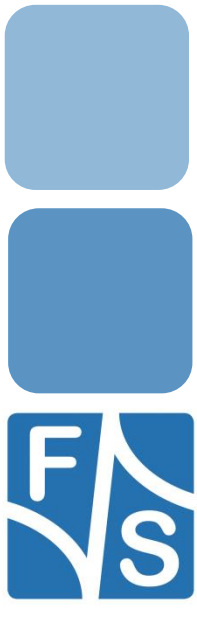
Boot Screen

NetDCU, PicoMOD, PicoCOM, efus, QBliss and armStone with Windwos Embedded CE Windows Embedded Compact

Version 1.4
2017-07-04



Elektronik Systeme

Untere Waldplätze 23
D-70569 Stuttgart
Fon: +49(0)711-123722-0
Fax: +49(0)711 – 123722-99

History

Date	V	Platform	A,M,R	Chapter	Description	Au
2009-04-15	1.0	*	A	*	First Version	MK
2009-05-18	1.1	*	M	1, 3.3.4	Using customized display settings adapted	MK
2009-08-13	1.2	*	A	3.3	Ndcucfg comman to download a boot image added	MK
2016-02-29	1.2	*	M	*	Changed to new corporate design	JG
2016-03-01	1.3	*	M	*	Change available boards and bitmap format information	MA
2017-07-04	1.4	*	M	*	Minor corrections	HF

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

About this document

This document describes how to install a boot screen on a NetDCU, PicoMOD, PicoCOM, efus, QBliss or armStone. In principle the procedure for establishing a boot screen is similar on all boards from F&S. There are only some small differences that are emphasized within this documentation.

Table of Contents

1 Introduction

Booting a WindowsCE kernel usually takes a few seconds. By default a connected display will stay blank until the operating system has booted and either the desktop or the application is displayed on the screen. Unlike showing a blank screen the boot up time can be used to display some information about the device or application. This feature is available for most F&S boards, optionally. Basically the boot screen is a bitmap that will be stored within the boot loader and loaded automatically into the display frame buffer during boot. Displaying dynamic data hence is not possible currently. The boot screen will be stored within the boot loader. For that reason WindowsCE has no access to this data and the boot screen will still be available, after updating the WindowsCE kernel image.
As a basic requirement the display interface must be configured by the boot loader to be able to display the boot screen. By default (when not using the boot screen) this is arranged by the WindowsCE display driver. To avoid a re-initialization of the display interface, which leads into display interferences, this behavior can be configured within the bootloader.
This following chapters will describe step by step how to establish such kind of boot screen.

Note:
Please note that the boot screen feature cannot be used without modifications. For that reason it is available as an add-on software package. To purchase this functionality please contact sales@fs-net.de .

2 Preparing the boot image

2.1 Requirements

The Boot screen feature is an add-on package for all boards from F&S**. A customized bootloader configuration will be required to get a boot screen working** in your application.

2.1.1 Bootloader version

The Following table shows the version requirements of the bootloader to be used depending for each board. Older versions may not support the boot screen functionality.

Board	Required bootloader version
NetDCU5.2	V1.06
NetDCU6	V1.08
NetDCU8	V1.06
NetDCU9	V1.06
NetDCU10	V1.06
NetDCU11	V1.06
PicoMOD1	V1.06
PicoMOD3	V1.06
PicoCOM2	V1.01
All newer boards already have the right bootloader version.	

Table 1: Bootloader version required of the boot screen feature.

Note:
The latest bootloader version can be download from fs-net.de.

2.1.2 Bitmap format

The size of the bitmap is restricted to **1 MB**. The color depth must be **8/16 bit**. For that reason heavy color gradients should be avoided.
The image must have a raw bitmap (bmp) format. Still having the possibility to use complex and pretentious screens, the images can be RLE-compressed.

2.2 Create the bitmap

The bitmap can be created using any image editing program, but there are two additional programs from F&S that can be used to optimize the boot screen bitmap.

2.2.1 Optimize the bitmap palette

To optimize the colour palette of the image specifically for the board the program `BMPcnv2` can be used. The basic palette is included to the converting program (WIN256.PAL).

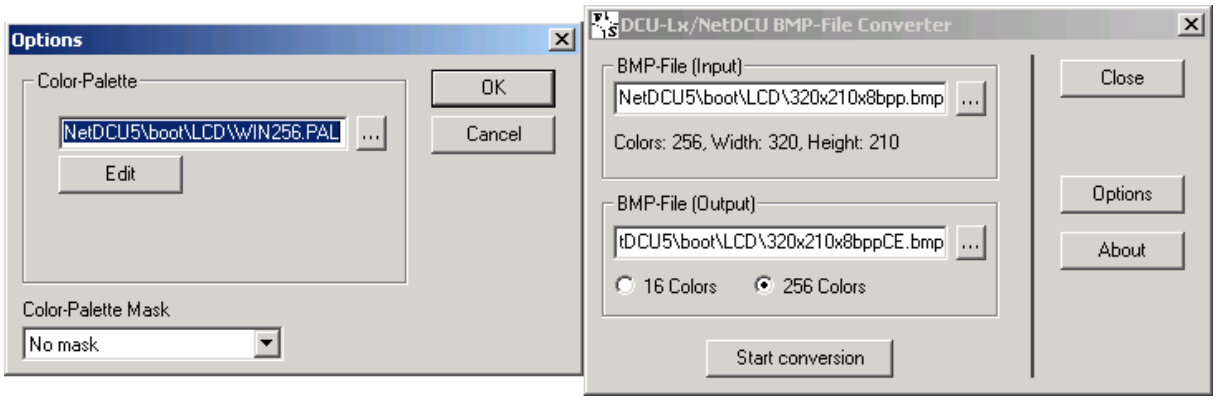


Figure 1: BMPcnv2 screenshoot.

2.2.2 RLE compression

To compress the bitmap using the RLE compression the program `BMPtoRLE` can be used.

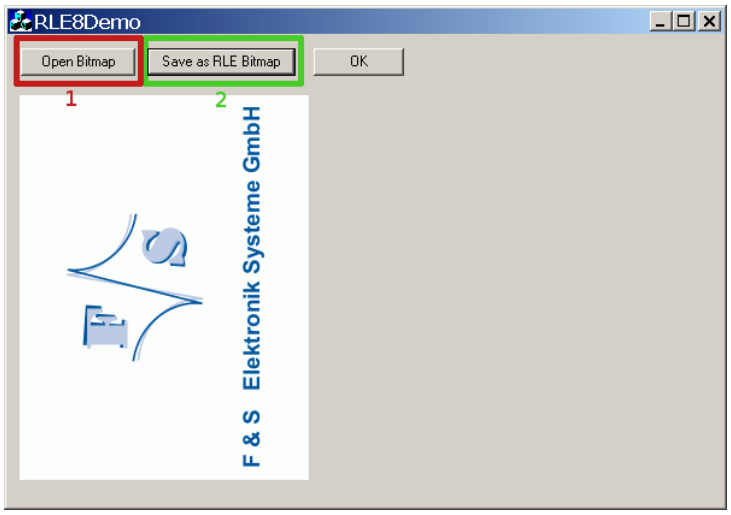


Figure 2: BMPtoRLE screenshoot

Simply open your image *(1)* within `BMPtoRLE` and save *(2)* it at a different location. The size of this image should now be less than the original.

3 Installing the boot screen

Except the NetDCU5.2 and NetDCU6 the NetDCU-USBLoader is used to transfer data to the board. The USBLoader thereby can not only be used to load new bootloader or kernel images, it even is prepared to load boot screen images onto the board. If you are not familiar in using the USBLoader already, please refer to the FirstSteps documentation of your corresponding board.

NetDCU5.2/NetDCU6:
The bitmap must be transfer using the serial debug line. This feature is still available for other boards (→ chapter 3.2).

3.1 Download boot image using USBLoader

Similar to loading a kernel or bootloader image the download process must be prepared in bootloader (command “DU”). If the boot image is the only file that should be transferred, it can directly be selected within the USBLoader dialog (…). To include the bitmap to a regular download process the NBI file must adapted similar to the following example:

```
; This section contains all files who should be transfered.
[Files]

; Boot screen.
BMP=C:\temp\FundS 240 320.BMP
Listing  1: Example NBI  entries to download a boot screen.
The debug output on the serial debug console should look like the following:
:> du
Watchdog disabled
Waiting for USB download
Download file information:
---------------------------------------------------------------------
[0]: Address=0x0  Length=0x12E76  Name=splash.bmp
INFO: Changed start address for splash.bmp to 0x81FD8000.

1 files tranfered
WriteBootBMP: Write ... Done
All files flashed
OEMReadData: 0x6
Leaving DownloadImage()
dwImageStart=0x0, dwImageLength=0x0, dwLaunchAddr=0x0
Press >S< to step into monitor...

Watchdog disabled

PICOCOM2 - WindowsCE Bootloader
:>
```

Listing 1: Debug output for downloading a boot screen image

3.2 Download boot image over serial

If the USBLoader functionality is unavailable or cannot be used, all bootloaders support downloading a boot image over the serial debug line. It is recommended to use DCUTermi for this purpose. To start this process the command “XWG” must be used. After initiating the download, the exact size of the bitmap must be entered. Afterwards the image must be transferred in binary mode (“File” → “Transmit binary file”).

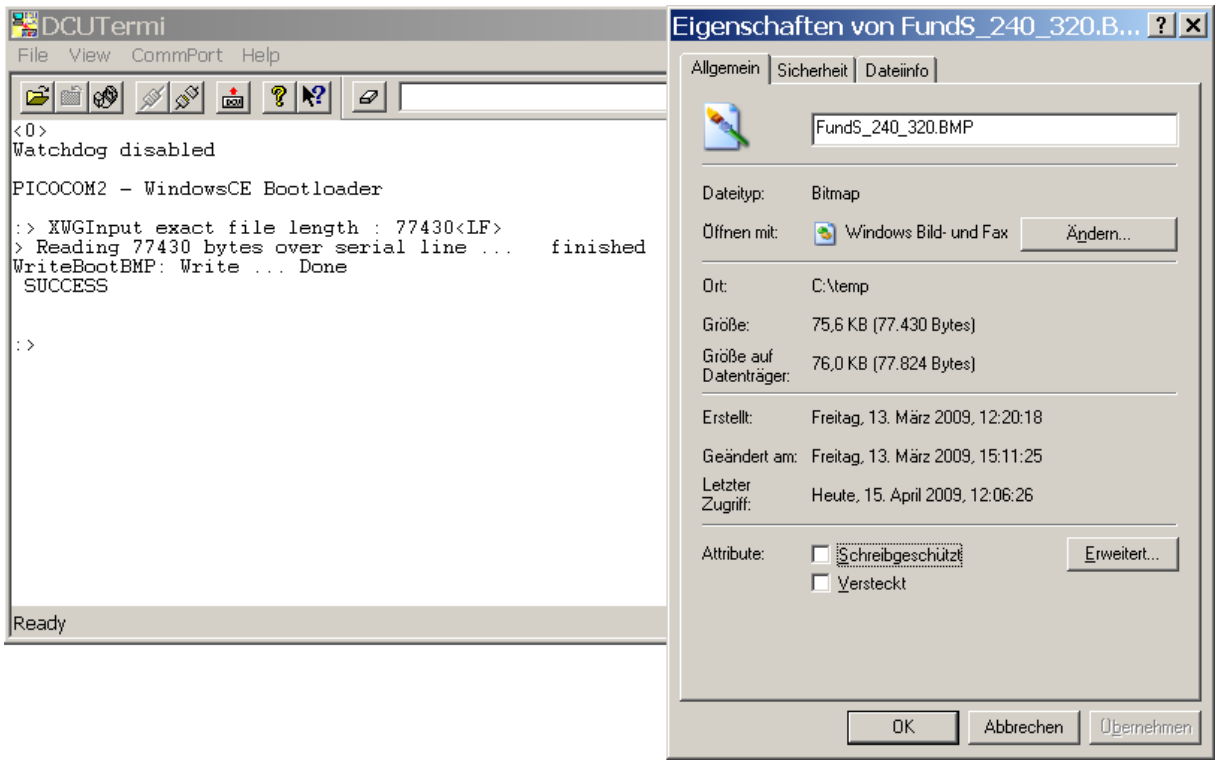


Figure 3: Download boot image over serial.

3.3 Download boot image out of WindowsCE

On most platforms there also is the possibility to update the bootscreen image out of a running WindowsCE system. Therefore the ndcucfg utility must be used. There is another hidden command available called "`boot bitmap write`". The following listing shows an example of how to use this command.

```
NetDCU Config Utility Ready
Version: 038
Type help for commands
!>boot bitmap write \FFSDISK\FundS BootScreen rle.bmp<LF>
OK
!>
```

Listing 2: Download boot image using NDCUCFG.

This especially is reasonable in combination with the update package from F&S, used to update applications and the WindowsCE kernel.

3.4 Bootloader configuration

All bootloaders include a basic set of display configurations. You will find these modes in the display driver documentation of the corresponding board. In most cases these configurations may not fit to your display. Included to this boot screen add-on package you will receive a customized bootloader configuration, including the display configuration set, fitting your needs. This configuration set is available in terms of a PSD file, which must be downloaded to the board using the USBLoader utility. To do so a customized NBI file **must** be created. Selecting the File directly within the USBLoader is not possible. Please create a NBI file similar to the following one:

```
; This section contains all files who should be transfered.
[Files]
PSD=C:\temp\PC2_Bootscreen_FundS-QVGA_V100.PSD
```

Listing 3: Example NBI file to download a bootloader configuration set (PSD-File).

After downloading this file, your customized display configuration will be available at **mode 20**. Please proceed with the next two chapters, describing the further steps to test and enable the boot screen functionality.

Note:
If you did not obtain a PSD file already, please contact our support team (support@fs-net.de).

Note:
On NetDCU5.2 and NetDCU6 downloading a PSD-File is not supported. If you are going to establish a boot screen for one of these boards please contact our support team (support@fs-net.de). We will assist you individually.

3.4.1 Testing the boot screen

The boot screen can be tested using the “hidden” command “XDT”. After typing the mode to be used, the image you download before should be displayed on the screen.

```
:> XDT
Input LCD mode [0] :20<LF>
InitLCD...
Display-Mode: 20, Name Hitachi TX09, 240x320, bpp=16, Voltage=3.3V
Width: 240 Height: 320 Bpp: 16
frame buffer phys: 0x20002000, virt: 0xA0002000
BMP: Width=240, Height=320, Compression=0, SizeImage=76800, Virtu
lStride=480 SCREEN: BPP 16
InitLCD done
:>
```

Listing 4: Testing the boot screen.

3.4.2 Enabling the boot screen permanently

After download the image the display-initialization is still disabled in bootloader. To enable it the command “XDE” is used. The boot screen will then automatically displayed every time the board boots up.

```
:> XDE
Enable/Disable LCD during boot [N] (Y/N) ? :Y
 > Current LCD mode: 20
:>
```

Listing 5: Enabling the boot screen permanently.

3.4.3 Display initialization

As already mentioned in the introduction the display will be initialized by the display driver of the WindowsCE kernel by default. If the display controller is already running at this point, this can lead into display interferences. For that reason the re-initialization can be disabled in bootloader with the command “XDI”.

```
:> XDI
Enable/Disable WindowsCE display initialization [Y] (Y/N) ? :N
 > Current LCD mode: 20
:>
```

Listing 6: Disabling the display initialization in WindowsCE.

4 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 one (1) year 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

Figures

Tables