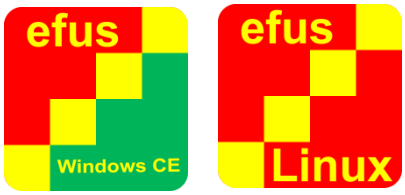
First Steps Documentation

Windows Embedded for efus™A9

Version 1.1
2014-10-01



Preliminary

This Document Is Subject to Change without Notice

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
2014-10-01	1.0	efusA9	A	*	Created Document	JG
2015-01-09	1.1	efusA9	M		Updated to new software	HF
2015-03-04	1.1	efusA9	M	3	Debug line (replace COM4: by COM1:)	

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

About this document

The following document describes the usage and handling of efusA9.

Table of Contents

History 2

About this document 2

Table of Contents 4

1 Getting started 5

1.1 Connecting basic peripheral devices 6

2 Download Area 8

3 Powering-on efusA9 11

4 Bootup sequence 13

5 Configure efusA9 15

5.1 The FSDeviceSpy Utility 16
5.2 Active Sync (Windows CE 6.0 and Compact 7 only) 18
5.3 Using NDCUCFG utility 18
5.4 Network interface configuration 20
5.4.1 Network – General Facts 20
5.4.3 Network – Network address 21
5.4.4 Network interface configuration with NDCUCFG 22
5.4.5 Network – saving the parameters to registry 23
5.5 Touch panel calibration 24
5.6 Device Management 25

6 Software Development 27

6.1 Visual Studio 2005/2008 – Managed Code 28
6.2 Visual Studio 2005/2008 – Native Code 29
6.2.1 Debugging an device application 30
6.3 Visual Studio 2012/2013 31
6.4 Visual Studio 2012/2013 – Managed Code 33
6.5 Visual Studio 2012/2013 – Native Code 34

7 Remote Tools 35

8 Firmware Update 37

8.1 The NetDCU-USBLoader utility 37
8.2 Updating – Windows CE kernel image 38
8.2.1 Preparing EBoot to download Kernel Image 38
8.2.2 Download via USB 38
8.2.3 Download via Ethernet 41
8.2.4 Partitioning the Flash storage 41
8.3 Updating EBoot 43
8.4 Configure EBoot 45
8.5 Updating NBoot 46

9 Appendix 47

Important Notice 47
Warranty Terms 48
Listings 49
Figures 50

1 Getting started

This documentation is a step by step introduction in how to use efusA9. 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

efusA9 is available with i.MX6 Solo, DualLite and Quad processor. Solo and DualLite share the same silicon. i.MX6 Quad has a different silicon. Because the differences are large, F&S has different binaries for Solo/DualLite (SDL) and Quad version (Q) of the board at the beginning of shipment of efusA9.
Naming of this software was as follows:

NBOOT	nbootIMX6Q_03.bin	iMX6 Quad
EBOOT	ebootIMX6Q_09.nb0	iMX6 Quad
WinCE 7 image	XIPIMX6Q_C7E_141006.bin	iMX6 Quad
NBOOT	nbootIMX6SDL_17.bin	iMX6 Solo/DualLite
EBOOT	ebootIMX6SDL_09.nb0	iMX6 Solo/DualLite
WinCE 7 image	XIPIMX6SDL_C7E_141006.bin	iMX6 Solo/DualLite

Today there is only one version of the software for iMX6 Solo, DualLite and Quad.

NBOOT	nbootIMX6_24.bin	iMX6 Solo/DualLite/Quad
EBOOT	ebootIMX6_10.nb0	iMX6 Solo/DualLite/Quad
WinCE 7 image	XIPIMX6_C7E_141231.bin	iMX6 Solo/DualLite/Quad

1.1 Connecting basic peripheral devices

Next picture shows efusA9 with the position and description of connectors. The board measures 4,7 x 6,21 cm (1,85 x 2,44 inches). You can find the schematic of the carrier board in the download area.

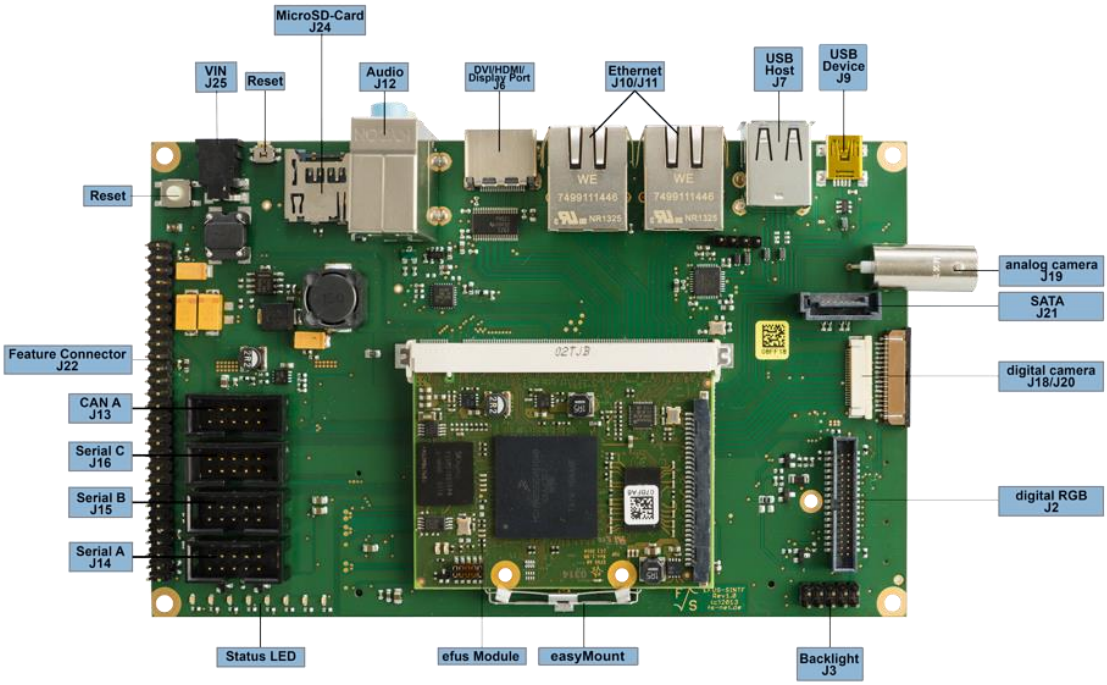


Figure 1:efusA9 on carrier board (top side)

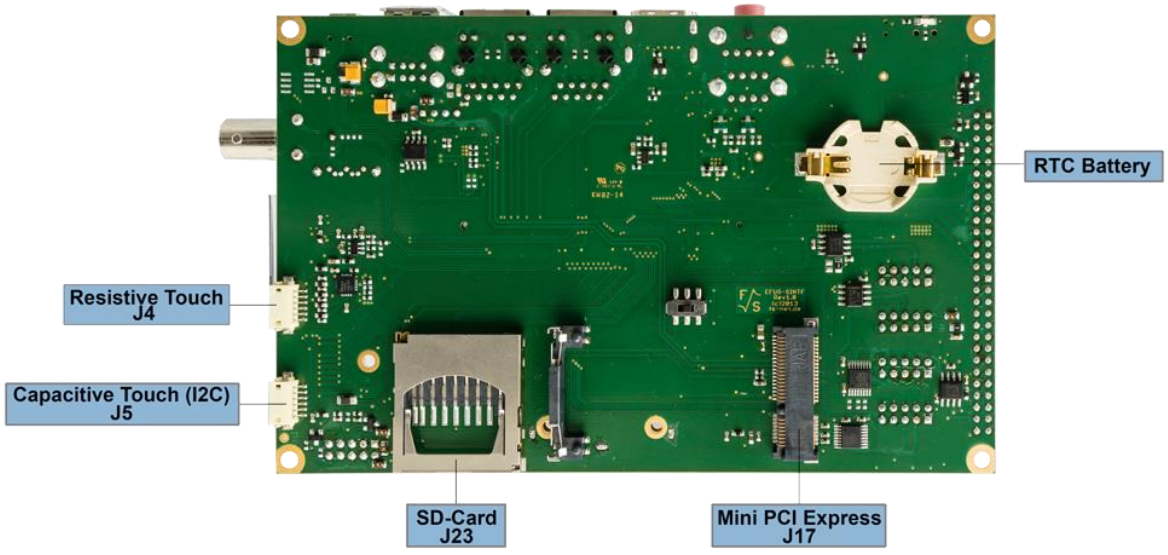


Figure 2:efusA9 on carrier board (bottom side)

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 efusA9 can be found in the efusA9 hardware documentation on our website.

2 Download Area

If you want to download hardware and software documentations, go to our main website

http://www.fs-net.de

and select *Products* from the menu at the top. Select the board family and then your specific board. The top half of the screen will now show the board and in the lower half of the screen you will find an additional menu where you can select *Documents*.

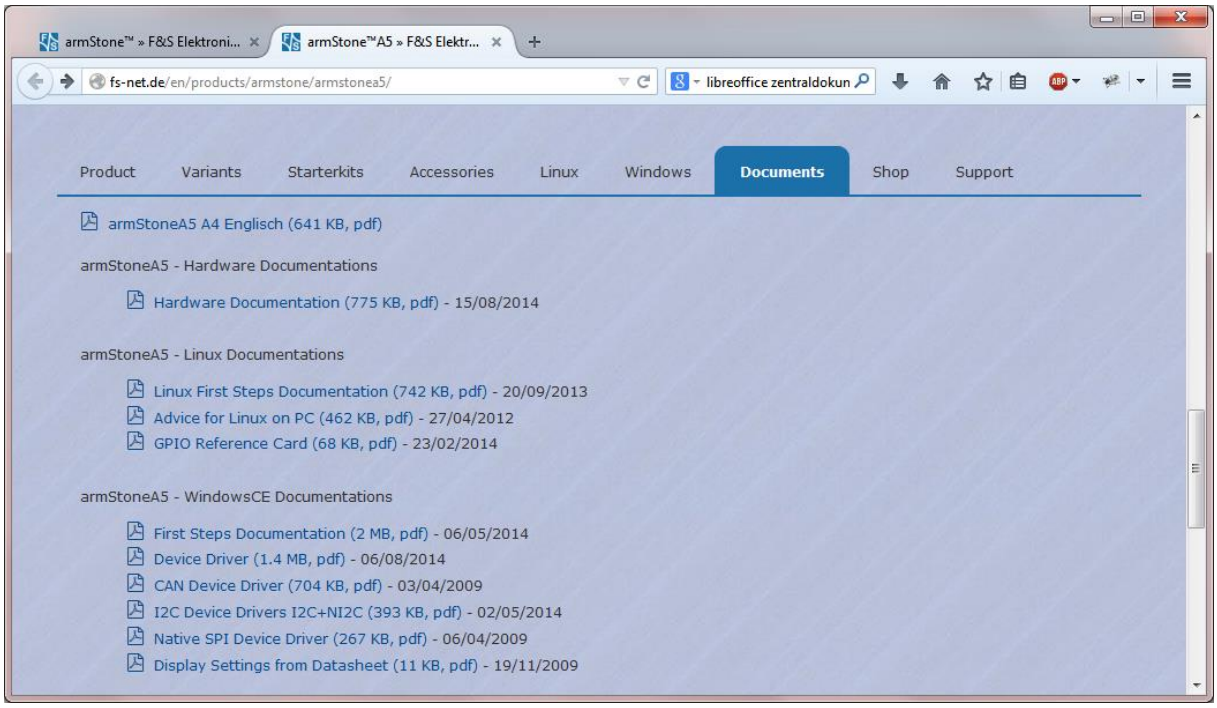


Figure 3: Download Documents from F&S website

If you want to download any software, you first have to register with the website. Click on *Login* right at the top of the window. Click on the text *“I am not registered, yet. Register now.”*

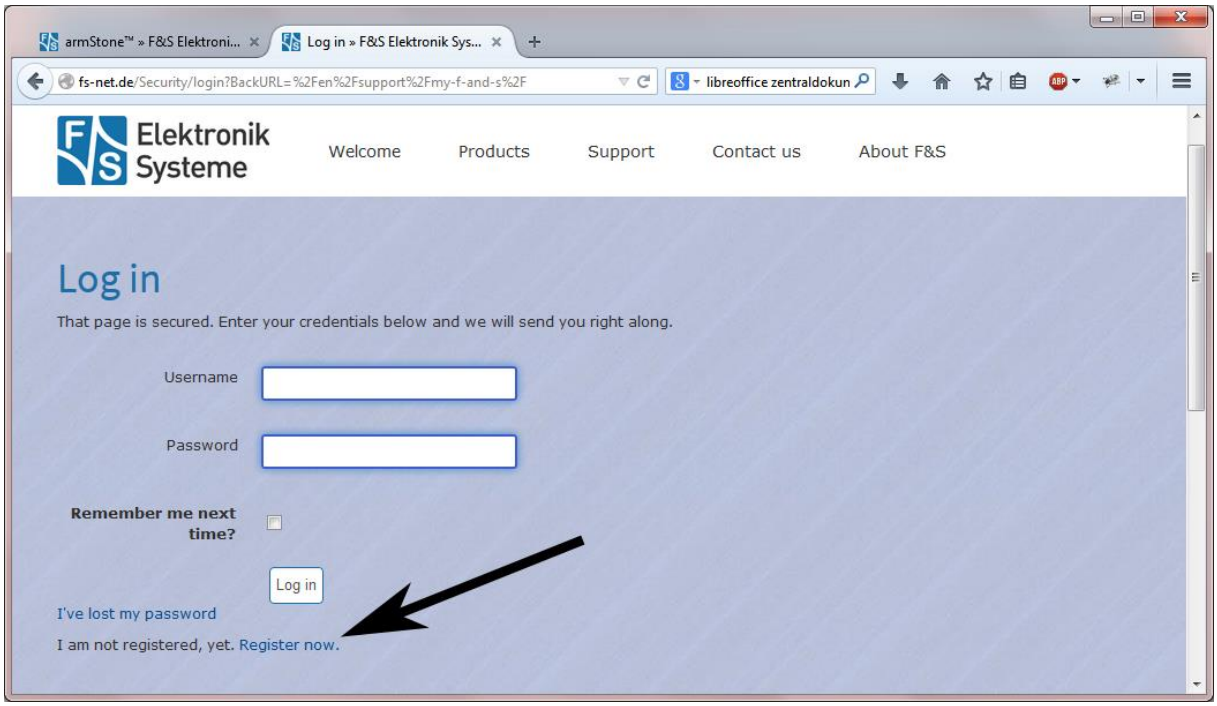


Figure 4: Register with F&S website

In the screen appearing now, fill in all fields and then click on Register. You are now registered and can use the personal features of the website, for example the Support Forum where you can look for solutions to any problems and where you can ask your own questions. These questions are usually answered by the F&S Support Team or also sometimes by other users.
After logging in, you are at your personal page, called “My F&S”. You can always reach this place by selecting *Support → My F&S* from the top menu. Here you can find all software downloads that are available for you.

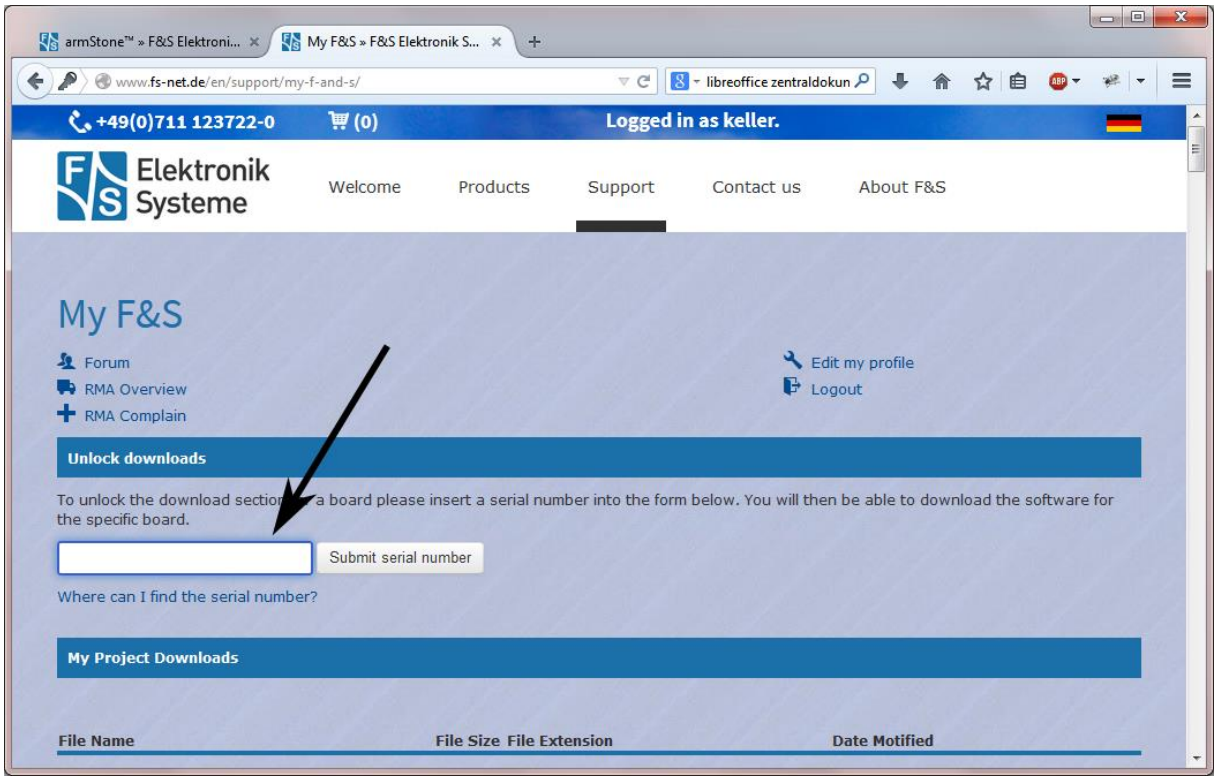


Figure 5: Unlock software with the serial number

To get access to the software of a specific board, you have to enter the serial number of one of these boards. Click on “Where can I find the serial number” to get pictures of examples where to find this number on your product. Enter the number in the white field and press *Submit serial number*. This enables the software section for this board type for you. There you will find Linux, Windows CE, and any other software and tools available for this platform.

“My F&S” also informs you about new software versions, fixed bugs and known bugs. For already released software versions you can click on *Changelog*. For future software releases you can click on *Roadmap*.

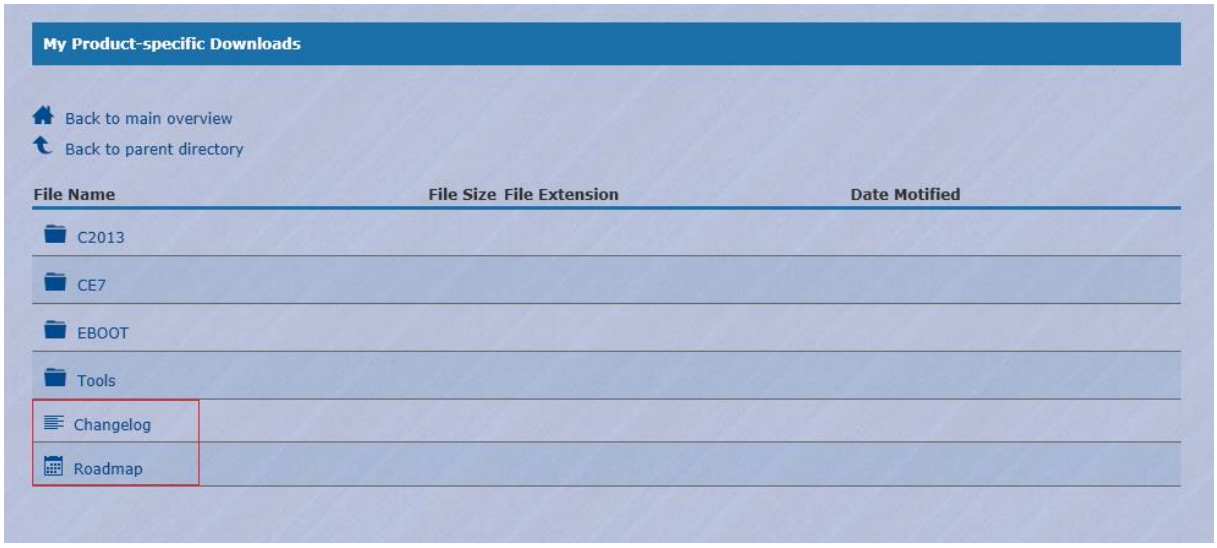


Figure 6: Changelog and Roadmap

3 Powering-on efusA9

Before powering on efusA9, you should establish a serial connection between efusA9 and your development PC. Please use the cables shipped with the efus Starterkit to connect your PC with the Debug-Port of efus (COM1).

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 efusA9.

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):

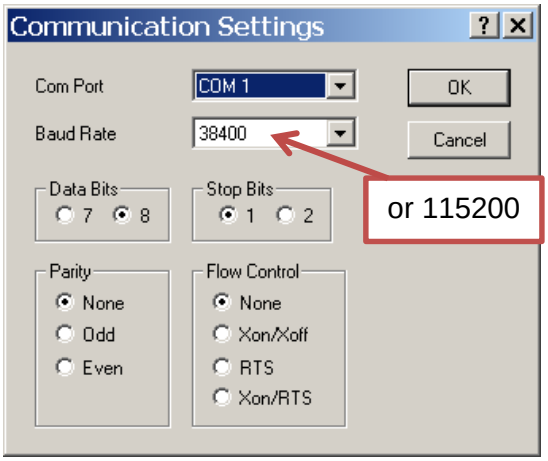


Figure 7: DCUTerm.exe configuration: CommPort → Settings

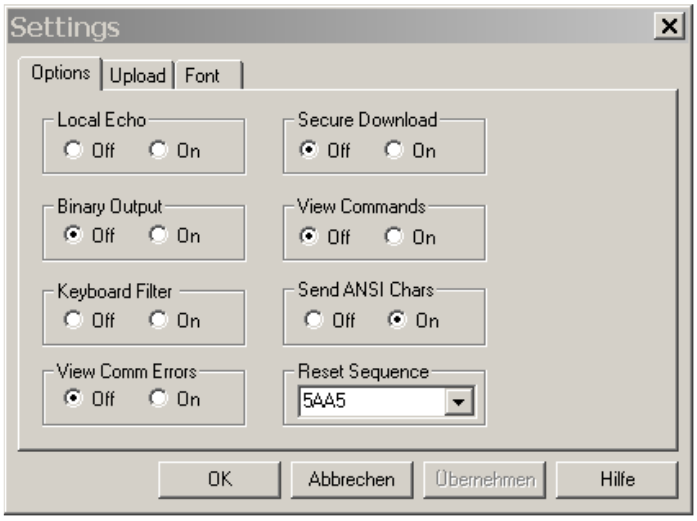


Figure 8: DCUTerm.exe configuration: View → Options

- Press the connect button in DCUTerm.exe

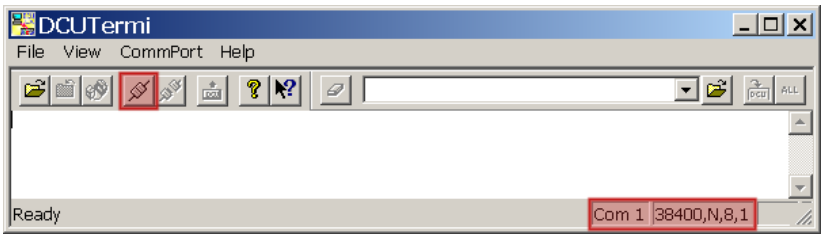


Figure 9: Opening the serial connection in DCUTerm.exe

- Use the serial cable shipped with efusA9 Starterkit to establish a connection for debugging between one of the serial ports on the board and the COM-port you are using on your PC (COM1 is used in this document)
- Plug a cable between power connector of efusA9 (see Figure 1:efusA9 on carrier board (top side)) and your power supply. Usually you need to connect ground and +5V (2A).
- Connect the board by LAN to your Ethernet.

4 Bootup sequence

The startup process of efusA9 is divided into three steps:

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

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

3. **Windows CE kernel image**
   - Windows CE operating system and all drivers for the efusA9.
   - Offers you the possibility to develop and debug custom applications.

Both bootloaders (NBoot and EBoot) are equipped with a small configuration menu, which is accessible via serial debug port (COM1).

To open one of these boot menus the following characters must be entered **while** booting the device. Reset the board and hold the keys, until the menu appears.

NBoot: **'s'**
EBoot: **<SHIFT>+'s'** (capital „s")

Note:

Details on updating NBoot, EBoot and WINCE kernel image can be found in chapter 8 Firmware Update.

As F&S delivers efusA9 with pre-installed bootloaders and kernel image you should see debug output on COM1 like here:

Bootloader:

```
Microsoft Windows CE Bootloader Common Library Version 1.2 Built Jul
11 2014 16:29:05
Microsoft Windows CE Bootloader for efusA9 Built Dec 31 2014
Portions copyright (c) 2012 F&S Elektronik Systeme GmbH
Boot Loader, Version 1.0
NBoot, Version VN24
HW rev. 1.10
```

Listing 1: Bootup: Bootloader

Read kernel image from NAND flash:

```
Kernel (3671kB) read from flash disk started finished in 1000
milliseconds
INFO: OEMLaunch: Jumping to Physical Address 0x102200000h (Virtual
Address 0x102200000h)...
```

Listing 2: Bootup: Read image from NAND flash

Starting kernel image:

```
Windows CE Kernel for ARM (Thumb Enabled)
efusA9 V1.00 - Firmware Init
Copyright (c) 2013 F&S Elektronik Systeme GmbH
Build: Apr 14 2014/12:31:11
```

Listing 3: Bootup: Start Windows CE

Loading device drivers:

```
BE2: Version 1.4, ActiveKey = Drivers\Active\03
NI2C: Version 0.7, ActiveKey = Drivers\Active\15
BE2: Version 1.4, ActiveKey = Drivers\Active\16
UART: Version 1.9, Key = Drivers\Active\17
UART: Port disabled. Serial debug is on !
UART: Version 1.9, Key = Drivers\Active\19
UART: Version 1.9, Key = Drivers\Active\20
TCH: Version 0.5, ActiveKey = Drivers\Active\21
CID: Version 2.4, ActiveKey = Drivers\Active\22
CID: Version 2.4, ActiveKey = Drivers\Active\23
ENET: Version 01.01, ActiveKey = Comm\ETHNETA
AIN: Version 1.2, ActiveKey = Drivers\Active\25
PWM: Version 1.2, ActiveKey = Drivers\Active\26
I2C: Version 4.2, ActiveKey = Drivers\Active\27
SDHC: Version 1.2, ActiveKey = Drivers\Active\29
WAV: Version 2.3, ActiveKey = Drivers\Active\31
ENET: Version 01.01, ActiveKey = Comm\ETHNETB
BCS: Version 1.4, ActiveKey = Drivers\Active\41
NSPI: Version 3.0, ActiveKey = Drivers\Active\42
DIO: Version 2.8, ActiveKey = Drivers\Active\43
[…]
```

Listing 4: Bootup: Loading drivers

Start NDCUCFG application (read chapter 5.3 Using NDCUCFG utility for detail information):

```
NDCUCFG V: 56 started. Platform: efusA9
NDCUCFG Open COM1: at 38400 Baud
```

Listing 5: Bootup: Flash initialization

Note:

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

5 Configure efusA9

Configuration of the efus 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 efus 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.

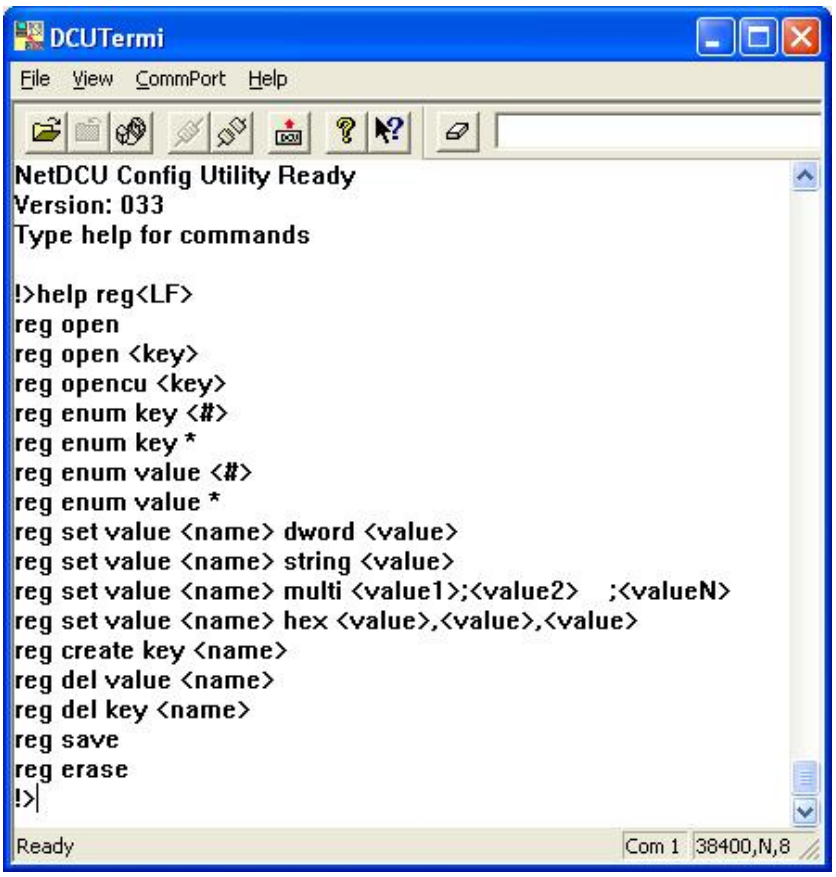


Figure 10: NDCUCFG over serial debug connection

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

5.1 The FSDeviceSpy Utility

There are different ways to make a connection between your PC and efus. 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, efus sends broadcast packet with some special information. This may take some time after a reset (~1 minute). 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.

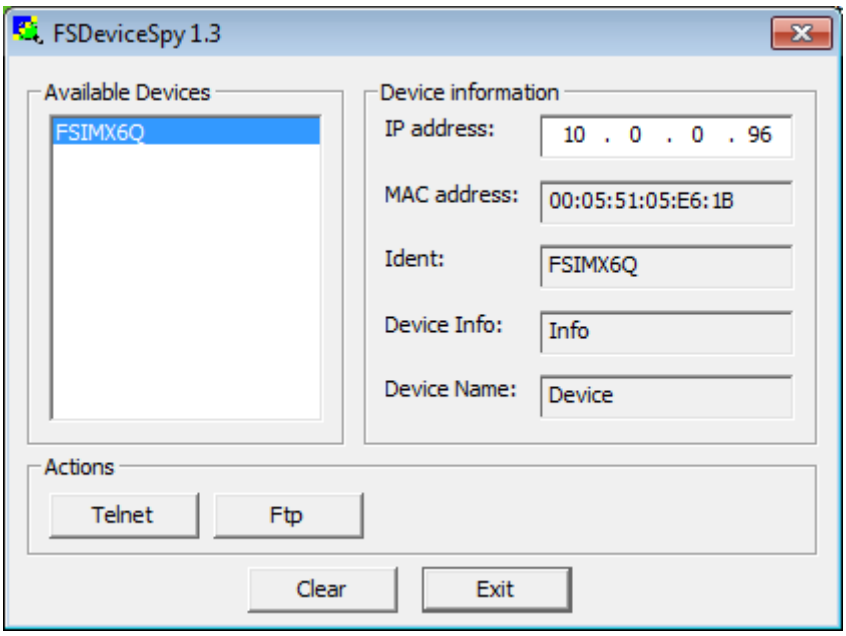


Figure 11: FSDeviceSpy

Note:

efus™A9 shares software platform with armStone™A9 and PicoMODA9. All these boards are based on Freescale i.MX6 processor. It's only one software for all boards. Because of that, efus™A9 is recognized as FSiMX6 in FSDeviceSpy.

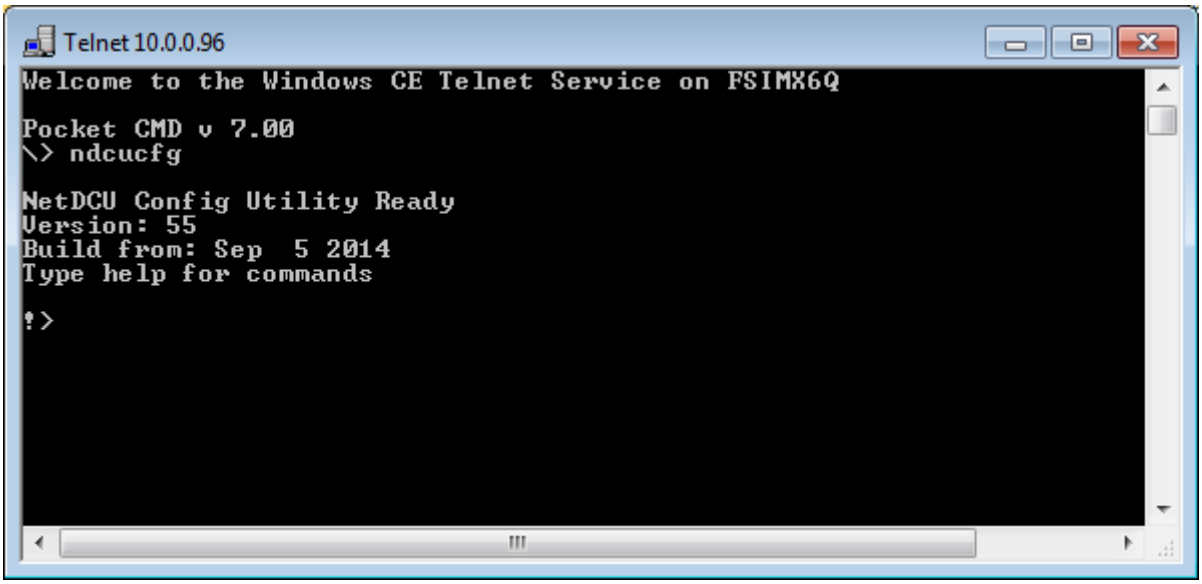


Figure 12: Telnet connection

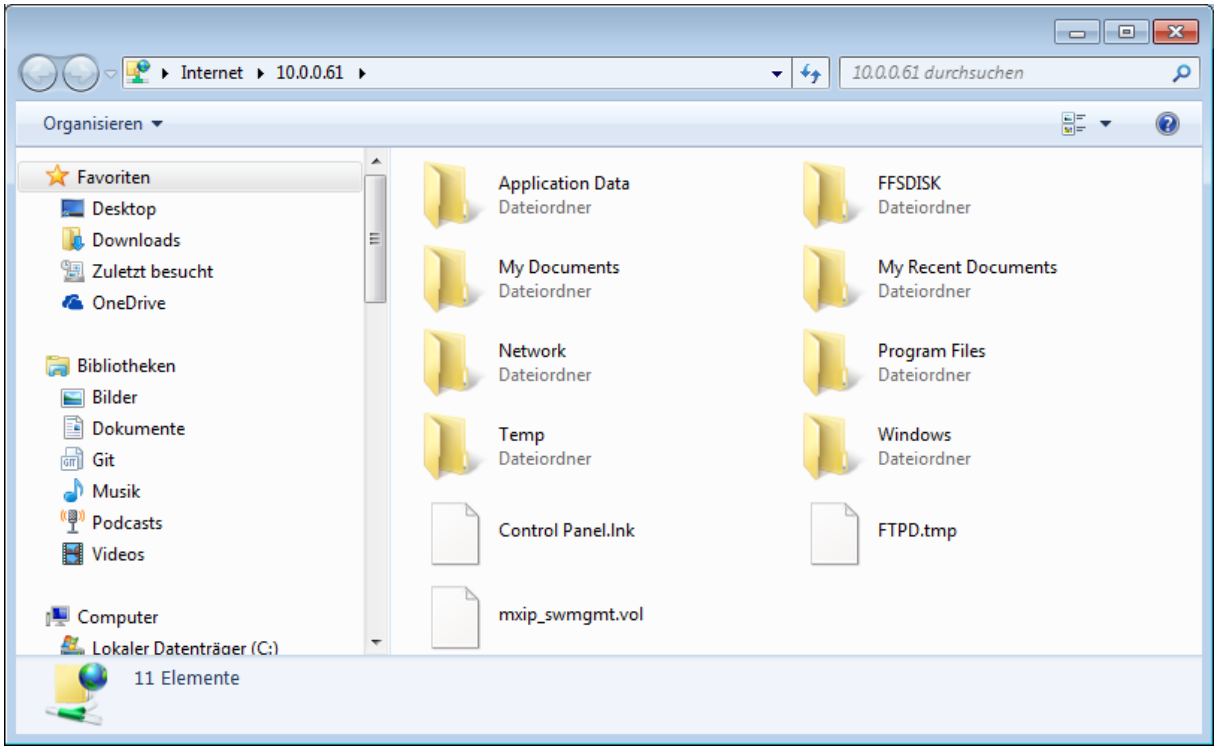


Figure 13: FTP connection

5.2 Active Sync (Windows CE 6.0 and Compact 7 only)

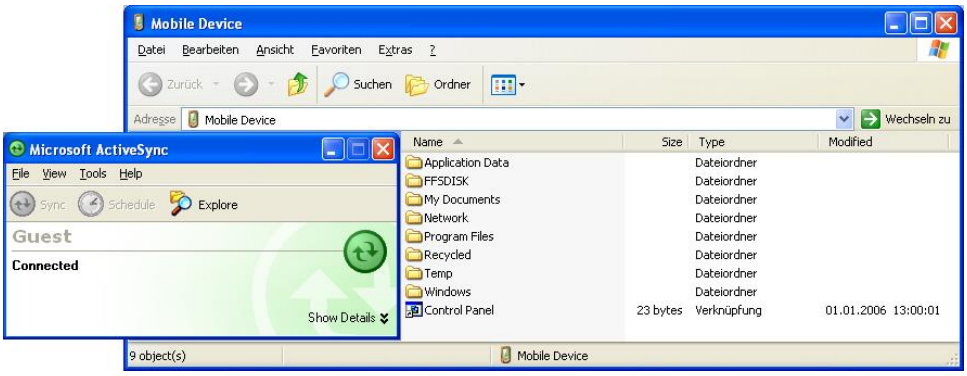


Figure 14: Browsing the device' file system with Active Sync

5.3 Using NDCUCFG utility

NDCUCFG is a command line tool for several configuration
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 output during boot [N] (Y/N) ? :Y
 > Debug output disabled!!!
 > Press S during boot to step into bootloader.
```

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

```
NetDCU Config Utility Ready
Version: 56
Type help for commands
!>
```

Listing 6: NDCUCFG command prompt

If this command prompt (!>) appears in the terminal program you are ready to pass commands to NDCUCFG utility. Otherwise something went wrong and the kernel image could not be loaded correctly. Please check various parameters described in chapter 3 Powering-on efusA9.

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:

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

Listing 7: NDCUCFG 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.

5.4 Network interface configuration

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

5.4.1 Network – General Facts

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

5.4.3 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 8: IP-configuration from command line

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

Windows IP configuration:

Ethernet Adapter ETHNETA:
IP address: **192.168.0.131**
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.

5.4.4 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 9: NDCUCFG: Registry commands

The Network parameters for efus can be found under:

[HKLM\Comm\ETHNETA1\Parms\TcpIp]

Execute the command:

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

Listing 10: 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 11: NDCUCFG: Changing TcpIp settings

5.4.5 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 efus device.

Enter:

```
!>reg save
```

Listing 12: NDCUCFG: Save modifications permanently

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

5.5 Touch panel calibration

The display, available with the Starterkit, is equipped with a 4 wire touch panel or a capacitive touch panel. At delivery status this touch panel should already be calibrated satisfyingly. But when using another display, or if the touch panel is not working properly, it might be required to calibrate the touch panel driver again. This is performed by using NDCUCFG.
The desired command is *"touch calibrate"*. After executing this command, the touch panel calibrating routine should be displayed on the LCD screen.

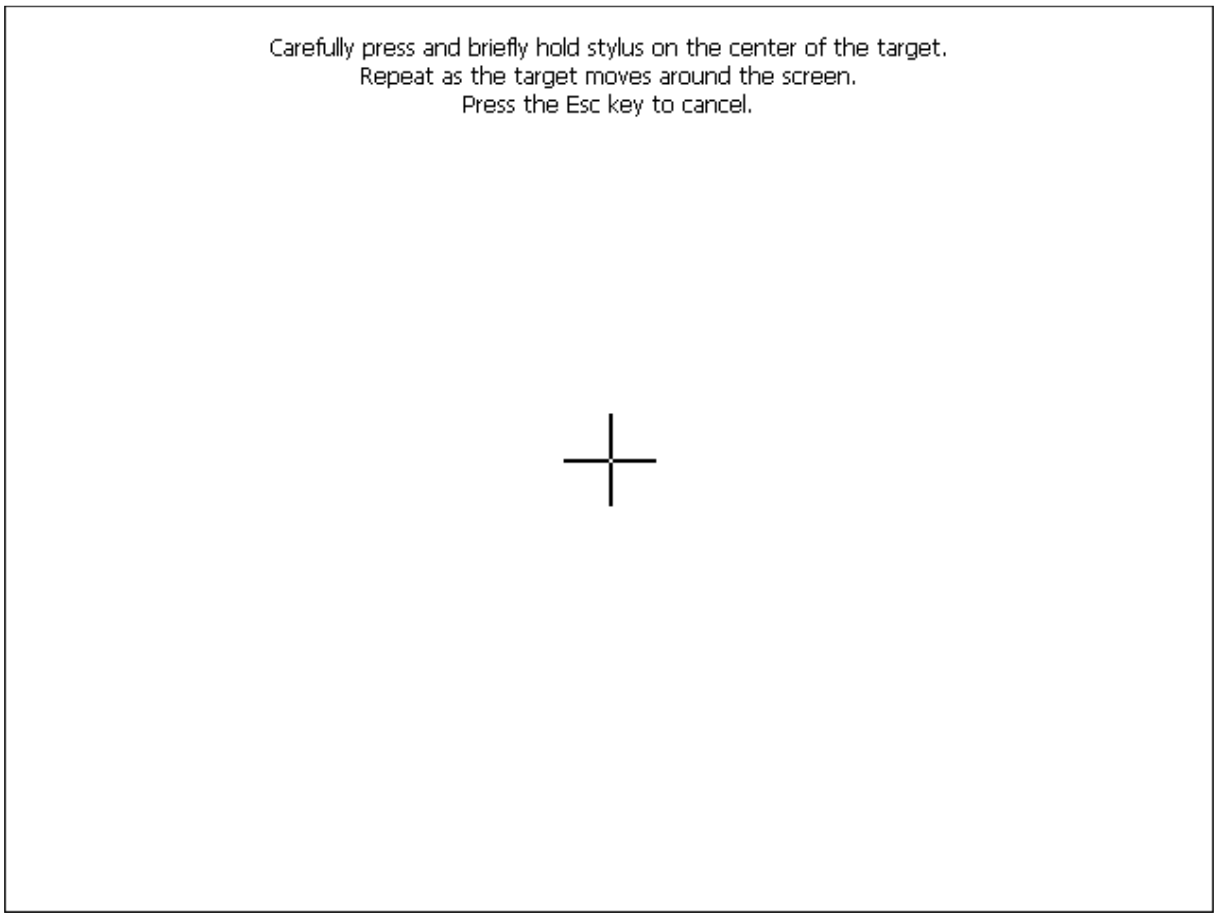


Figure 15: Initial screen of the touch panel calibration program.

After holding the stylus on the cross in the center of the screen, the cross moves around to each corner where it must be 'touched' again. When the 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 the 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.

5.6 Device Management

Standard image for efus loads a lot of drivers and services during boot time. To speed up boot time or to change function of a multiplexed port you can activate/deactive devices.

To get a list of all devices type:

```
!>device enum<LF>
 A   Name: CCS0:  Key: Drivers\BuiltIn\ccfgsvc
 A   Name: LPC2:  Key: Drivers\BuiltIn\ALPCD
 A   Name: SCA1:  Key: Drivers\BuiltIn\SmartCard
 A   Name: NFY0:  Key: Notify
 A   Name: SSP0:  Key: Drivers\BuiltIn\NTLMSSP_SVC
 A   Name: WAM1:  Key: Drivers\BuiltIn\WAPIMAN
 A   Name: CRD0:  Key: Drivers\BuiltIn\credsvc
 A   Name: SDC1:  Key: Drivers\BuiltIn\SDBusDriver
 A   Name: HCD1:  Key: Drivers\BuiltIn\USB1
 A   Name: USB1:  Key: Drivers\USB\USBHCK
 A   Name: I2C3:  Key: Drivers\BuiltIn\I2C3
 A   Name: COM3:  Key: Drivers\BuiltIn\efusA9\UART3
 A   Name: COM2:  Key: Drivers\BuiltIn\efusA9\UART2
 A   Name: COM1:  Key: Drivers\BuiltIn\efusA9\UART1
 A   Name: TCH1:  Key:Drivers\BuiltIn\efusA9\Touch_SX865x_ADPEDT
 A   Name: AIN1:  Key: Drivers\BuiltIn\efusA9\ANALOGIN
 A   Name: PWM1:  Key: Drivers\BuiltIn\efusA9\PWM
 A   Name: I2C8:  Key: Drivers\BuiltIn\efusA9\WaveDevI2C
 A   Name: SHC1:  Key: Drivers\BuiltIn\SDMMC_CH2
 A   Name: NDS0:  Key: Drivers\BuiltIn\NDIS
 A   Name: WAV1:  Key: Drivers\BuiltIn\efusA9\WaveDev
 A   Name: HID1:  Key: Drivers\USB\ClientDrivers\Hid\Hid_Class
 A   Name: NWF1:  Key: Comm\nwifi
 A   Name: WSS1:  Key: Drivers\BuiltIn\Ws2Serv
 A   Name: UIO1:  Key: Drivers\BuiltIn\NDISUIO
 A   Name: BTD0:  Key: Drivers\BuiltIn\BTD
 A   Name: ARS1:  Key: Drivers\BuiltIn\autoras
 A   Name: NPW1:  Key: Drivers\BuiltIn\NdisPower
 A   Name: DP60:  Key: Comm\Devices\TCPIP6\DHCPV6
 A   Name: DP40:  Key: Comm\Devices\TCPIP\DHCP
 A   Name: ETM1:  Key: Drivers\BuiltIn\Ethman
 A   Name: BCS1:  Key: Drivers\BuiltIn\BCSend
 A   Name: SPI1:  Key: Drivers\BuiltIn\SPI1
 A   Name: DIO1:  Key: Drivers\BuiltIn\DIGITALIO
 A   Name: NSI1:  Key: Drivers\BuiltIn\NSIPROXY
 A   Name: ETS0:  Key: Drivers\BuiltIn\EAP3SVC
 A   Name: NSI0:  Key: Drivers\BuiltIn\NSISVC
 A   Name: EHS0:  Key: Drivers\BuiltIn\EAPSVC
 A   Name: CMS0:  Key: Drivers\BuiltIn\CmService
 A   Name: RDR0:  Key: Drivers\BuiltIn\Redir
 A   Name: LPC1:  Key: Services\LPCD
 A   Name: DCM1:  Key: Services\DCOMSSD
 A   Name: LAS0:  Key: Services\LASSD
 A   Name: FTP0:  Key: Services\FTPD
 A   Name: TEL0:  Key: Services\TELNETD
 A   Name: UFN1:  Key: Drivers\BuiltIn\USBFN
 A   Name: BTS1:  Key: Services\BTSVC
```

```
 A   Name: COM9:  Key: \Drivers\USB\FunctionDrivers\Serial_Class
 A   Name: SSH0:  Key: Services\SSHD
 A   Name: DST0:  Key: Services\DSTSVC
 A   Name: PXY0:  Key: Services\Proxy
 A   Name: SNA0:  Key: Services\SNApi
 D   Name: SIP0:  Key: Drivers\Builtin\SIP
 D   Name: DSK1:  Key: Drivers\Builtin\NANDFLASH
 D   Name: tch1:  Key: Drivers\Builtin\Touch_SX865x
 D   Name: TCH2:  Key: Drivers\Builtin\Touch_MXT224
 D   Name: TCH3:  Key: Drivers\Builtin\Touch_EDT
OK
!>
```

Listing 13: NDCUCFG: List Devices

From the above listing you can see, that in front of each driver there is a capital `A` or a capital `D`. The meaning is `A` stands for `Activated` and `D` stands for `Deactivated`.

Use the command *help* device to get explanation of how to activate or deactivate a device.

6 Software Development

For software development you have to use Visual Studio from Microsoft. Please take a look to the following table to select the right version of Visual Studio.

	VS2005	VS2008	VS2012/2013
Windows CE 6, C++	yes	yes	
Windows CE 6, CF.NET 2.0	yes		
Windows CE 6, CF.NET 3.5		yes	
Windows Compact 7, C++	yes	yes	
Windows Compact 7, CF.NET 3.5		yes	
Windows Compact 2013			yes

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 efus SDK that you also find in the download area. For Compact 2013 you need to install the SDK for .NET development too.

To connect Visual Studio to efus for software development you can use a USB device connection and/or Ethernet connection. For Compact 2013, only Ethernet is possible

The best way (because easy to handle) for Windows CE 6 and Windows Compact 7 is to connect via USB using Microsoft Device Center (ActiveSync). For this install the latest version of Microsoft Device Center on your PC (download Microsoft Device Center from http://www.microsoft.com) and connect efus and PC using the USB device cable shipped with the SKIT. The connection is established automatically.

For Compact 2013, you have to use TCP/IP connection over Ethernet. In the future TCP/IP over USB will also be possible.

6.1 Visual Studio 2005/2008 – 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 efus.

Note:

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

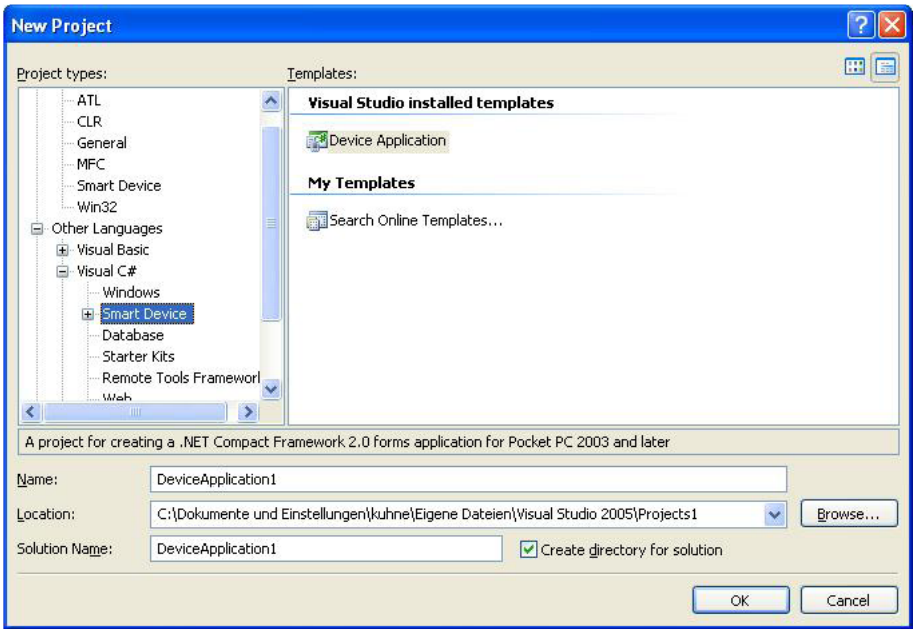


Figure 16: New managed smart device application

6.2 Visual Studio 2005/2008 – Native Code

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

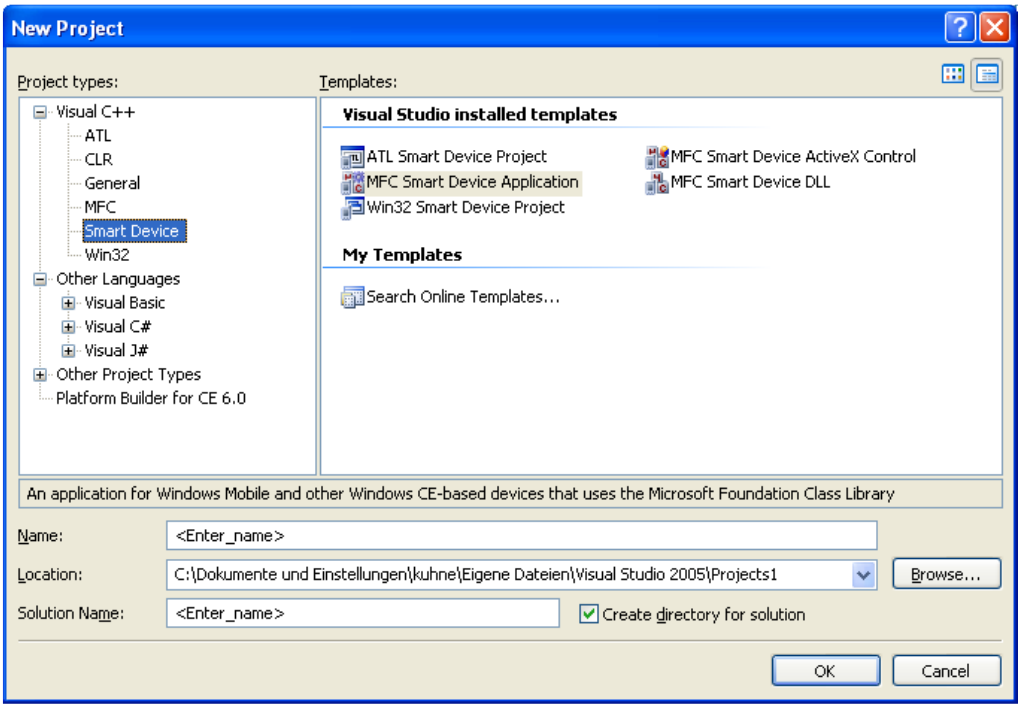


Figure 17: New native smart device application

Select the respective SDK:

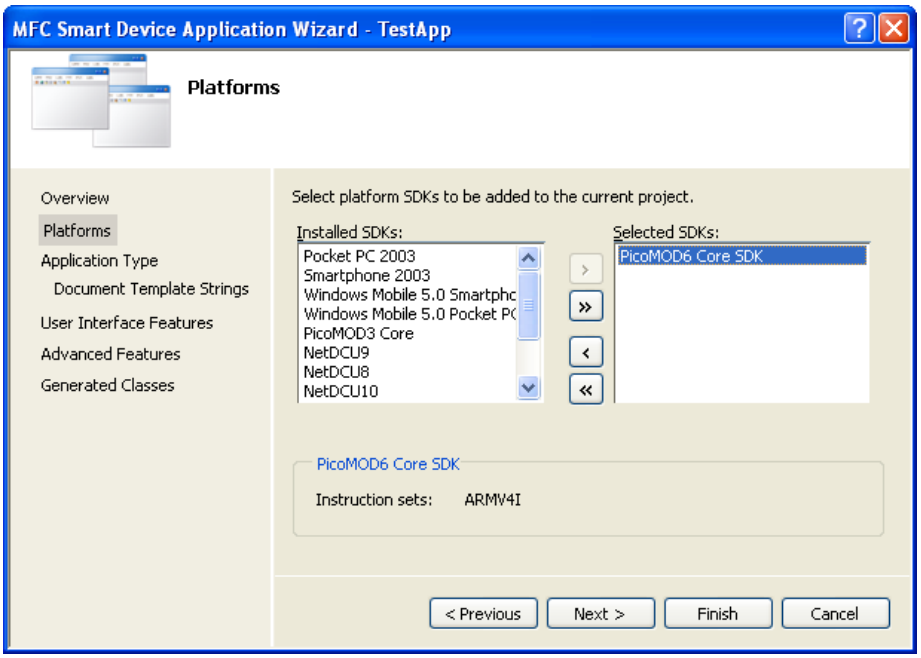


Figure 18: SDK for native smart device

6.2.1 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 efus 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:

To debug via Ethernet, you first need to copy the following files from your development-PC in "Drive:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\<CPU>" to the flash memory of the efus. 'X' stands for the version of the file:

- Clientshutdown.exe
- ConmanClientX.exe
- CMaccept.exe
- eDbgTL.dll
- TcpConnectionA.dll

The process now is almost the same as in chapter 6.3 Visual Studio 2012/2013. Only the device-IP is entered somewhere else. In Visual Studio 2005/2008 you open *Tools → Options*, search for the CPU under *Device Tools → Devices* and get into the Properties of it. Click on *Configure* on the right side of *TCP Connect Transport*. Now you can enter the *specific IP address*.

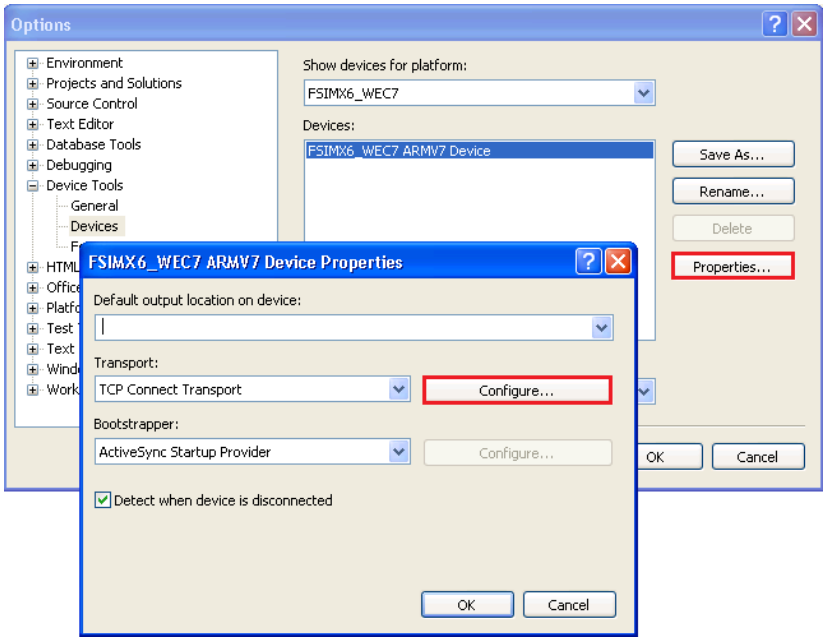


Figure 19: Location to enter Device-IP

6.3 Visual Studio 2012/2013

Before you can develop software you have to download the SDK for efus and install it on your PC. With this SDK you also get the PC tool FSDevicSpy.exe. Start this tool before you start the efus. Network interface is configured for DHCP by default. If you don't have a DHCP server in your network you have to do the steps in chapter 5.4.4 Network interface configuration with NDCUCFG.

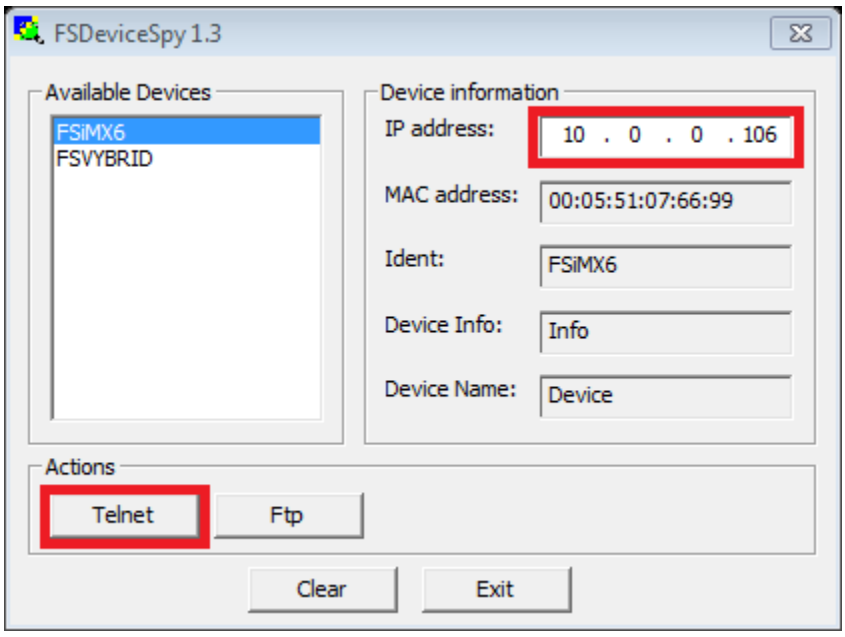


Figure 20: Start PC application FSDevcieSpy to make Telnet connection

As already noted, with Compact 2013 you need a TCP/IP connection between your development PC and efus. To establish this connection you need access to command shell of efus. This can be done by opening a Telnet connection.
At the command shell of efus you must start conmanclient3.exe and cmaccept3.exe.

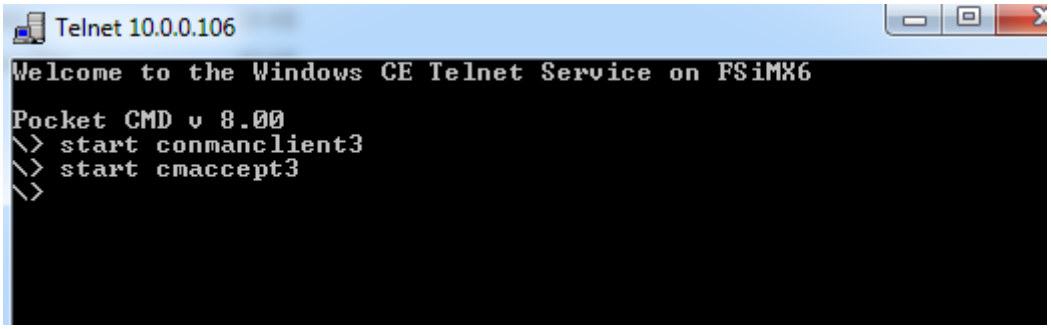


Figure 21: Start conmanclient3 and cmaccept3

You should start deploying your application or debugging from within Visual Studio now. Don't wait too long because cmaccept3 will time out. Enter IP-Address of efus if asked or changed.

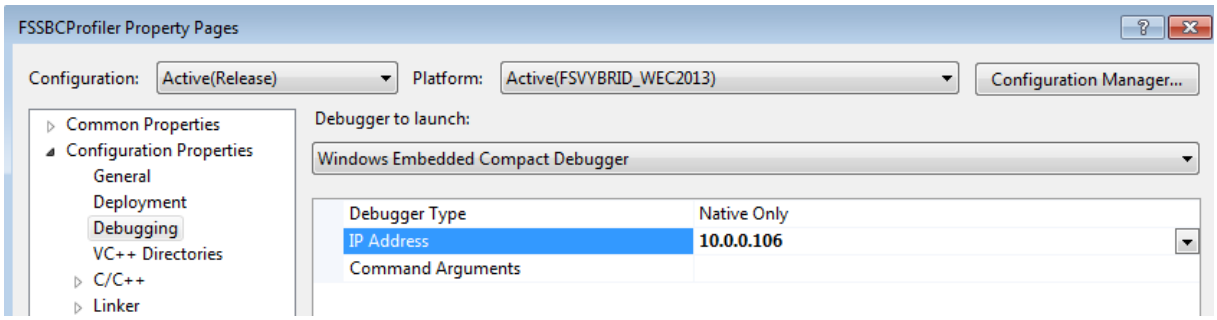


Figure 22: Enter IP in Visual Studio

6.4 Visual Studio 2012/2013 – Managed Code

The application programmer can develop the application in C# or VB.NET using the Compact Framework 3.9 which is part of the Windows CE kernel for efus.

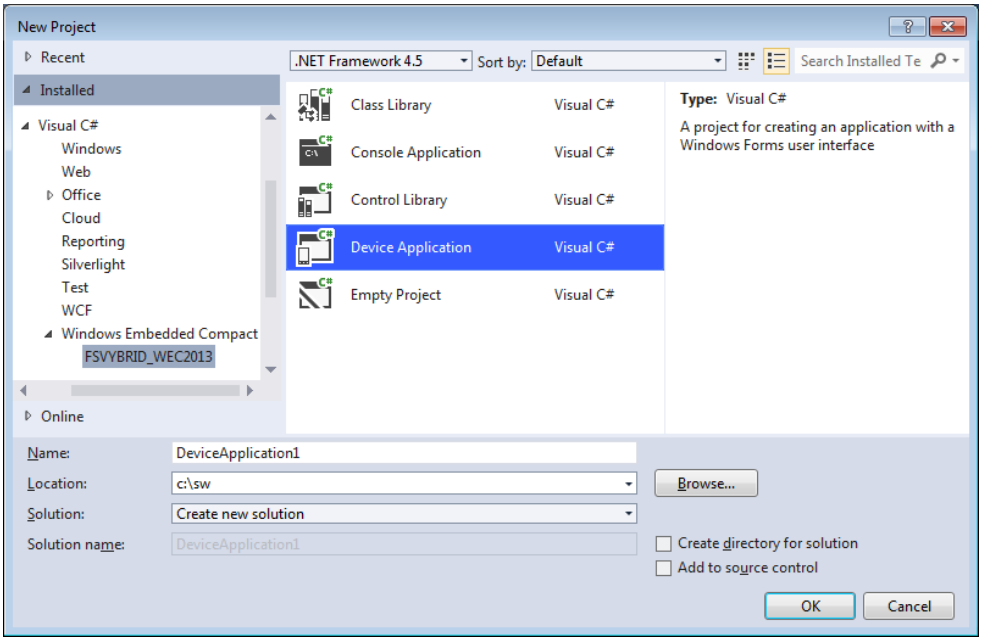


Figure 23: New managed Windows Compact 2013 application

6.5 Visual Studio 2012/2013 – Native Code

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

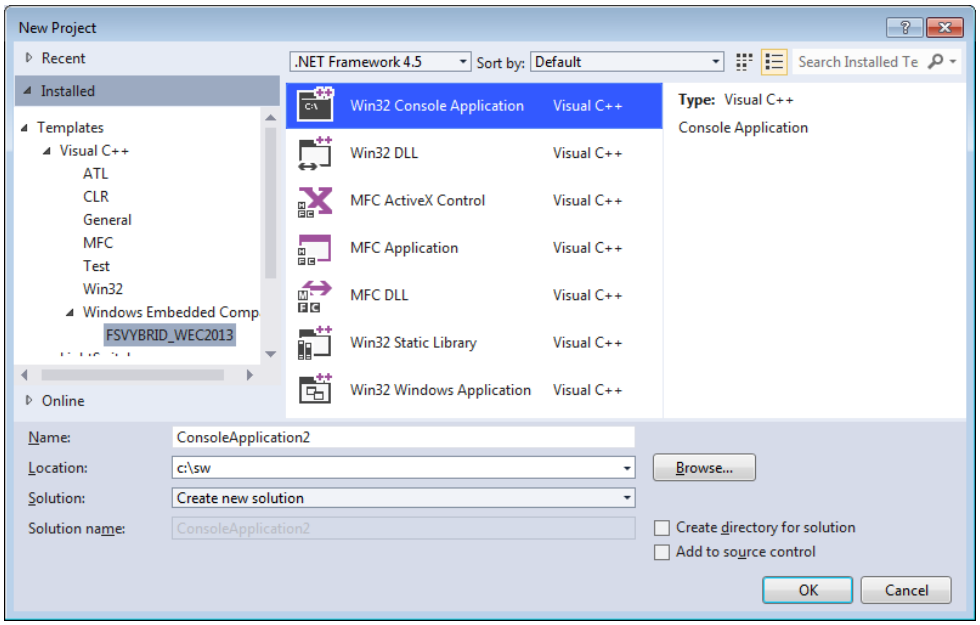


Figure 24: New native Windows Compact 2013 application

7 Remote Tools

Microsoft Visual Studio 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

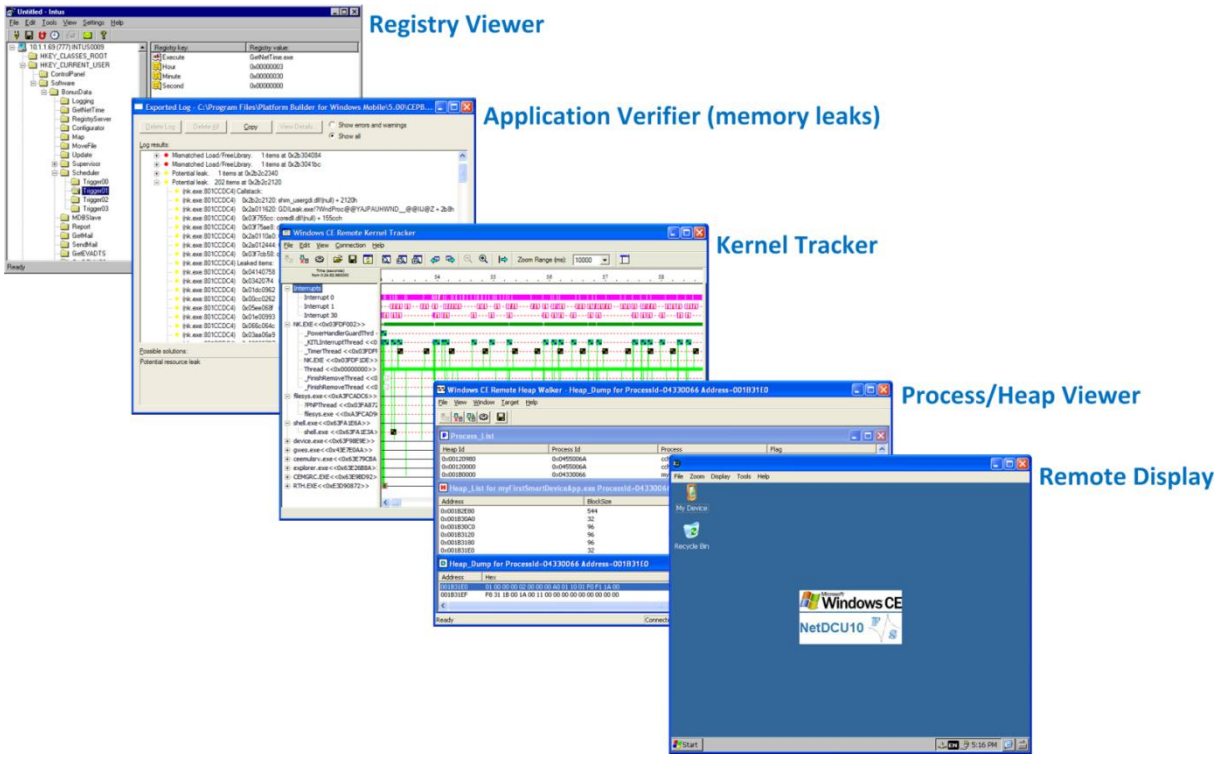


Figure 25: Visual Studio Remote Tools

With Visual Studio 2012/2013 there is also a new generation of remote tools. All remote tools are collected in program “Remote Tools Shell”.

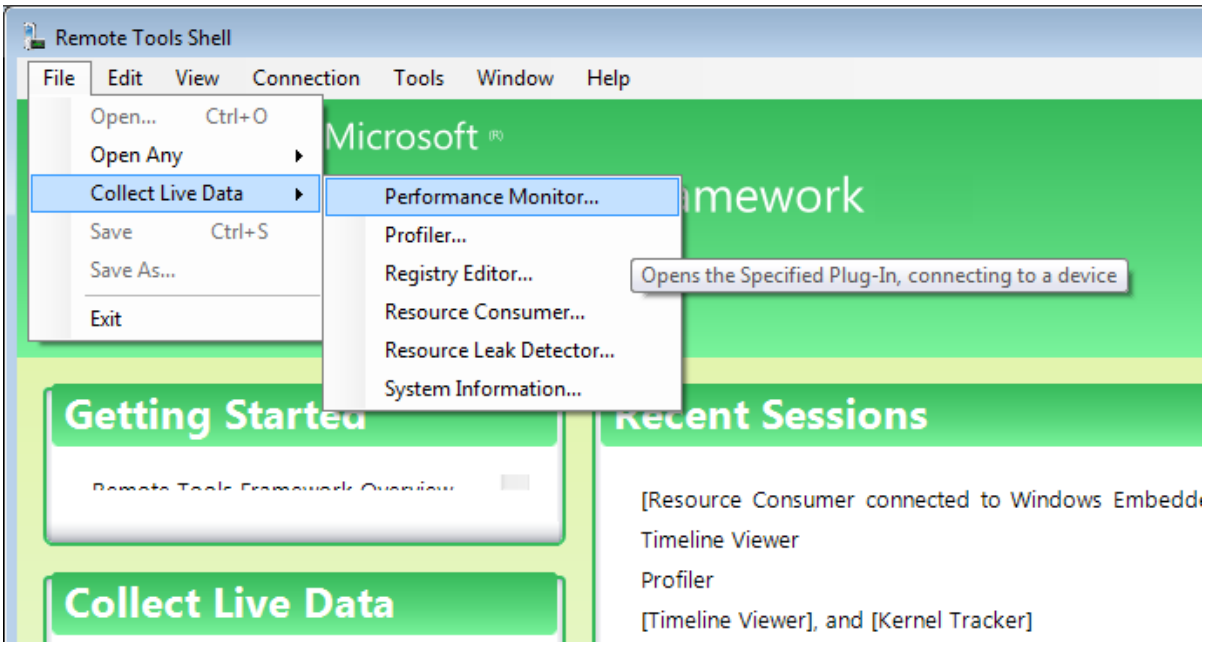


Figure 26: Visual Studio 2012/2013 Remote Tools Shell

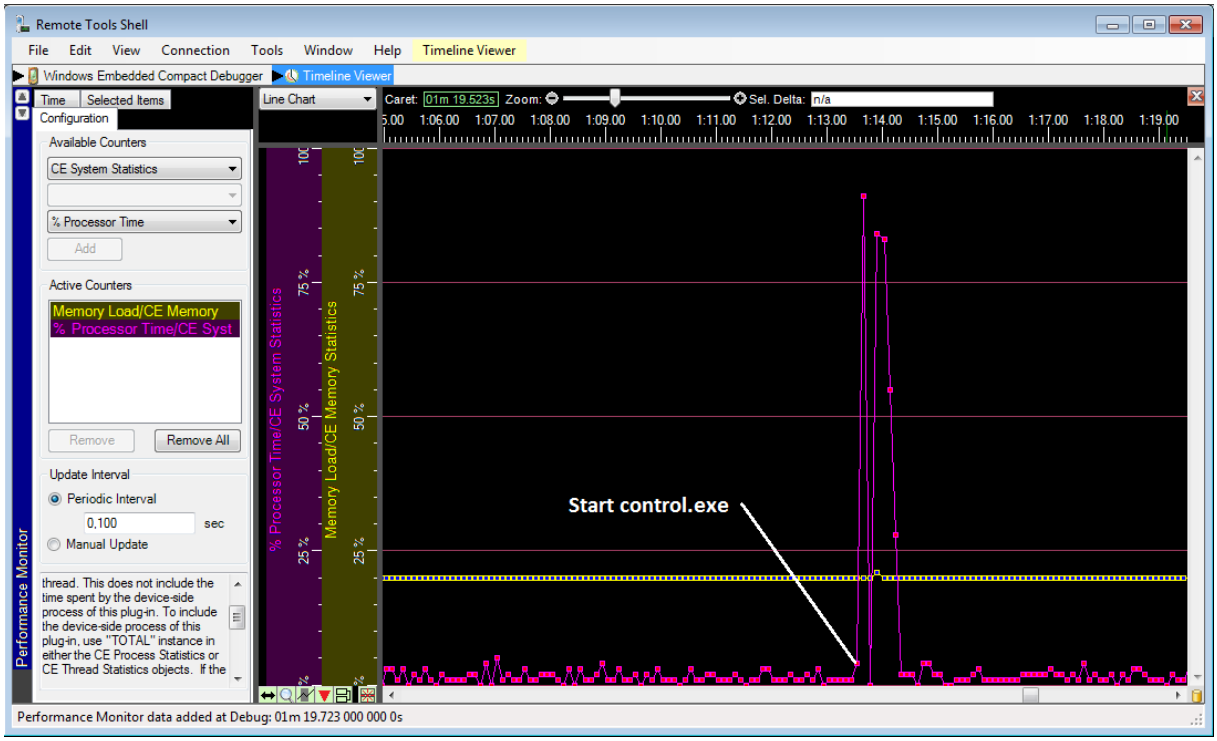


Figure 27: Visual Studio 2012/2013 TimeLine Viewer

8 Firmware Update

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

8.1 The NetDCU-USBLoader utility

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

When connecting efus and NetDCU-USBLoader for the very first time (see chapter 3) 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 8.2, 8.3 and 8.5.

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 ***FSBoardUSB.inf*** which can be found under *<InstallationPathOfNetDCUCUSBLoader>\Driver.*

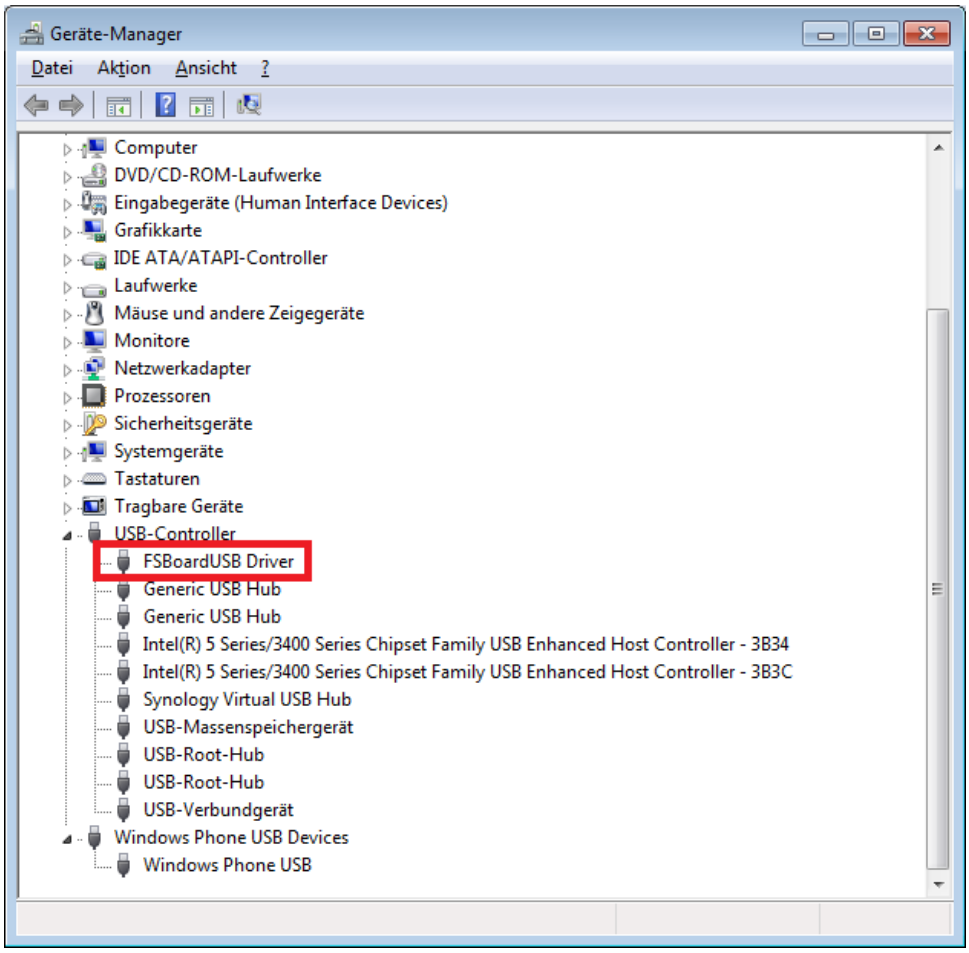


Figure 28: NetDCU-USBLoader driver installation (1)

8.2 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 continuously **<SHIFT> + 's'** while powering on the efus.

8.2.1 Preparing EBoot to download Kernel Image

To store 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.

```
efusA9 - WindowsCE Bootloader

:> MF

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

Listing 14: Preparing EBoot for download

In some cases it is necessary to adapt the size of the partitions. Please read chapter 8.2.4 Partitioning the Flash storage.

8.2.2 Download via USB

To download the WindowsCE Kernel by USB use the command `DU` (1) 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 (2). Select the respective `<NK-kernel-image>.bin` file (3) and click on 'Start' button (4). You should now see progress similar to Figure 29.

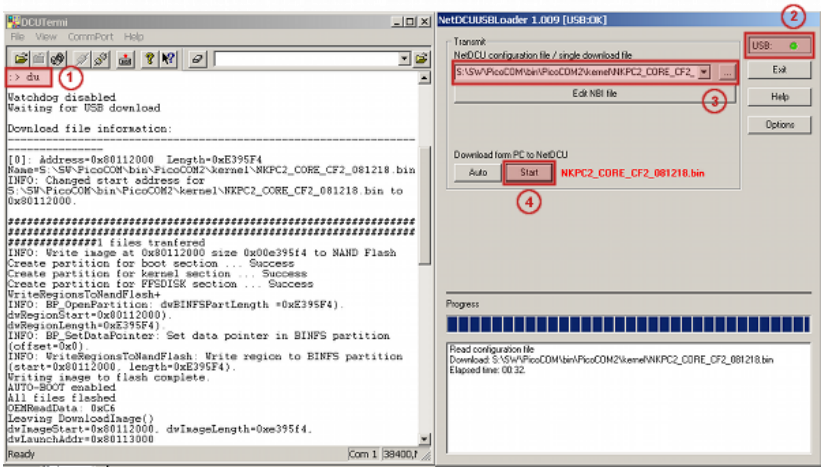


Figure 29: Using the NetDCU-USBLoader utility

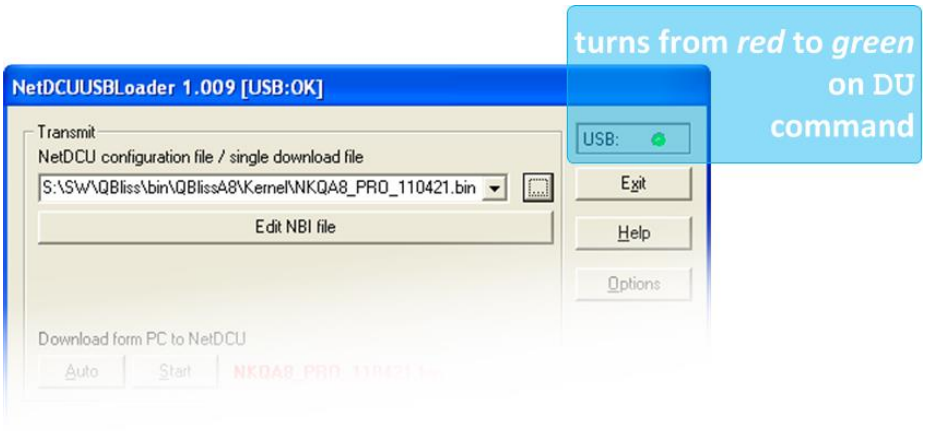


Figure 30: Using the NetDCU-USBLoader utility (2)

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

```
efusA9 - WindowsCE Bootloader

:> MF

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

:> DU
HW-Watchdog: OFF
Waiting for USB download
Connected at high-speed
Connected at high-speed
```

Next listing shows the messages while downloading the binary (after the NDCUUSBLoader *start button* (4) has been activated):

```
Download file information:
[0]: Address=0x80100000  Length=0x31DBD48
Name=S:\SW\FSiMX6\WEC2013\Kernel\XIPVYB_C8E_140414.bin
BIN detected. Check MinImageStart
(S:\SW\FSiMX6\WEC2013\Kernel\XIPVYB_C8E_140414.bin)
INFO: Changed start address for
S:\SW\FSiMX6\WEC2013\Kernel\XIPVYB_C8E_140414.bin to 0x80100000.
######
1 files transferred
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+

Writing single region/multi-region update, dwBINFSPartLength:
52280648
INFO: BP_OpenPartition: dwBINFSPartLength =0x31DBD48).
dwRegionStart=0x80100000).
dwRegionLength=0x31DBD48).
```

```
INFO: BP_SetDataPointer: Set data pointer in BINFS partition
(offset=0x0).
INFO: WriteRegionsToNandFlash: Write region to BINFS partition
(start=0x80100000, length=0x31DBD48).
Writing image to flash complete.
AUTO-BOOT enabled
All files flashed
```

Listing 15: Download Image using USB interface

After the download operation has finished and the kernel has been started, the Windows desktop should appear on the connected display.
On serial debug line, the NDCUCFG utility will be activated by default. EBoot menu is no longer prompted on start up. to get more details about the NDCUCFG utility.
Additionally ActiveSync should open a connection- if the efus 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

8.2.3 Download via Ethernet

In addition to using NetDCU-USBLoader to update WindowsCE kernel image via USB, the Microsoft utility called Eshell can be used to download a kernel image via Ethernet. For this purpose, enter `DE` command within Eboot and start the Eshell utility on your development machine. As soon as efus 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 efus.

8.2.4 Partitioning the Flash storage

Depending on the Windows CE kernel image you are going to run on efus, the default size of kernel partition might not be sufficient for your needs and it might be required to repartition the flash storage.

There are up to 3 partitions possible on efus:

- OS-Image (BINFS): The WindowsCE kernel is stored in this partition.
- FFSDISK: These partitions can be used to store user data and applications. It is available under '\FFSDISK' on a running WindowsCE system.
- SECOND Partition: The extended partition must be administrated in WindowsCE.

There are no partitions available by default, so you have to create them using theStorage-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 get a list of all available EBoot commands. Partitioning is setup with the command `P`.

```
:> P
------------------PARTITION CONFIGURATION--------------
  Current settings:
  Total : 128 MB
  OS-Image: 64 MB
  FFSDISK : 64 MB, Part type: FAT
  SECOND : 0 MB, Part type: EXTENDED
  Enter maximal size for OS-Image [64]:
```

Listing 16: Flash partitioning (part1)

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

Listing 17: Flash partitioning (part2)

You can ignore the FMD messages. Some blocks are reserved for storing the bootloader and you should not worry about these messages.
Please wait for Eboot prompt (:>) to return before continuing.

8.3 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 holding **<SHIFT>+'s'** while powering on the efus. 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 efus device.
Enter NBoot by holding 's' while powering the device. You will see output like:

```
F&S Nand Loader VN08 built Apr  2 2014 08:57:45
efusA9 Rev. 1.10
1024MB RAM 128MB FLASH (2cf1) 792MHz
booted primary image from page 4
serial number not set

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 18: 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 (ebootimx6_v007.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 19: 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
```

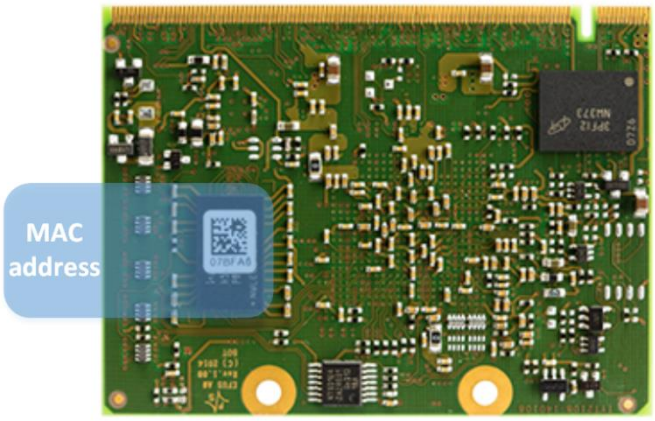


Figure 31: efus MAC address

If double characters are appearing and you find it quite irritating, you may want to turn off “Local Echo” (see Figure 8: DCUTerm.exe configuration: View → Options).

After MAC setup you will enter efus 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 efus download area.

8.4 Configure EBoot

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

```
efusA9 - WindowsCE Bootloader

:> ?
efusA9 - 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 20: EBOOT command shell

8.5 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 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.
Specific testing of all parameters of each device is not necessarily performed unless required by law or regulation.
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.
Specifications are subject to change without notice.

Warranty Terms

Hardware Warranties
F&S guarantees hardware products against defects in workmanship and material for a period of two (2) years from the date of shipment. Your sole remedy and F&S's sole liability shall be for F&S, at its sole discretion, to either repair or replace the defective hardware product at no charge or to refund the purchase price. Shipment costs in both directions are the responsibility of the customer. This warranty is void if the hardware product has been altered or damaged by accident, misuse or abuse.

Software Warranties
Software is provided "AS IS". F&S makes no warranties, either express or implied, with regard to the software object code or software source code either or with respect to any third party materials or intellectual property obtained from third parties. F&S makes no warranty that the software is useable or fit for any particular purpose. This warranty replaces all other warranties written or unwritten. F&S expressly disclaims any such warranties. In no case shall F&S be liable for any consequential damages.

Disclaimer of Warranty
THIS WARRANTY IS MADE IN PLACE OF ANY OTHER WARRANTY, WHETHER EXPRESSED, OR IMPLIED, OF MERCHANTABILITY, FITNESS FOR A SPECIFIC PURPOSE, NON-INFRINGEMENT OR THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION, EXCEPT THE WARRANTY EXPRESSLY STATED HEREIN. THE REMEDIES SET FORTH HEREIN SHALL BE THE SOLE AND EXCLUSIVE REMEDIES OF ANY PURCHASER WITH RESPECT TO ANY DEFECTIVE PRODUCT.

Limitation on Liability
UNDER NO CIRCUMSTANCES SHALL F&S BE LIABLE FOR ANY LOSS, DAMAGE OR EXPENSE SUFFERED OR INCURRED WITH RESPECT TO ANY DEFECTIVE PRODUCT. IN NO EVENT SHALL F&S BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES THAT YOU MAY SUFFER DIRECTLY OR INDIRECTLY FROM USE OF ANY PRODUCT. BY ORDERING THE PRODUCT, THE CUSTOMER APPROVES THAT THE F&S PRODUCT, HARDWARE AND SOFTWARE, WAS THOROUGHLY TESTED AND HAS MET THE CUSTOMER'S REQUIREMETS AND SPECIFICATIONS

Listings

Listing 1: Bootup: Bootloader....13
Listing 2: Bootup: Read image from NAND flash....13
Listing 3: Bootup: Start Windows CE....14
Listing 4: Bootup: Loading drivers....14
Listing 5: Bootup: Flash initialization....14
Listing 6: NDCUCFG command prompt....18
Listing 7: NDCUCFG help command....19
Listing 8: IP-configuration from command line....21
Listing 9: NDCUCFG: Registry commands....22
Listing 10: NDCUCFG: Opening TcpIp settings....22
Listing 11: NDCUCFG: Changing TcpIp settings....23
Listing 12: NDCUCFG: Save modifications permanently....23
Listing 13: NDCUCFG: List Devices....26
Listing 14: Preparing EBoot for download....38
Listing 15: Download Image using USB interface....40
Listing 16: Flash partitioning (part1)....41
Listing 17: Flash partitioning (part2)....41
Listing 18: NBOOT command shell....43
Listing 19: NBOOT after download of EBOOT....43
Listing 20: EBOOT command shell....45

Figures

Figure 1:efusA9 on carrier board (top side) 6
Figure 2:efusA9 on carrier board (bottom side) 6
Figure 3: Download Documents from F&S website..... 8
Figure 4: Register with F&S website..... 9
Figure 5: Unlock software with the serial number 10
Figure 6: Changelog and Roadmap.....10
Figure 7: DCUTerm.exe configuration: CommPort → Settings 11
Figure 8: DCUTerm.exe configuration: View → Options 11
Figure 9: Opening the serial connection in DCUTerm.exe 12
Figure 10: NDCUCFG over serial debug connection 15
Figure 11: FSDeviceSpy.....16
Figure 12: Telnet connection 17
Figure 13: FTP connection 17
Figure 14: Browsing the device' file system with Active Sync 18
Figure 15: Initial screen of the touch panel calibration program. 24
Figure 16: New managed smart device application 28
Figure 17: New native smart device application.....29
Figure 18: SDK for native smart device 29
Figure 19: Location to enter Device-IP 30
Figure 20: Start PC application FSDevcieSpy to make Telnet connection 31
Figure 21: Start conmanclient3 and cmaccept3.....31
Figure 22: Enter IP in Visual Studio.....32
Figure 23: New managed Windows Compact 2013 application 33
Figure 24: New native Windows Compact 2013 application 34
Figure 25: Visual Studio Remote Tools 35
Figure 26: Visual Studio 2012/2013 Remote Tools Shell.....36
Figure 27: Visual Studio 2012/2013 TimeLine Viewer 36
Figure 28: NetDCU-USBLoader driver installation (1).....37
Figure 29: Using the NetDCU-USBLoader utility.....38
Figure 30: Using the NetDCU-USBLoader utility (2) 39
Figure 31: efus MAC address.....44