# F&S i.MX6-UltraLite Linux

## *First Steps*

Version 2.5
(2023-12-20)
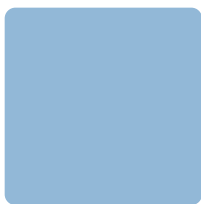
armStone    efus    NetDCU

PicoCOM    PicoCore    PicoMOD    QBliss

**F&S Elektronik Systeme**

# About This Document

This document shows how to bring up F&S boards and modules under Linux and how to use the system and the devices. It is intended to help you get to know the board so that you can take the first steps to try out and test all the peripherals.

## Remark

The version number on the title page of this document is the version of the document. It is not related to the version number of any software release! The latest version of this document can always be found at http://www.fs-net.de.

## How To Print This Document

This document is designed to be printed double-sided (front and back) on A4 paper. If you want to read it with a PDF reader program, you should use a two-page layout where the title page is an extra single page. The settings are correct if the page numbers are at the outside of the pages, even pages on the left and odd pages on the right side. If it is reversed, then the title page is handled wrongly and is part of the first double-page instead of a single page.
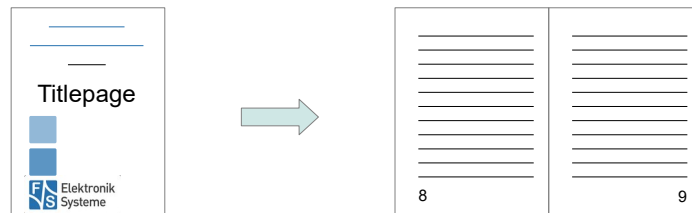


## Typographical Conventions

We use different fonts and highlighting to emphasize the context of special terms:

`File names`

*Menu entries*

```
Board input/output
```

`Program code`

```
PC input/output
```

`Listings`

```
Generic input/output
```

`Variables`

# History

| Date | V | Platform | A,M,R | Chapter | Description | Au |
|---|---|---|---|---|---|---|
| 2015-11-30 | 1.0 | fsimx6ul | M | * | Ported from fsimx6 to fsimx6ul; many small modifications regarding i.MX6UL | HK |
| 2015-12-02 | 1.0 | fsimx6ul | A | 4 | Linux Device Trees | HK |
| 2015-12-03 | 1.0 | fsimx6ul | A | 5.3.5 | Handling Device Trees in U-Boot | HK |
| 2016-08-19 | 1.1 | fsimx6ul | A,M,R | * | Address differences between multiple documents | PH |
| 2016-09-14 | 1.1 | fsimx6ul | A, M | 2.3, 3.3.1, 7.14, 8.2, 8.5 | Change NBoot upgrade info, formatting, add mkimage tool installation | PH |
| 2017-06-28 | 1.1 | fsimx6ul | A,M,R | 7.*, 8 | Add chapter 8 'Graphical Environment' merge similar subchpters and remove them | PH |
| 2017-11-14 | 1.3 | fsimx6ul | M | 6.2.* <br> 9.3 | Command should be nand erase.part <br> Explain new version of install-sources.sh with option --dry-run | PH <br> HK |
| 2018-02-19 | 1.3 | fsimx6ul | M | 3.3.5 | Add efusa7ull as platform for efusA7UL with i.MX6ULL | HK |
| 2018-05-30 | 1.4 | fsimx6ul | M <br> M <br> A <br> M <br> M <br><br><br> M <br> A | 1 <br> 2 <br> 2.1.2 <br> 2.2, 2.3 <br> 3.3 <br><br><br> 9.1 <br> 9.2 | Add NetDCUA9 and PicoCOM1.2 to available platforms <br> Add reference to F&S development machine <br> Add PicoCOM1.2 <br> Mention Yocto releases, add Yocto file names to content list <br> Change manual installation procedure to use file names from sdcard directory; they are the same for Buildroot and Yocto <br> Collect Buildroot build process chapters under 9.1 <br> Add chapter 9.2 with Yocto build process | HK <br> HK <br> HK <br> HK <br> HK <br><br><br> HK <br> HK |
| 2019-03-15 | 1.5 | fsimx6ul | M <br><br> A <br> A <br> R <br> M <br><br> M | 1 <br><br> 1.1 – 1.3 <br> 2.1.3 <br> 2.3 <br> 2.2, 2.3, 3.2, 9.1 <br> 2.3, 3.3, 5.1, 5.3, 6.6, 9.1, 9.2 | Add PicoCoreMX6UL and PicoCoreMX6SX to plaforms, drop fss5pv210 and picocom4 architectures, add fsimx7ulp <br> Split into chapters, add chapter with links to F&S website <br> Add PicoCoreMX6UL <br> Remove hardware documents from list of contents <br> Use generic <v> for versions in most places, explain old & new version numbers <br> Several replacements: uImage → zImage, fs-toolchain-5.2.0 → fs-toolchain-7.4, u-boot-2014.07 → u-boot-2018.03, linux-4.1.5 → linux-4.9.88, buildroot-2016.05 → buildroot-2018.11, 80800000 → 81000000, 81000000 → 82000000 | HK <br><br> HK <br> HK <br> HK <br> HK <br><br> HK |
| 2019-03-18 | 1.6 | fsimx6ul | M | 1.3 | Explain available documents in more detail | HK |
| 2019-11-14 | 2.0 | fsimx6ul | M | * | Large parts are now part of LinuxOnFSBoards and can be removed here. Reduce to startng the board, automatic installation if software is not installed and using the peripherals. Use new board-logos on title page. Use new images for PicoCOM1.2 and PicoCoreMX6UL. Several enhancements in the chapter for using the Linux system. | HK |
| 2020-03-26 | 2.1 | fsimx6ul | M | 4.16 | Add distinguish between Yocto and Buildroot | PG |
| 2020-07-22 | 2.2 | fsimx6ul | A | 1.1 <br> 2.1.3 <br> 5.2.3 | Add chapters about PicoCOMA7 | PG |
| 2020-08-19 | 2.3 | fsimx6ul | M <br> M <br> M <br> A | 3.1 <br> 3.3 <br> 3.4 <br> 3.5 | Add description for board specific naming and eMMC devices. <br> Erase command on eMMC only devices currently not supported <br> Add description for creating partitions in NBoot <br> Add description for mounting partitions in NBoot over USB | KM <br> KM <br> KM <br> KM |
| 2021-08-04 | 2.4 | fsimx6ul | A | 1.1 <br> 2.1.5 <br> 5.2.5 | Add chapters about PicoCoreMX6UL100 | PJ |
| 2023-12-20 | 2.5 | fsimx6ul | R | 4.0 | Remove chapter 4 "Using the Standard System and Devices" It can be found in LinuxOnFSBoards | PG |

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

*F&S i.MX6UL Linux First Steps*

# Table of Contents

# 1 Introduction

## 1.1 F&S Board Families And CPU Architectures

F&S offers a whole variety of Systems on Module (SOM) and Single Board Computers (SBC). There are different board families that are named NetDCU, PicoMOD, PicoCOM, armStone, QBliss, efus and PicoCore (see Table 1).

| Family | Type | Size |
|--------|------|------|
| NetDCU | Single Board Computer | 80 mm x 100 mm |
| PicoMOD | System on Module | 80 mm x 50 mm |
| PicoCOM | System on Module | 40 mm x 50 mm |
| armStone | Single Board Computer | 100 mm x 72 mm (PicoITX) |
| QBliss | System on Module | 70 mm x 70 mm (Qseven) |
| efus | System on Module | 62 mm x 47 mm |
| PicoCore | System on Module | 40 mm x 35 mm |

*Table 1: F&S Board Families*

Linux is available for all of these platforms. F&S combines releases for platforms with the same CPU – or rather SoC (System on Chip) – as so-called *architecture releases*. All the boards of the same architecture can use the same sources, and the binaries can be used on any board of this architecture. Please note the difference: *board families* are grouped by form factor, *architectures* are grouped by CPU type, i.e. they usually contain boards of different families.

Table 2 shows all the architectures that are currently supported by F&S.

| Architecture | CPU | Platforms |
|--------------|-----|-----------|
| fsvybrid | NXP Vybrid VF6xx | PicoCOMA5, NetDCUA5, armStoneA5, PicoMOD1.2 |
| fsimx6 | NXP i.MX6 | efusA9, QBlissA9, QBlissA9r2, armStoneA9, armStoneA9r2, PicoMODA9, NetDCUA9 |
| fsimx6sx | NXP i.MX6-SoloX | efusA9X, PicoCOMA9X, PicoCoreMX6SX |
| fsimx6ul | NXP i.MX6-UL/ULL | efusA7UL, PicoCOM1.2, PicoCOMA7, PicoCoreMX6UL, PicoCoreMX6UL100 |
| fsimx7ulp | NXP i.MX7ULP | PicoCoreMX7ULP |

*Table 2: F&S Architectures*

**Remark**

In December 2015, the two companies Freescale and NXP merged and both companies are now working under the brand name NXP. The name Freescale will disappear in the future, which is why we only use "NXP" throughout this document now. However some programs still output "Freescale" at some places. We have not touched this output to reflect the situation as it is.

## 1.2   Scope of This Document

This document describes the *fsimx6ul* architecture. That means all F&S boards and modules based on the NXP i.MX6-UltraLite SoC (i.MX6UL and i.MX6ULL). The steps in this document will help you getting to know your board and do some basic operations in Linux, so that you can try out all the periphery and do some first tests and comparisons.

The additional document `LinuxOnFSBoards_eng.pdf` explains the more generic ideas and concepts of Linux on F&S boards and modules. So after having become acquainted with the board, you should continue reading that Linux document to get a more in-depth knowledge of the board and software.

# 2 Setting up the Board

In this chapter we will show how to connect the board to the PC. For a first test of the board functions, we only need a serial connection between PC and board. So as a first step, we will introduce all the boards and Starterkits of the *fsimx6* architecture and show the location of all connectors, especially the debug port.

## 2.1 Locating the Connectors on the Starterkit

### 2.1.1 efusA7UL

The Starterkit includes all components that are required for an initial setup. This includes:

- Cables (Ethernet, serial, power, USB, …).
- Software (source, binaries, install scripts, examples).
- Starterkit carrier board that offers connectivity for most interfaces available in efusA7UL.
- efusA7UL module.

For basic operation please make sure that power and Serial A debug port are connected correctly.
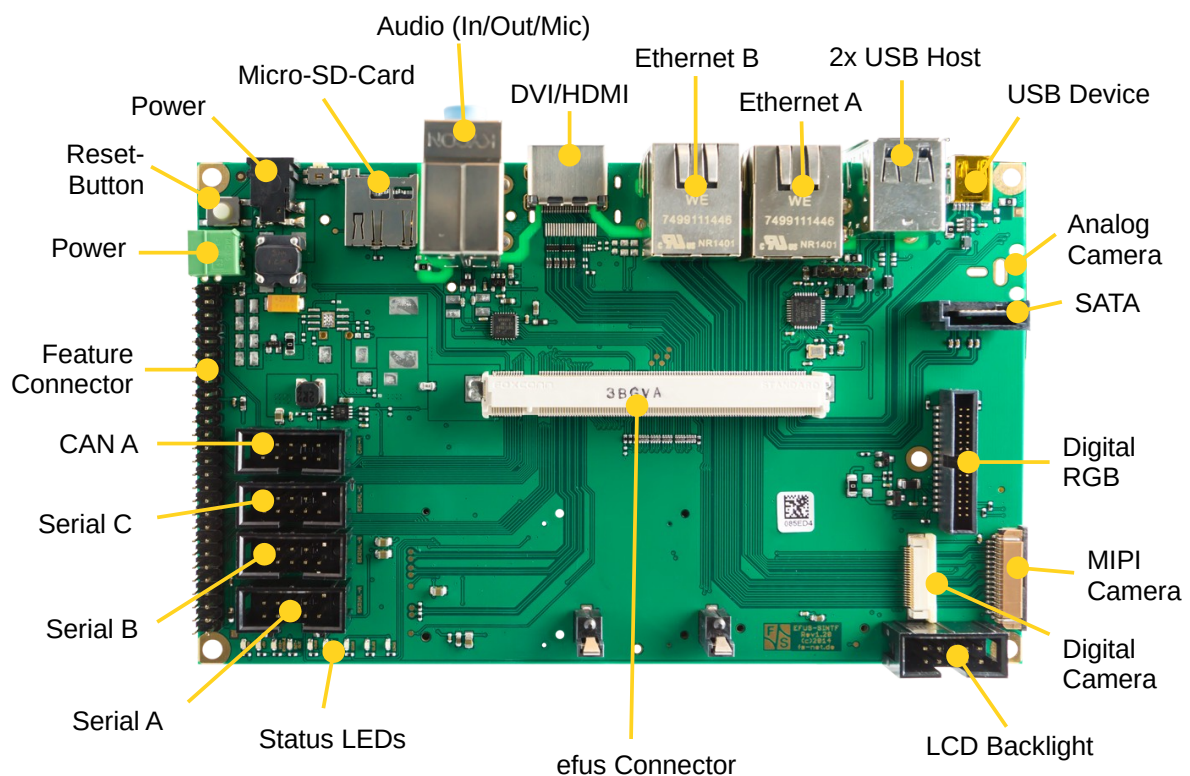


*Figure 1: Top side of efus SKIT carrier board*

*Setting up the Board*

Figures 1 shows the connectors available on the top side of the efus SKIT carrier board. This carrier board is also available for other efus modules like the efusA9 and efusA9X. Not all features are available on all efus modules. For example the efusA7UL has no support for DVI/HDMI, SATA, any kind of camera, MiniPCIe and for additional status LEDs. The Micro-SD-Card slot can only be used if no on-board WLAN is equipped and the normal-sized SD-Card slot can only be used if no eMMC is equipped.
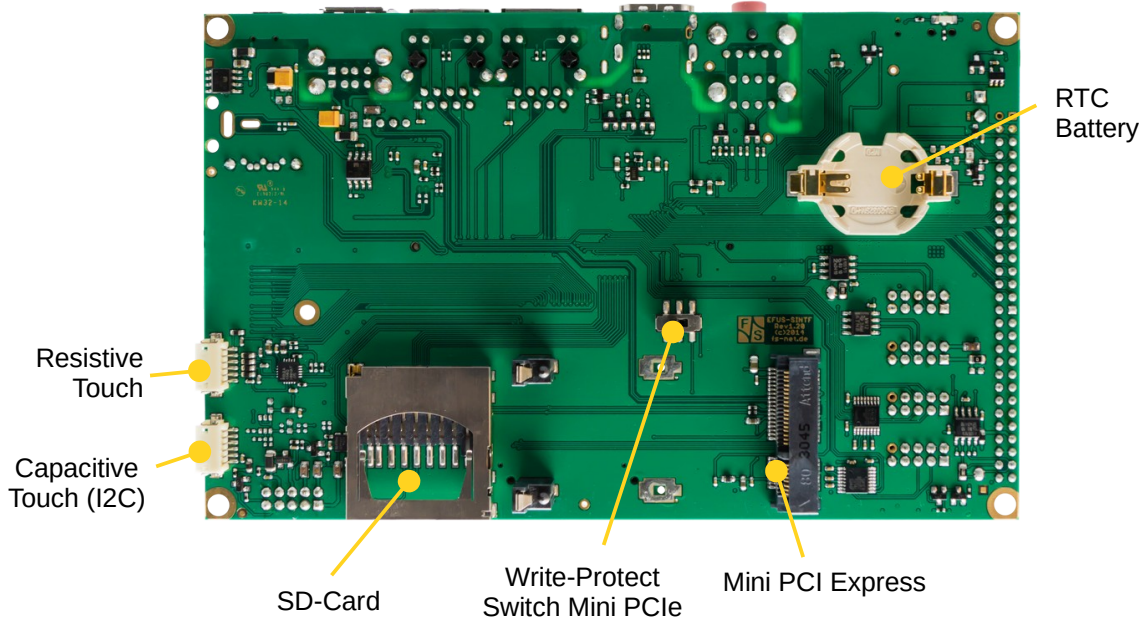


*Figure 2: Bottom side of efus SKIT carrier board*

The connections available from the bottom side of the efus SKIT can be seen in Figure 2. The efusA7UL module itself is shown in Figure 3.
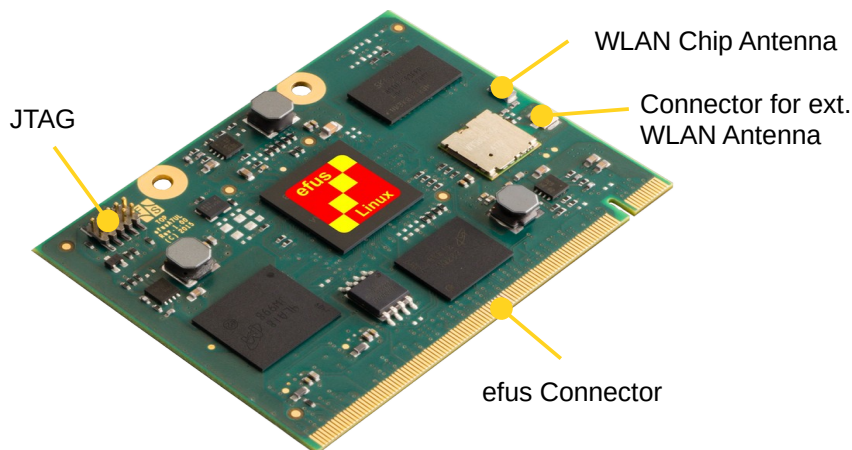


*Figure 3: Top side of efusA7UL module*

More detailed information about connectors available on the SKIT and the efusA7UL can be found in the "Hardware documentation for efus Startinterface" and the "efusA7UL Hardware Documentation".

*F&S i.MX6UL Linux First Steps*

## 2.1.2   PicoCOM1.2

The Starterkit includes all components that are required for an initial setup. This includes:

- Cables (Ethernet, serial, power, USB, …).
- Software (source, binaries, install scripts, examples).
- Starterkit carrier board that offers connectivity for most PicoCOM1.2 interfaces.
- PicoCOM1.2 module.

Figure 4 shows the Starterkit baseboard with the PicoCOM1.2 module mounted. For basic operation please make sure that power and UART_C debug port are connected correctly.
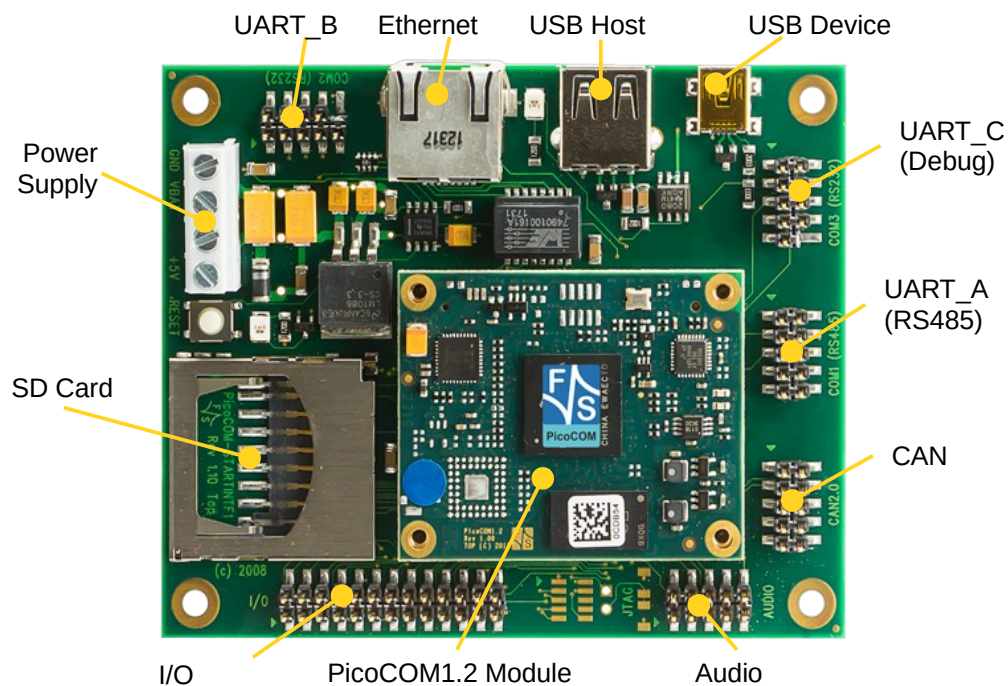


*Figure 4: PicoCOM1 Starterkit with PicoCOM1.2 module*

Figure 5 shows the PicoCOM1.2 module itself.
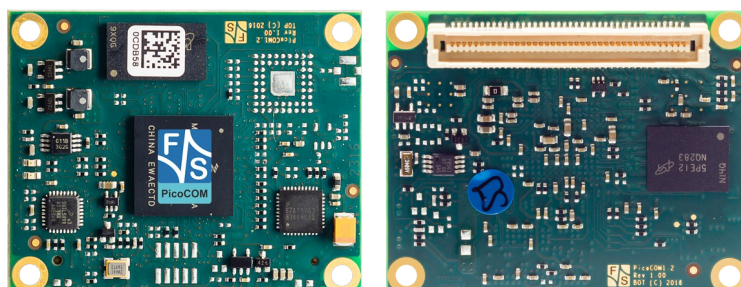


*Figure 5: PicoCOM1.2 top and bottom view*

### 2.1.3   PicoCOMA7

The Starterkit includes all components that are required for an initial setup. This includes:

- Cables (Ethernet, serial, power, USB, …).
- Software (source, binaries, install scripts, examples).
- Starterkit carrier board that offers connectivity for most PicoCOMA7 interfaces.
- PicoCOMA7 module.

Figure 6 shows the Starterkit baseboard with the PicoCOMA7 module mounted. For basic operation please make sure that power and UART_A debug port are connected correctly.
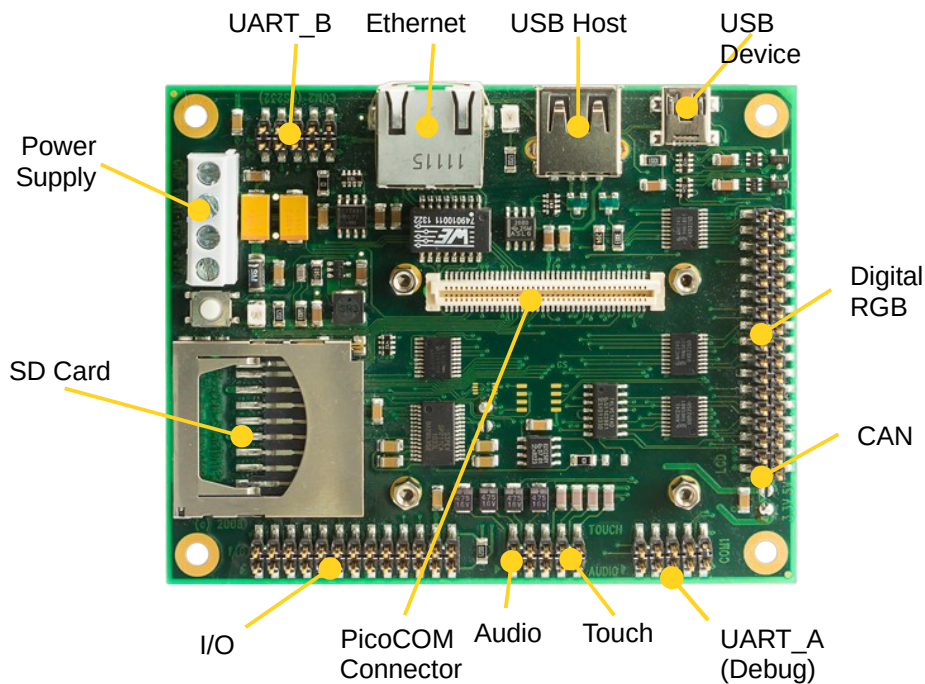


*Figure 6: PicoCOM STARTINTF2 Starterkit*

Figure 7 shows the PicoCOMA7 module itself.



*Figure 7: PicoCOMA7 top and bottom view*

### 2.1.4   **PicoCoreMX6UL**

The Starterkit includes all components that are required for an initial setup. This includes:

- Cables (Ethernet, serial, power, USB, …).

- Software (source, binaries, install scripts, examples).

- Starterkit carrier board that offers connectivity for most PicoCoreMX6UL interfaces.

- PicoCoreMX6UL module.

Figure 8 shows the Starterkit baseboard. For basic operation please make sure that power and UART_A debug port are connected correctly.



*Figure 8: PicoCoreMX6UL Starterkit baseboard (PicoCoreBBRGB)*

Figure 9 shows the PicoCoreMX6UL module itself.
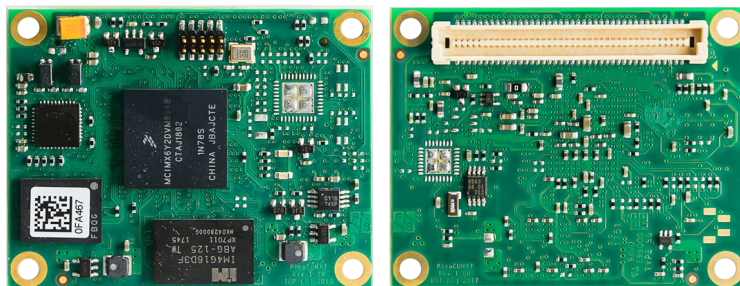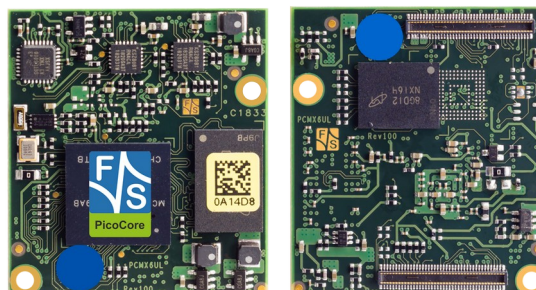


*Figure 9: PicoCoreMX6UL top and bottom view*

## 2.1.5  PicoCoreMX6UL100

The Starterkit includes all components that are required for an initial setup. This includes:

- Cables (Ethernet, serial, power, USB, …).

- Software (source, binaries, install scripts, examples).

- Starterkit carrier board that offers connectivity for most PicoCoreMX6UL100 interfaces.

- PicoCoreMX6UL100 module.

Figure 10 shows the Starterkit baseboard. For basic operation please make sure that power and UART_A debug port are connected correctly.
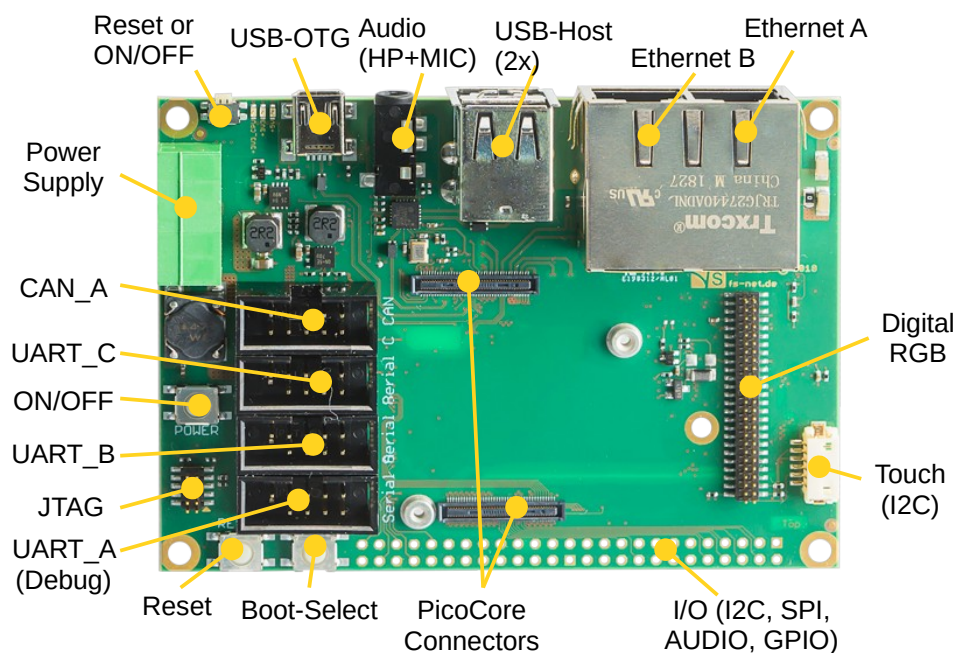


*Figure 10: PicoCoreMX6UL100 Starterkit baseboard top (PicoCoreBBDSI)*

RGB/LVDS

mPCIe

MIPI-CSI (Only available on PicoCoreMX8MP)

*Figure 11: PicoCoreMX6UL100 Starterkit baseboard bot (PicoCoreBBDSI)*

Figure 12 shows the PicoCoreMX6UL100 module itself.
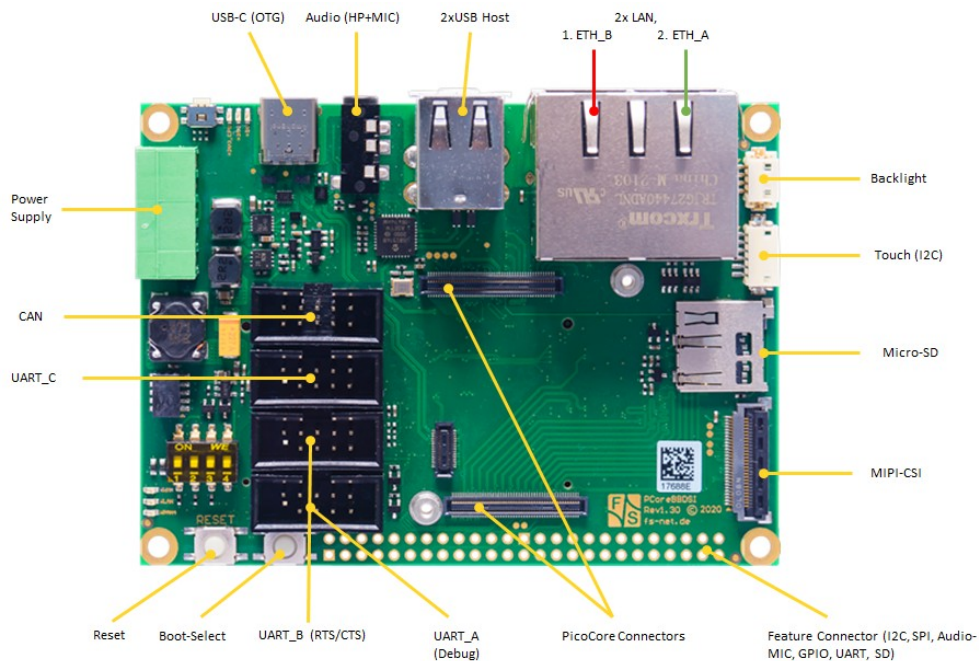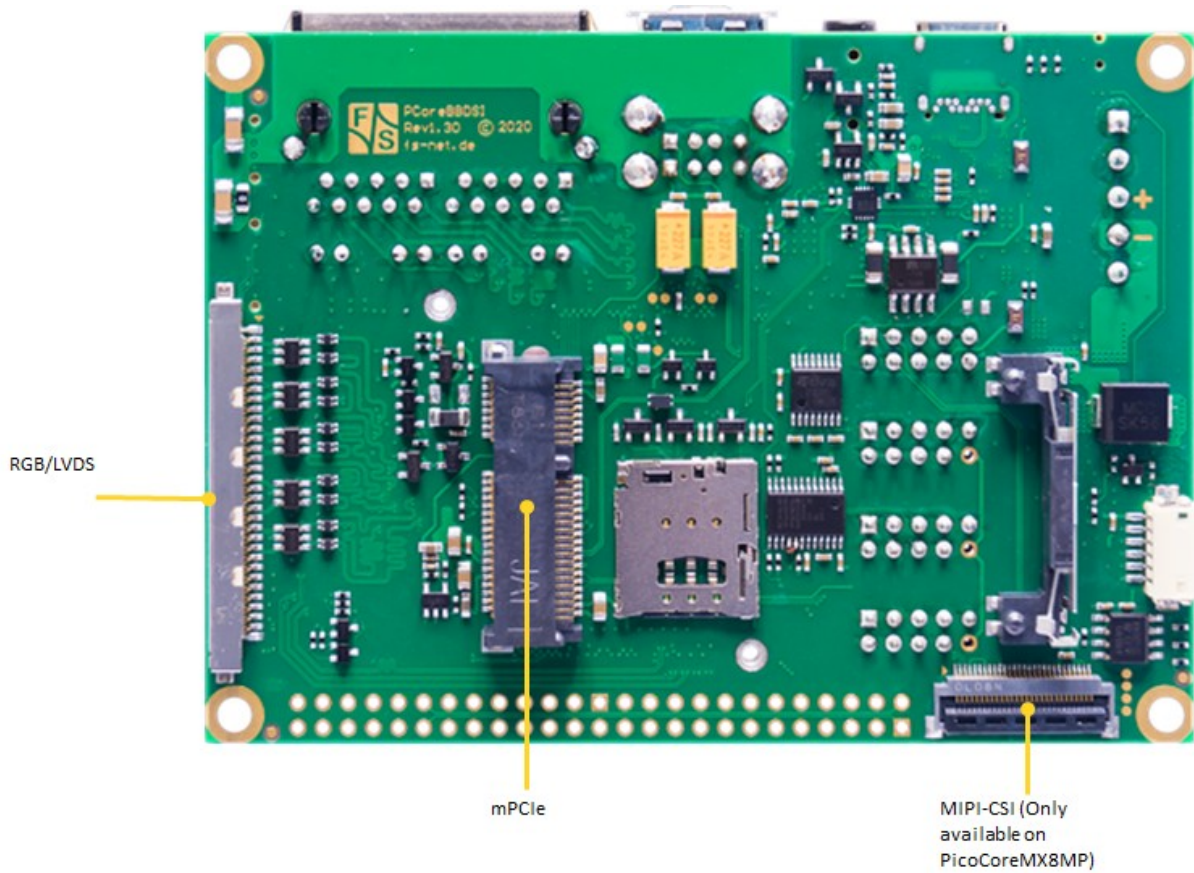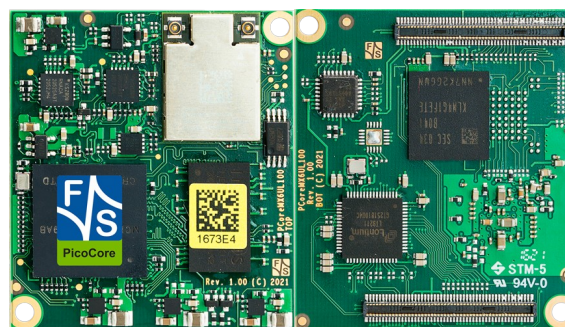


*Figure 12: PicoCoreMX6UL100 top and bottom view*

## 2.2   Serial Connection

To work with the board, you need a serial connection with your PC. Use the provided Null-Modem cable and connect the debug port of the board (or Starterkit baseboard) with the serial port of a PC. Please refer to chapter 2.1 for the location of the COM ports. A serial port is mandatory on your PC, because we control the whole board via the serial port. If your PC does not provide a serial port, you have to either use a USB-to-serial adapter or you need to install a PCIe extension card with a serial port.
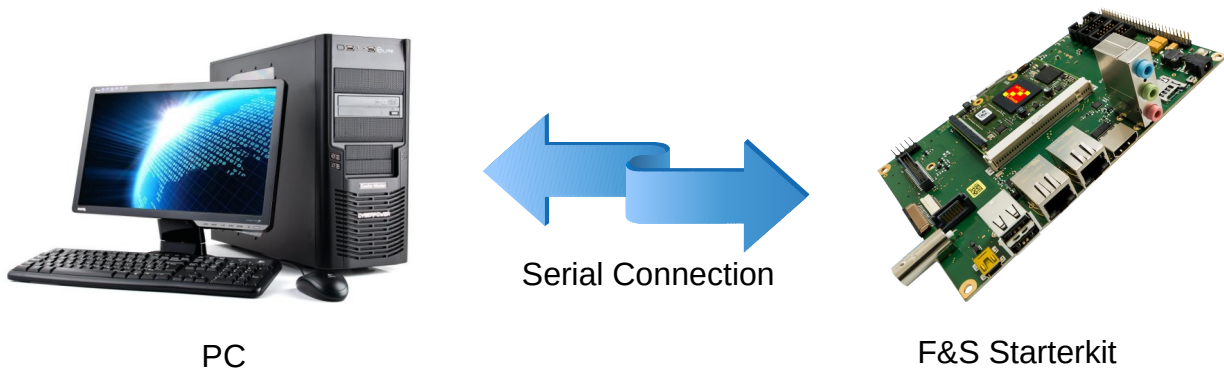


Serial Connection

PC                                                            F&S Starterkit

*Figure 13: Serial connection from board to PC*

For a first test, a Linux PC is not necessarily required. You can also use a Windows PC. But later for development, you definitely need a Linux PC, either native or as a Virtual Machine. With a Virtual Machine, you compile your software in Linux but you can still have the serial connection done in Windows and use tools from Windows. This uses the best of both worlds.

On your PC, start a terminal program and open a serial connection to the board. Use 115200 baud, 1 start, 1 stop bit, no flow control. We recommend a terminal program that supports a 1:1 binary download and also supports ANSI Escape Sequences for colour and text highlighting. Examples are:

- `TeraTerm` (Windows)
- `PuTTY` (Windows/Linux, does not support 1:1 download)
- `minicom` (Linux, does not support 1:1 download, but not needed in Linux)

F&S also provides a small terminal program for Windows called `DCUTerm`. You can find `DCUTerm` in the Tools-Section of the Download Area (in *My F&S*). However `DCUTerm` does not support ANSI Escape Sequences, which means the output of a Linux command like `ls` is nearly unreadable. Instead of different colours for different file types, you will see a mixture of file names and verbatim escape sequences. Also accessing the command history with the up and down arrow keys is not possible in `DCUTerm`. So `DCUTerm` is not suited very well for Linux. However it supports a 1:1 binary download. So `DCUTerm` is actually a good companion for `PuTTY`. Use `DCUTerm` for serial downloads and `PuTTY` for everything else.

## 2.3   Start Board

Connect a power supply to the board. Please refer to chapter 2.1 for the location of the power supply pins. You need to supply +5V.

Now switch on the power supply. Quite immediately the terminal program should show boot messages from the booting Linux system. This will go on for a few seconds and then a login prompt should appear.

```
Welcome to F+S i.MX6UL
fsimx6ul login:
```

Enter `root` to log in. In the default configuration, no password is required.

If everything went well, you can skip the next chapter and proceed with entering Linux commands.

# 3 Software Installation

When you get a Starterkit from F&S, the Linux system is usually pre-installed and boots to the Linux login prompt right away. In this case you can skip this chapter. But if you are switching over from a different operating system, if you are upgrading from a previous release, or if your board is empty for some other reason, the following sections describe how to install some standard software on your platform.

Here we will only show a very simple automatic installation procedure using an SD card or USB stick and some pre-compiled images from the F&S website. This is the easiest way to get to a running system. Of course, there are other ways to install software, for example via network (TFTP). However, this would go beyond the scope of this First Steps document.

## 3.1 Download Images From F&S Website

To download any software, go to the F&S main website

<p align="center">https://www.fs-net.de</p>
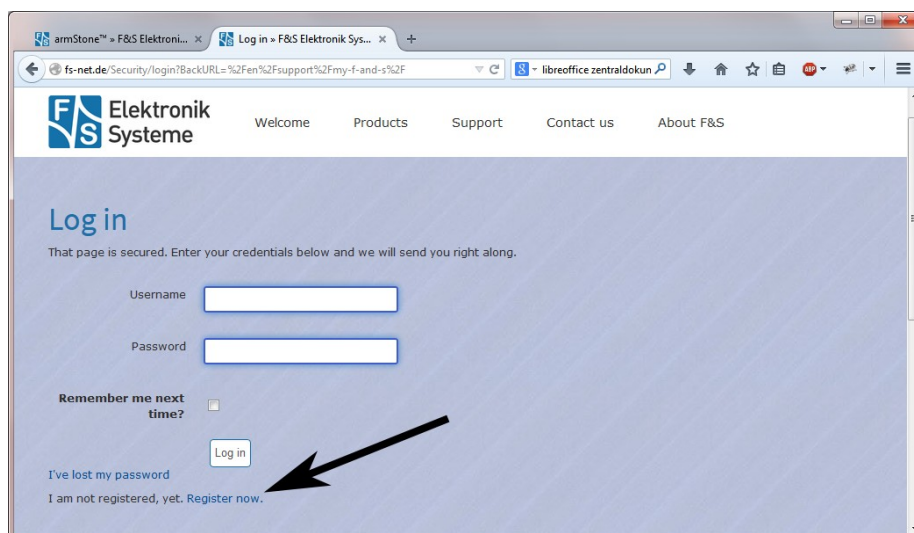


*Figure 14: Register with F&S website*

To download any software, you first have to register with the website. Click on *Login* right at the top of the window and on the text "I am not registered, yet. *Register now*" (Figure 14).

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, like the Support Forum and downloading software.

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. In the top sections there are private downloads for you or your company (may be empty) and in the bottom section you will find generic downloads for all registered customers.

*Figure 15: Unlock software with the serial number*

To get access to the software of a specific board in the bottom section, you have to enter the serial number of one of these boards (see F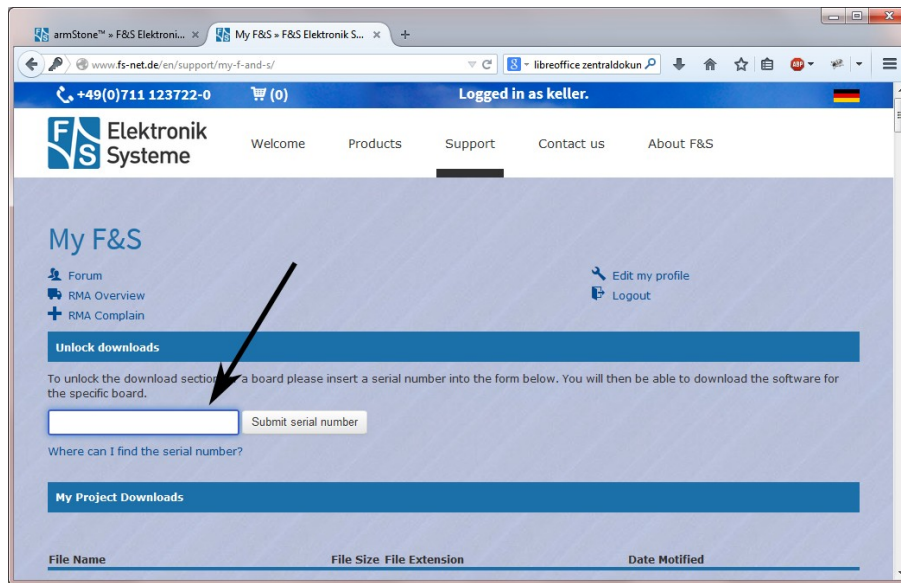igure 15). Click on "Where can I find the serial number" to get pictures of examples where to find this number on your product. Enter the number in the white field and press *Submit serial number*. This enables the software section for this board type for you. You will find Linux, Windows CE, and all other software and tools available for this platform like `DCUTerm` or `NetDCUUsbLoader`.

First click on the type of your board, e.g. efusA7UL, then on *Linux*. Now you have the choice of *Buildroot* or *Yocto*. For the first steps here, we will use the newest Buildroot release, because this is the software that is also installed on our Starterkits. So click on *Buildroot*. This will bring up a list of all our Buildroot releases. Old releases up to 2018 had `V<x>.<y>` as version identifier, new releases use `B<year>.<month>`. We will abbreviate this as `<v>` from now on. Select the newest version, for example *fsimx6ul-B2019.11*. This will finally show two archives that can be downloaded.

`fsimx6ul-<v>.tar.bz2`............This is the main release itself containing all sources, the binary images, the documentation and the toolchain.

`sdcard-fsimx6ul-<v>.tar.bz2`     Files that can be stored on an SD Card or USB stick to allow for easy installation.

For board specific releases the name consists of the name of the board instead of the architecture. (e.g. `picocoremx6ul` instead of `fsimx6ul`)

For now we will only need the SD card archive. This archive contains some pre-compiled images of bootloaders, Linux kernel, device trees and root filesystem. It is compressed with bzip2. To see the files, you first have to unpack the archive, for example in Linux with

```
tar xvf sdcard-fsimx6ul-<v>.tar.bz2
```

This will create a directory `sdcard` for NAND and `sdcard_mmc` for eMMC that contains all necessary files. Choose the right directory for your device or else the system will not boot. Now copy these files to an SD card or USB stick. We will call this the *installation media*. It has to be formatted with the FAT filesystem. Do not create any subdirectories, the files have to reside directly in the top directory of the media.

## 3.2  Enter NBoot

NBoot is a small first-level bootloader that is running before the main bootloader. It is the same for Linux and Windows CE and always remains on the board, even if the whole flash memory is erased. As long as NBoot is available on the board, it is always possible to bring up the whole system again without the need for any special hardware or software tools.

Normally, NBoot is completely invisible and just loads and starts the main bootloader. Then the first output on the serial port is from the main bootloader itself. But now we will use NBoot to erase any old content and download the correct U-Boot image. This means we have to stop the boot process right at the beginning and enter NBoot.

This requires the serial setup as explained in Chapter 2.2. And we need a 1:1 download option, which means PuTTY is not suited for this. So use one of the other terminal programs. Open the serial connection, then press and hold key *s* (lower case S). While holding this key, switch on power of the board (or press the reset button). This should bring you into NBoot. You should see something like this (output is taken from efusA7UL, the real messages may vary slightly depending on the software version):

```
F&S Nand Loader VN38 built Jul 15 2019 11:20:38

efusA7UL Rev. 1.20

...

Please select action

'd' -> Serial download of bootloader
'E' -> Erase flash
'B' -> Show bad blocks
Use NetDCUUsbLoader for USB download
```

## 3.3  Erase Flash

To erase any old content, simply press *E* (upper-case e). This removes everything that was on the board before. Don't be afraid, this won't erase NBoot itself.

For the eMMC only version of NBoot the erase command currently isn't supported.

## 3.4   Create Partitions

> **Note**
>
> The following part of the chapter is only viable for devices with eMMC only and NBoot version 43 or higher. Functionality was added to store the U-Boot together with Kernel and Device Tree in a FAT partition, which is different from earlier implementations.

For eMMC only devices a FAT partition and MBR must be present at the `User` partition to write U-Boot. To initialize the partitions press *w* (lower-case W). Automatically a FAT partition with the size 40 MB and an ext4 partition with the size of 800 MB will be created. You can define more partitions with partition type and size (e.g. 0x83 for ext4 and 40960 for 40 kB). The characters typed in will not show on the terminal. If you don't need any further partitions, you can simply press Enter without typing in values for type and size. When all partitions are defined, a prompt is shown to validate the writing of the partitions. This will result in overwriting any current partition table.

```
Create MBR

Creating boot partition 0 (FAT16, 40MiB)

creating partition  0 with type 0x06 and size 40 MiB

Creating rootfs partition 1 (Linux, 800MiB)

creating partition  1 with type 0x83 and size 800 MiB

enter additional data partition 2

filesystem type:filesystem size in MiB:

WARNING: writing the MBR will delete all data!


Enter 'y' to proceed, any other key to exit

writing MBR

writing FAT
```

## 3.5   Download and Save U-Boot

Unfortunately NBoot for fsimx6ul is not capable of accessing files on an SD card or USB stick. Which means you have to download the U-Boot image via the serial line. This is the part where the 1:1 download comes in. While still in NBoot, press *d* (lower-case D). This will show some message similar to this:

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

Now the steps are different, depending on the terminal program that you use.

- In `DCUTerm`, go to File → *Transmit Binary File...* and open the file `ubotmx6u.nb0` from the `sdcard` directory. This starts the serial download.

- In `TeraTerm`, go to *File → Send File...* and activate the *Binary* option checkbox. This is important! Then open the file `ubotmx6u.nb0` from the `sdcard` directory. This starts the serial download.

- In Linux, you can copy the U-Boot image `ubotmx6u.nb0` from the `sdcard` subdirectoy directly to the serial port device. This is possible even if the terminal program does not support a 1:1 download function. For example if `/dev/ttyS0` is your serial port, just use a separate shell and enter:

```
dd if=ubotmx6.nb0 of=/dev/ttyS0.
```

Just note that you must not enter any characters in the terminal program while download is in progress. The character would also be sent to the serial port and would be inserted at an arbitrary position in the sequence of bytes. This would shift the remaining file content and would result in a damaged and unusable download.

> **Note**
>
> After pressing *d*, you have exactly 60 seconds to start the download. Then the download command times out and the menu is shown again. This means if you start to send the file too late, every byte of the file is interpreted as an own NBoot command, which is definitely not what you want.

During download, progress is shown by an increasing number of dots and the number of transmitted bytes from time to time.

```
.............................................................. 65536 Bytes
.............................................................. 131072 Bytes
.............................................................. 196608 Bytes
.............................................................. 262144 Bytes
.............................................................. 327680 Bytes
.............................................................. 393216 Bytes
.............................................................. 458752 Bytes
.............................................................. 524288 Bytes
Success, checksum: 0xaa7d

>>> U-Boot image loaded (524288 bytes) <<<

Please select action
'f' -> Save image to flash
'x' -> Execute image
'd' -> Serial download of bootloader
'E' -> Erase flash
'B' -> Show bad blocks

Use NetDCUUsbLoader for USB download
```

When download is complete, you see a checksum (which may differ in your case) and the menu again, which now has additional entries. Save U-Boot by pressing *f* (lower case F). This should show

```
Saving U-Boot...Success
```

For devices with only eMMC the MBR and partitions have to be present. If they are not, then you need to initialize them with *w* (lower case W) first, before saving the U-Boot. (Refer to Chapter 3.4)

Also the NBoot MSD (Mass Storage Device) can to be activated by pressing *Z* (upper case Z). When the USB port is connected to the Host-PC, the partitions will be shown as USB storage devices and the U-Boot can simply be put in the right partition. The naming of the U-Boot file is critical, so be sure to use the same file name. (uboot.nb0)

## 3.6 Install Kernel, Device Tree And Root Filesystem

Now insert the installation device into the board or Starterkit baseboard. There is one installation device for NAND and one for eMMC. The remaining installation is fully automatic and is done by U-Boot. As U-Boot is still available in RAM from the previous step, you can directly start it by pressing *x* (lower-case X). This will show something like this:

```
U-Boot 2018.03 (Nov 11 2019 - 11:16:22 +0100) for F&S

CPU:   Freescale i.MX6ULL rev1.0, 900 MHz (running at 396 MHz)
CPU:   Commercial temperature grade (0C to 95C)
Reset: POR
Board: efusA7UL Rev 1.20 (2x LAN, WLAN, eMMC, 1x DRAM)
I2C:   ready
DRAM:  512 MiB
NAND:  256 MiB
MMC:   FSL_SDHC: 0
In:    serial
Out:   serial
Err:   serial
Net:   FEC [PRIME]
Hit any key to stop autoboot:  3
```

The number in the last line will count down to zero, then the installation procedure will start. The files are loaded from the installation media and are stored in NAND flash on the board. When the installation is over, you will see the following line

```
Installation complete
Please set/verify ethernet address(es) now and call saveenv
```

## 3.7 Set MAC Address

When we erased the flash content in Chapter 3.3, we also erased the U-Boot environment including the MAC address for the Ethernet chip. We have to set it again now and save it permanently.

The MAC address is a unique identifier for a network device. Each network device has its own address that should be unique across the whole world. So each network port on each board needs a unique MAC address.

A MAC address consists of twelve hexadecimal digits (0 to 9 and A to F), that are often grouped in pairs and separated by colons. The first six digits for F&S boards are always the same: 00:05:51, which is the official MAC address code for the F&S company. The remaining six digits can be found on the bar-code sticker directly on your board (see Figure 16).

The full MAC address for this example would be 00:05:51:07:93:4B. If your board supports two ethernet ports, you need two MAC addresses. The second one is the first one plus 1, i.e. 00:05:51:07:93:4C.
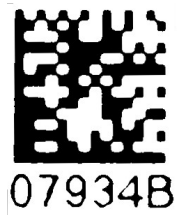
*Figure 16: Bar-code sticker*

The following two commands will set the MAC addresses and stores the current environment (including the newly set MAC addresses) in NAND flash. Of course you have to replace `xx:yy:zz` with the six hex digits from the bar-code sticker on your board (and xx:yy:vv with the six hex digits plus 1).

```
setenv ethaddr 00:05:51:xx:yy:zz
setenv eth1addr 00:05:51:xx:yy:vv
saveenv
```

**Warning**

If you do not set this unique address, a default address is used that is the same for all boards of this type. This will definitely lead to problems in real networking scenarios.

## 3.8   Restart Board

Installation is complete. To check if everything was done correctly, restart the board. You can either enter U-Boot command, ...

```
reset
```

… or press the reset button, or simply switch the power off and on again. Like in chapter 2.3, the terminal program should show boot messages from the booting Linux system. This will go on for a few seconds and then a login prompt should appear.

```
Welcome to F+S i.MX6UL
fsimx6ul login:
```

Enter `root` to log in. In the default configuration, no password is required.

If this is still not working, you should repeat the steps from the whole chapter.

# 4    Next Steps

This document only showed a very basic usage of the board and the Linux system. The next logical step is the generic Linux documentation `LinuxOnFSBoards_eng.pdf`. It will show you the ideas and concepts behind the F&S Linux environment and how you can work efficiently with these boards.

## 4.1    F&S Workshops

F&S also offers several workshops. Especially if you are new to working with embedded boards or even new to Linux, we recommend visiting the workshop "Linux on F&S Modules". Working with an embedded system is quite different to working with a desktop Linux. This workshop will show you a basic introduction to Linux, how to use NBoot, U-Boot and Linux on an F&S board, how to compile the system software, how to download files to the board, and how to write your own programs. The workshop lasts four hours and takes place in Stuttgart at the F&S company building. It may save you many hours of reading, trying, and even frustration.

Additional workshops are available for working with Buildroot, Asymmetric Multiprocessing, Secure Boot, Working with GIT. Please look at our website for any additional offerings.

## 4.2    Further Information

Many additional resources of information are available on the F&S website.

| Document | Description |
| --- | --- |
| `AdvicesForLinuxOnPC.pdf` | Explains how to install server software  and tools on a Linux development PC that is used with F&S Linux boards. |
| `*-GPIO-Reference-Card_eng.pdf` | Lists all pins of the board and which GPIO number needs to be used in Linux |
| `*_Hardware_eng` | Hardware documentation; there are separate documents for each board and also for the Starterkit baseboards. F&S also offers Eagle layout files for some of our Starterkits. |

*Table 3: Important documents, available on the F&S website*

We do not include all these documents in the release to make sure that you always get the newest version when you start. The following sections give direct links to important places like documentation and add-ons.

*Next Steps*

A good source for information is also our internet forum. If you have any questions or specific problems, please feel free to go to: https://forum.fs-net.de/.

### 4.2.1 Resources for efusA7UL

Hardware documentation for efusA7UL module itself, Starterkit baseboard, including schematics and Eagle-Layout files:

https://www.fs-net.de/en/products/efus/efusa7ul-with-nxp-imx6-ultralite/#panel-7

Available accessories, adapters and extensions:

https://www.fs-net.de/en/products/efus/efusa7ul-with-nxp-imx6-ultralite/#panel-4

### 4.2.2 Resources for PIcoCOM1.2

Hardware documentation for PicoCOM1.2 module itself, Starterkit baseboard, including schematics:

https://www.fs-net.de/en/products/picocom/picocom1-2-nxp-i-mx-6ultralite-processor/#panelde/en/products/picocom/picocom-7

Available accessories, adapters and extensions:

https://www.fs-net.de/en/products/picocom/picocom1-2-nxp-i-mx-6ultralite-processor/#panel-4

### 4.2.3 Resources for PIcoCOMA7

Hardware documentation for PicoCOMA7 module itself, Starterkit baseboard, including schematics:

https://www.fs-net.de/en/products/picocom/picocoma7/#panel-7

Available accessories, adapters and extensions:

https://www.fs-net.de/en/products/picocom/picocoma7/#panel-4

### 4.2.4 Resources for PicoCoreMX6UL

Hardware documentation for PicoCoreMX6UL module itself, Starterkit baseboard, including schematics and Eagle-Layout files:

https://www.fs-net.de/en/products/picocore/picocoremx6ul-with-nxp-i-mx6-ull-processor/#panel-6

Available accessories, adapters and extensions:

https://www.fs-net.de/en/products/picocore/picocoremx6ul-with-nxp-i-mx6-ull-processor/#panel-4

## 4.2.5   Resources for PicoCoreMX6UL100

Hardware documentation for PicoCoreMX6UL100 module itself, Starterkit baseboard, including schematics and Eagle-Layout files:

https://fs-net.de/en/embedded-modules/computer-on-module-picocore/picocoremx6ul100/#panel-6

Available accessories, adapters and extensions:

https://fs-net.de/en/embedded-modules/computer-on-module-picocore/picocoremx6ul100/#panel-4

# 5    Appendix

## List of Figures

## List of Tables

## Listings

# 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 authorised 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 unauthorised 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 unauthorised use, even if such claim alleges that F&S Elektronik Systeme was negligent regarding the design or manufacture of said product.