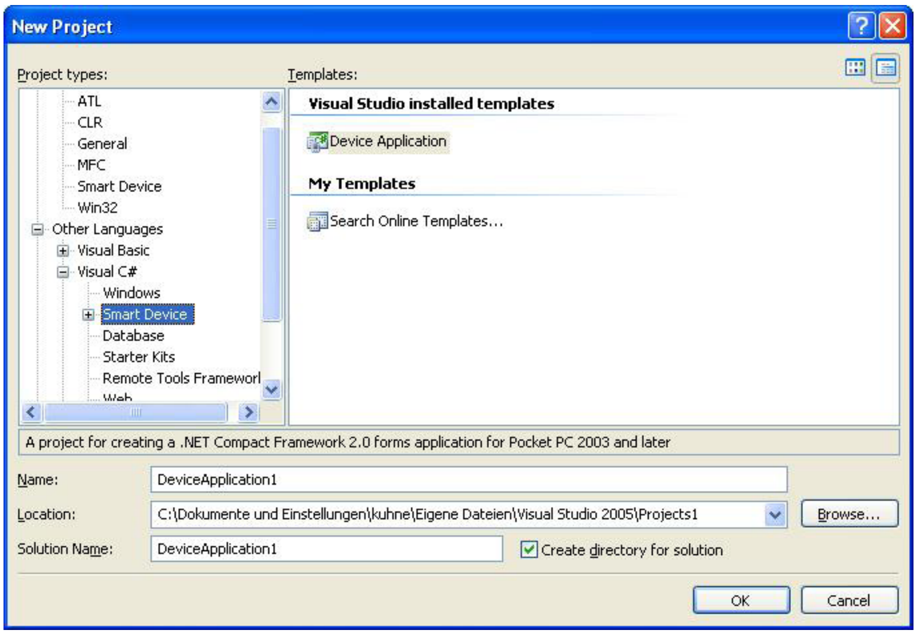
The best way (because easy to handle) is to connect via USB using Microsoft ActiveSync. For this install the latest version of ActiveSync on your PC (download ActiveSync from http://www.microsoft.com ) and connect armStone and PC using the USB device cable shipped with the SKIT. The connection is established 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 Windows CE kernel for armStone.

*Note:*

To write for / with CF3.5 you need VS2008 installed on your development pc. In case of CF2.0 VS2005 is needed.



A project for creating a .NET Compact Framework 2.0 forms application for Pocket PC 2003 and later

Name: DeviceApplication1

Location: C:\Dokumente und Einstellungen\kuhne\Eigene Dateien\Visual Studio 2005\Projects1

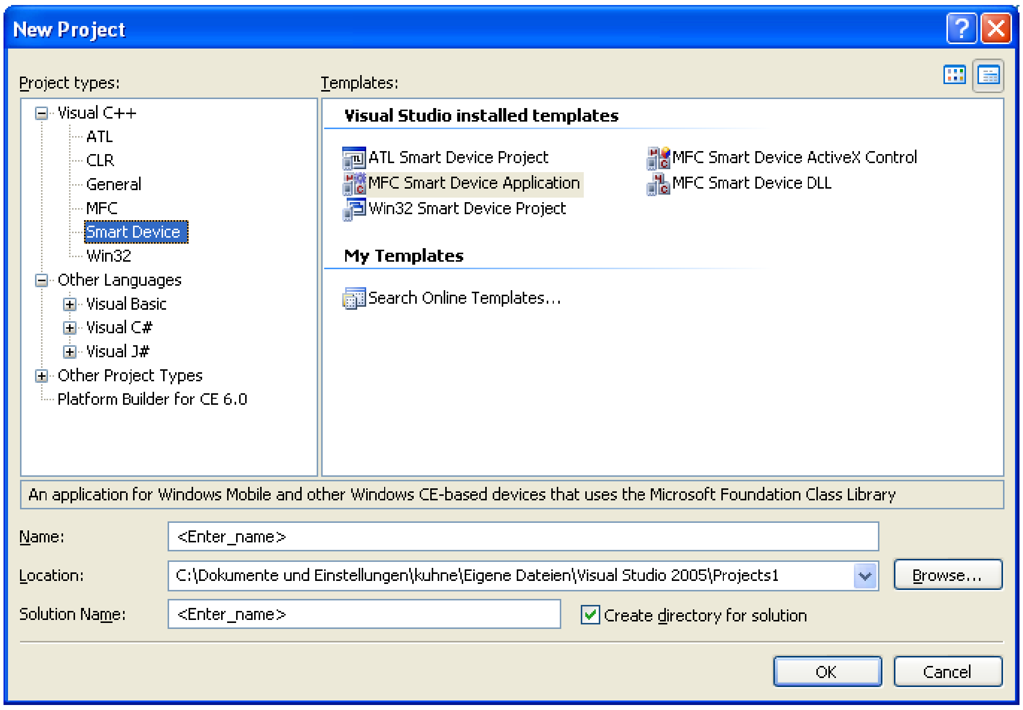
Solution Name: DeviceApplication1

☑ Create directory for solution

*Figure 13: New managed smart device application*

## 8.2 Visual Studio – Native Code

The application programmer can develop the application in C++ using the armStone SDK which can be found in our download area.


New Project

Project types:
- Visual C++
  - ATL
  - CLR
  - General
  - MFC
  - Smart Device
  - Win32
- Other Languages
  - Visual Basic
  - Visual C#
  - Visual J#
- Other Project Types
- Platform Builder for CE 6.0

Templates:

Visual Studio installed templates
- ATL Smart Device Project
- MFC Smart Device Application
- Win32 Smart Device Project
- MFC Smart Device ActiveX Control
- MFC Smart Device DLL

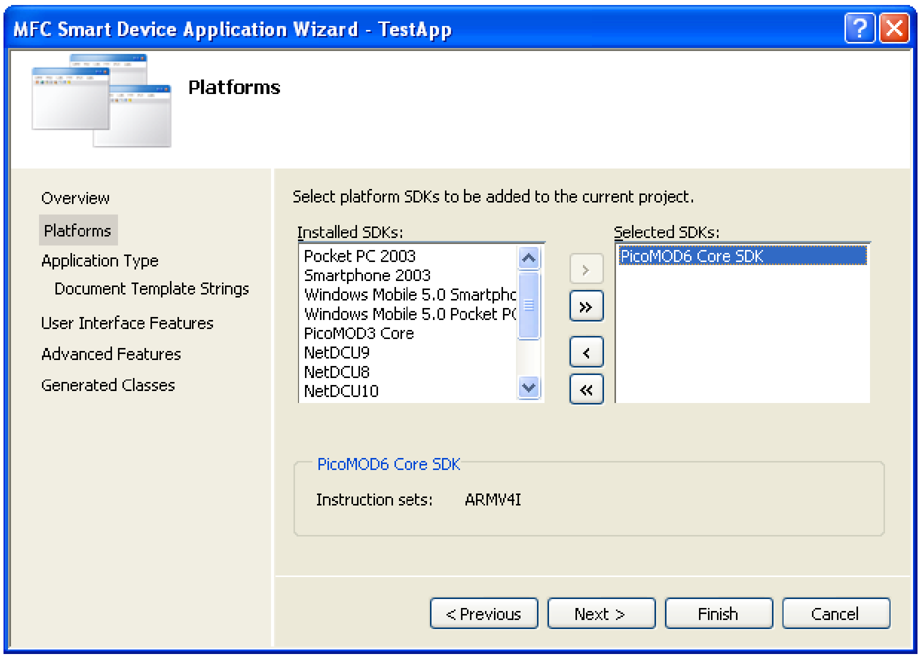
My Templates
- Search Online Templates...

An application for Windows Mobile and other Windows CE-based devices that uses the Microsoft Foundation Class Library

Name: <Enter_name>
Location: C:\Dokumente und Einstellungen\kuhne\Eigene Dateien\Visual Studio 2005\Projects1
Solution Name: <Enter_name>
[✓] Create directory for solution

Figure 14: New native smart device application

Select the respective SDK:


MFC Smart Device Application Wizard - TestApp

Platforms

- Overview
- Platforms
- Application Type
  - Document Template Strings
- User Interface Features
- Advanced Features
- Generated Classes

Select platform SDKs to be added to the current project.

Installed SDKs:
- Pocket PC 2003
- Smartphone 2003
- Windows Mobile 5.0 Smartphc
- Windows Mobile 5.0 Pocket P(
- PicoMOD3 Core
- NetDCU9
- NetDCU8
- NetDCU10

Selected SDKs:
- PicoMOD6 Core SDK

PicoMOD6 Core SDK
Instruction sets: ARMV4I

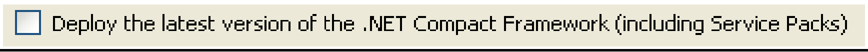
Figure 15: SDK for native smart device

### 8.2.1 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 armStone 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 device, 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 Firmware Update

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

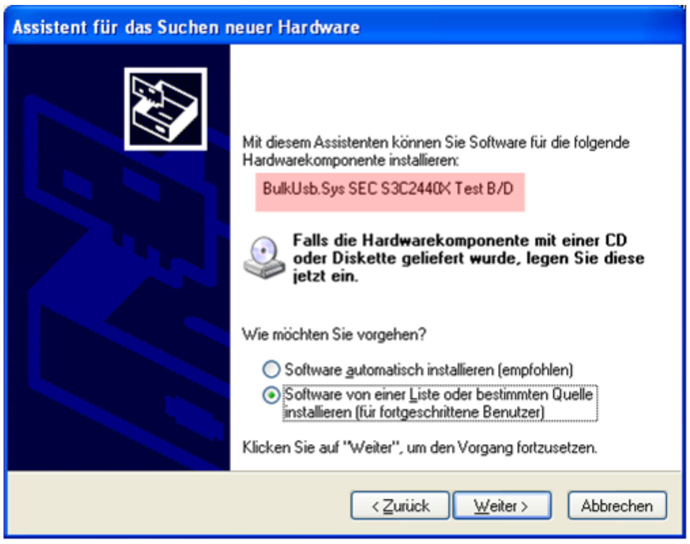
## 9.1 The NetDCU-USBLoader utility

The preferred method to update armStone is using the NetDCU-USBLoader which offers the possibility to download the bootloaders (NBOOT and EBOOT) and the Windows CE kernel to armStone via USB. The NetDCU-USBLoader can be found in the armStone download area on our website.

When connecting armStone 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.*


Assistent für das Suchen neuer Hardware

Mit diesem Assistenten können Sie Software für die folgende Hardwarekomponente installieren:

BulkUsb.Sys SEC S3C2440X Test B/D

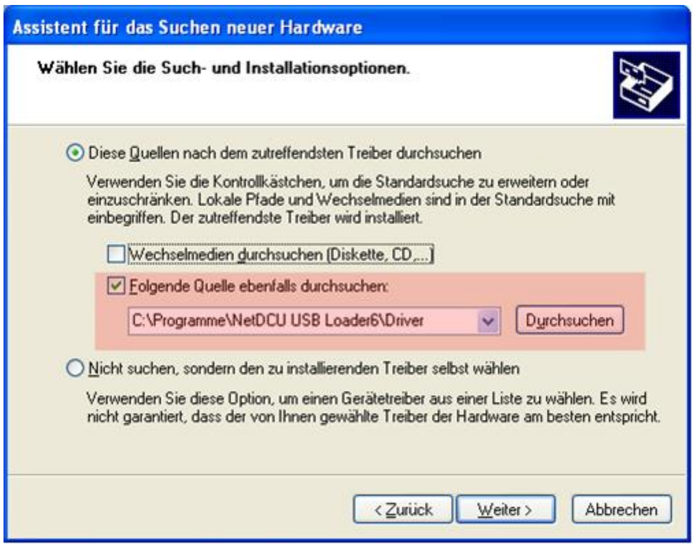
**Falls die Hardwarekomponente mit einer CD oder Diskette geliefert wurde, legen Sie diese jetzt ein.**

Wie möchten Sie vorgehen?

Software automatisch installieren (empfohlen)
[✓ Software von einer Liste oder bestimmten Quelle installieren (für fortgeschrittene Benutzer)]

Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

*Figure 16: NetDCU-USBLoader driver installation (1)*



Assistent für das Suchen neuer Hardware

**Wählen Sie die Such- und Installationsoptionen.**

[✓ Diese Quellen nach dem zutreffendsten Treiber durchsuchen]

Verwenden Sie die Kontrollkästchen, um die Standardsuche zu erweitern oder einzuschränken. Lokale Pfade und Wechselmedien sind in der Standardsuche mit einbegriffen. Der zutreffendste Treiber wird installiert.

☐ Wechselmedien durchsuchen (Diskette, CD,...)

☑ Folgende Quelle ebenfalls durchsuchen:

C:\Programme\NetDCU USB Loader6\Driver

Nicht suchen, sondern den zu installierenden Treiber selbst wählen

Verwenden Sie diese Option, um einen Gerätetreiber aus einer Liste zu wählen. Es wird nicht garantiert, dass der von Ihnen gewählte Treiber der Hardware am besten entspricht.

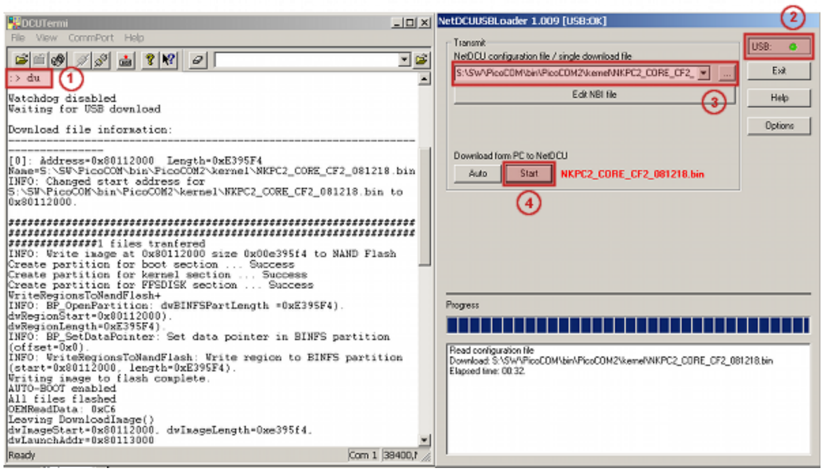
*Figure 17: NetDCU-USBLoader driver installation (2)*

## 9.2 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'.



```
:> du

Watchdog disabled
Waiting for USB download

Download file information:
----------------------------------------------------------------
---------------
[0]: Address=0x80112000  Length=0xE395F4
Name=S:\SW\PicoCOM\bin\PicoCOM2\kernel\NKPC2_CORE_CF2_081218.bin
INFO: Changed start address for
S:\SW\PicoCOM\bin\PicoCOM2\kernel\NKPC2_CORE_CF2_081218.bin to
0x80112000.

################################################################
################################################################
###############1 files tranfered
INFO: Write image at 0x80112000 size 0x00e395f4 to NAND Flash
Create partition for boot section ... Success
Create partition for kernel section ... Success
Create partition for FFSDISK section ... Success
WriteRegionsToNandFlash+
INFO: BP_OpenPartition: dwBINFSPartLength =0xE395F4).
dwRegionStart=0x80112000).
dwRegionLength=0xE395F4).
INFO: BP_SetDataPointer: Set data pointer in BINFS partition
(offset=0x0).
INFO: WriteRegionsToNandFlash: Write region to BINFS partition
(start=0x80112000, length=0xE395F4).
Writing image to flash complete.
AUTO-BOOT enabled
All files flashed
OEMReadData: 0xC6
Leaving DownloadImage()
dwImageStart=0x80112000, dwImageLength=0xe395f4.
dwLaunchAddr=0x80113000
```

NetDCUUSBLoader 1.009 [USB:OK]

NetDCU configuration file / single download file: S:\SW\PicoCOM\bin\PicoCOM2\kernel\NKPC2_CORE_CF2_

Download form PC to NetDCU: NKPC2_CORE_CF2_081218.bin

```
Read configuration file
Download: S:\SW\PicoCOM\bin\PicoCOM2\kernel\NKPC2_CORE_CF2_081218.bin
Elapsed time: 00:32
```

*Figure 18: 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 armStone 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

## 9.3 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'.

You can also download the Eboot bootloader via the serial debug port.
Use the DCUTerm terminal program to connect to the serial debug port of your armStone device.
Enter NBoot by holding 's' while powering the device. You will see output like:

```
F&S Nand Loader VN12 built Mar 13 2012 17:17:56
armStoneA8 Rev. 1.10
256 MB RAM (2 chips) 128 MB FLASH 1000 MHz


Please select action
'd' -> Serial download of bootloader
'c' -> Load bootloader from SD card
'E' -> Erase flash
'B' -> Show bad blocks
Use NetDCUUsbLoader for USB download
```

*Listing 13: NBOOT command shell*

Now press 'd' to start serial download. You will see message:

```
Waiting for bootloader...
```

Go to the File menu and select *"Transmit Binary File…"*. Then change to the folder where eboot.nb0 is located (ebootv210_v103.nb0) and confirm by open button.

You will see download progress by some dots. After download finished (transmit message box disappears) you will see output like:

```
Success, checksum: 0x3fc1

>>> EBoot image loaded (262144 bytes) <<<

Please select action
'f' -> Save image to flash
'x' -> Execute image
'd' -> Serial download of bootloader
'c' -> Load bootloader from SD card
'E' -> Erase flash
'B' -> Show bad blocks
Use NetDCUUsbLoader for USB download
```

*Listing 14: NBOOT after download of EBOOT*

Press ‘f’ to save u-boot and then re-power the device. You will then asked to enter the Windows CE MAC address:

```
Windows CE ethernet MAC address not set.
 Enter WindowsCE MAC address (actual ff:ff:ff:ff:ff:ff):
000551037D22
```

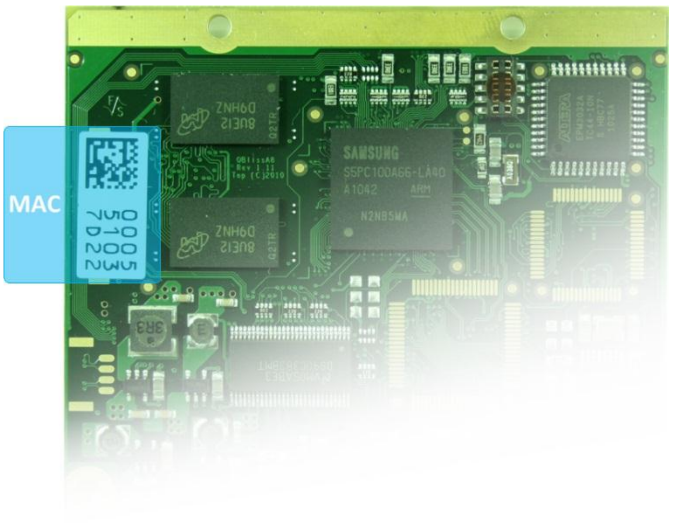


*Figure 19: armStone MAC address*

After MAC setup you will enter armStone WinEC Bootloader.

**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.

## 9.4 Configure EBoot

You can setup EBoot by several commands. An overview is printed by the '?' command:

```
armStoneA8 - WindowsCE Bootloader

:> ?
armStoneA8 - WindowsCE Bootloader
Monitor Help
I   - Displays bootloader settings
N   - Set parameters for Ethernet protocol
P   -  Setup partitioning information
E   - Total flash disk erase
R   - Total registry erase
DE  - Download Kernel (Ethernet)
DU  - Download Kernel/Bootloader (USB)
A0  - Don't start download after boot
AE  - Start download after boot using Ethernet interface
AU  - Start download after boot using USB interface
L0  - Don't launch previously stored kernel after boot
L1  - Launch previously stored kernel after boot
LC  - Clear launch address stored in parameter RAM
MF  - Store kernel on Flash disk
MR  - Store kernel in RAM memory
F   - Enter F3S serial number
O   - Enable/Disable serial debug output
C   - Reset to factory default configuration
T   - Total self-test of most of peripherals of NetDCU system
BPC - Select PWM channel
BPF - Adjust PWM base frequency
BPD - Adjust PWM duty cycle
BPE - Enable currently selected PWM channel

:>
```

*Listing 15: EBOOT command shell*

## 9.5 Partitioning the Flash storage

There are up to 3 partitions possible on armStone.

- **OS-Image (BINFS):** The Windows CE 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 Windows CE system.
- **Extended Partition:** The extended partition must be administrated in Windows CE. 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 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 16: 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 17: Flash partitioning process*

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

## 9.6 Updating NBoot

In case of a newer NBoot or when switching from Windows Embedded CE to Linux you need to download NBoot. This is done similar to download the EBoot. But in all cases an already running NBoot is required to download a newer version.

Please contact support@fs-net.de for more information.

## 9.7 Windows Embedded Compact 7 Kernel Image

### 9.7.1 Flashing Windows Embedded Compact 7 Kernel Image

Use DCUTerm as a terminal program to connect to the serial debug port of your armStone device. Enter EBoot by pressing ‘S’. Then run the commands to download the WinEC7 kernel image to NAND flash.

**Preparing EBoot to download WinEC7 Kernel Image:**

To store WinEC7 Kernel Image permanently we use the command ‘MF’. We also want to start the image after we flashed it, this is setup by command ‘L1’. As the best way to download the image is by USB we run the ‘DU’ command. To perform these steps enter EBoot by holding ‘S’ while powering on the device.

**Starting download:**

The best way to download the WinEC7 Kernel Image is by USB using the NetDCUUSBLoader application. Select the respective binary image (NKQA8_PRO_110421.bin) and activate the ‘start’ button when ready. Be sure to have a working connection to the device, which is signaled by a green ‘led’ on the upper right button. Please read chapter *Installing NetDCUUSBLoader application* for a short description of the installation process of the NetDCUUSBLoader application.

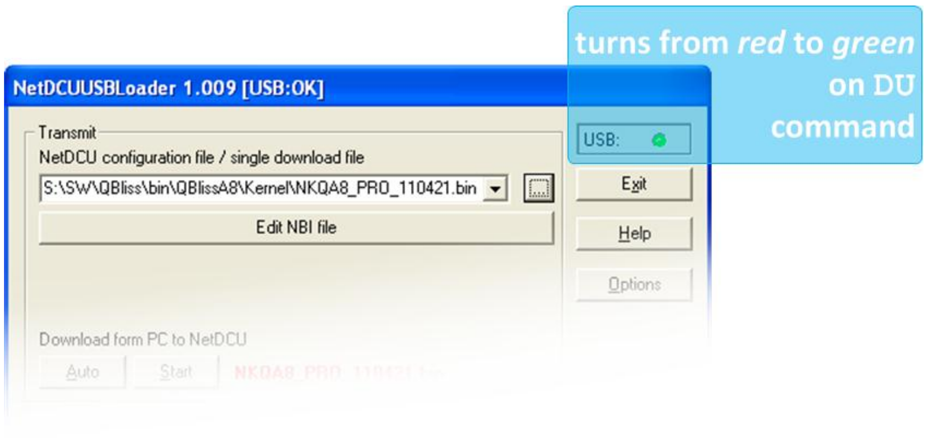

turns from red to green on DU command
NetDCUUSBLoader 1.009 [USB:OK]
Transmit
NetDCU configuration file / single download file
S:\SW\QBliss\bin\QBlissA8\Kernel\NKQA8_PRO_110421.bin
Edit NBI file
USB:
Exit
Help
Options
Download form PC to NetDCU
Auto
Start
NKQA8_PRO_110421.bin

*Figure 20: NetDCUUSBLoader application*

Here is the EBoot output when starting the connection by ‘DU’ command:

```
armStoneA8 - WindowsCE Bootloader

:> MF

:> L1 > After next Reboot Kernel will be started automatically

:> DU
HW-Watchdog: OFF
Waiting for USB download
Initialize IRQ Vector Tables
HighSpeed detected
HighSpeed detected
```

Next listing shows the messages while downloading the binary (after the NDCUUSBLoader *start button* has been activated):

```
Download file information:
--------------------------------------------------------------------
-----------
[0]: Address=0x80100000  Length=0x1494598  Name=
S:\SW\FSS5PV210\WCE6\Kernel\NKV210_CE6P_120321.bin
INFO: Changed start address for
S:\SW\FSS5PV210\WCE6\Kernel\NKV210_CE6P_120321.bin to 0x80100000.
--------------------------------------------------------------------
-----------
1 files tranfered
INFO: Write image at 0x80100000 size 0x01494598 to NAND Flash
Create partition for boot section ...
Success
Create partition for kernel section ...
Success
Create partition for FFSDISK section ...
Success
Create partition for EXTENDED section ...
Success
WriteRegionsToNandFlash+r

Writing single region/multi-region update, dwBINFSPartLength:
21579160
INFO: BP_OpenPartition: dwBINFSPartLength =0x1494598).
dwRegionStart=0x80100000).
dwRegionLength=0x1494598).
INFO: BP_SetDataPointer: Set data pointer in BINFS partition
(offset=0x0).
INFO: WriteRegionsToNandFlash: Write region to BINFS partition
(start=0x80100000, length=0x1494598).
Writing image to flash complete.
AUTO-BOOT enabled
All files flashed
```

When re-power the device the WinEC7 Kernel Image will be extracted to RAM and started by EBoot.
You can re-enter EBoot by pressing ‘S’ while powering on the device.

# 10 Appendix

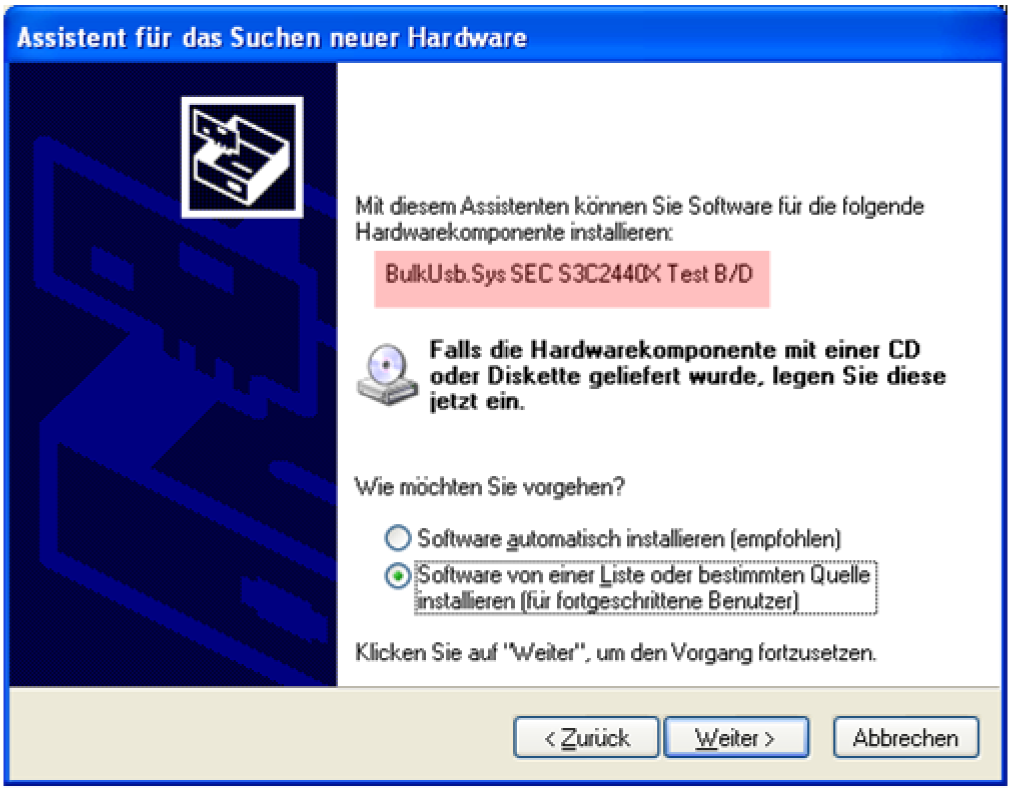
## 10.1 Installing NetDCUUSBLoader application

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

When connecting armStone and NetDCUUSBLoader for the very first time you have to install an USB driver on your development PC. That driver is shipped with NetDCUUSBLoader installer and gets copied to its installation directory.

**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*.*


Assistent für das Suchen neuer Hardware

Mit diesem Assistenten können Sie Software für die folgende Hardwarekomponente installieren:

BulkUsb.Sys SEC S3C2440X Test B/D

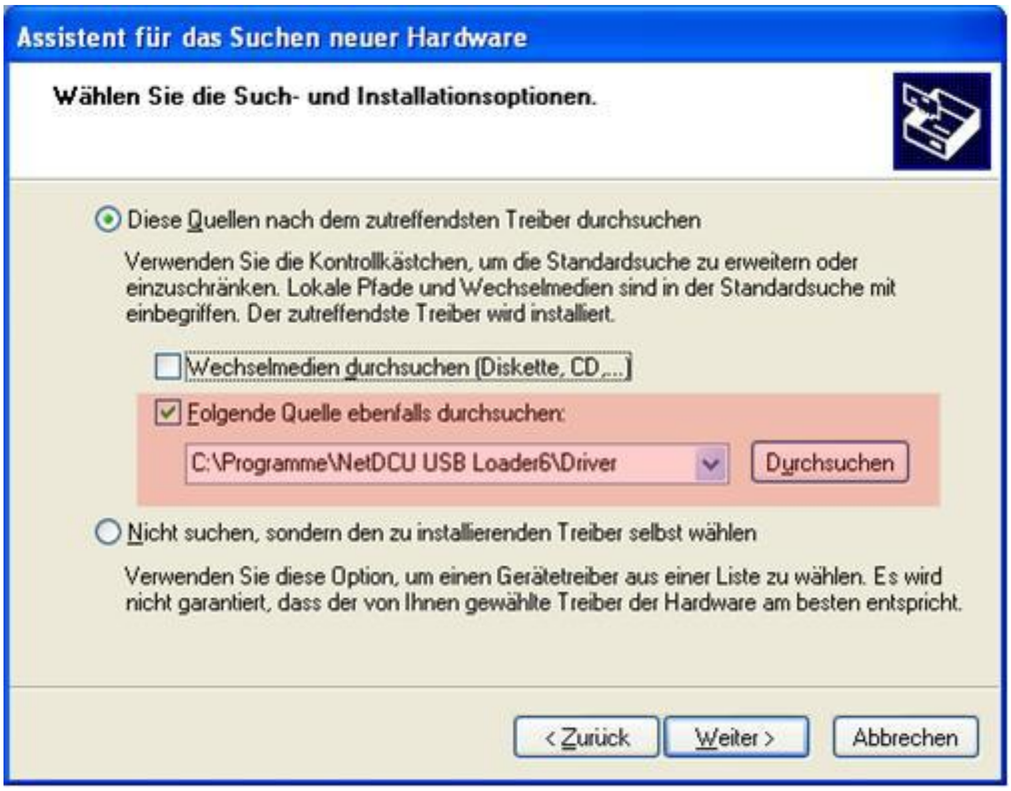
Falls die Hardwarekomponente mit einer CD oder Diskette geliefert wurde, legen Sie diese jetzt ein.

Wie möchten Sie vorgehen?

- Software automatisch installieren (empfohlen)
- [✓ Software von einer Liste oder bestimmten Quelle installieren (für fortgeschrittene Benutzer)]

Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

*Figure 21: NetDCUUSBLoader driver installation (1)*


Assistent für das Suchen neuer Hardware

Wählen Sie die Such- und Installationsoptionen.

[✓ Diese Quellen nach dem zutreffendsten Treiber durchsuchen]

Verwenden Sie die Kontrollkästchen, um die Standardsuche zu erweitern oder einzuschränken. Lokale Pfade und Wechselmedien sind in der Standardsuche mit einbegriffen. Der zutreffendste Treiber wird installiert.

- [ ] Wechselmedien durchsuchen (Diskette, CD,...)
- [✓] Folgende Quelle ebenfalls durchsuchen:

C:\Programme\NetDCU USB Loader6\Driver

Nicht suchen, sondern den zu installierenden Treiber selbst wählen

Verwenden Sie diese Option, um einen Gerätetreiber aus einer Liste zu wählen. Es wird nicht garantiert, dass der von Ihnen gewählte Treiber der Hardware am besten entspricht.

*Figure 22:NetDCUUSBLoader driver installation (2)*

## 10.2 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

## Figures