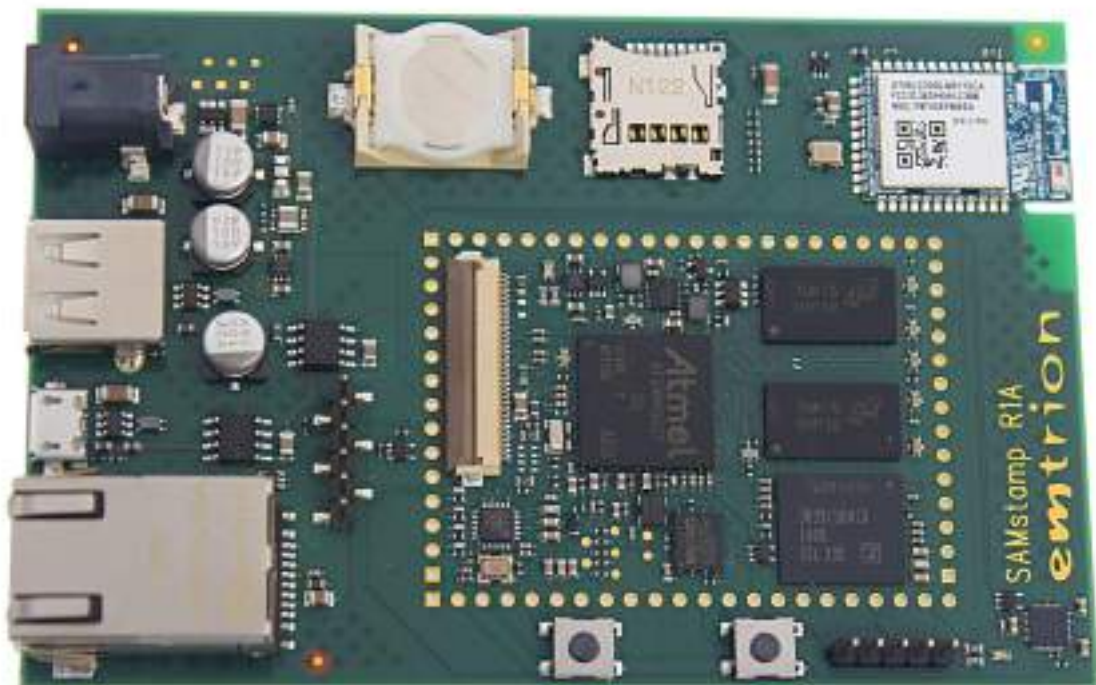


emSTAMP Helium Linux Developer Kit

Software Manual

Rev02 / 13.07.2018



© Copyright 2018 **emtrion GmbH**

All rights reserved. This documentation may not be photocopied or recorded on any electronic media without written approval. The information contained in this documentation is subject to change without prior notice. We assume no liability for erroneous information or its consequences. Trademarks used from other companies refer exclusively to the products of those companies.

Revision: **02 / 13.07.2018**

Rev	Date/Signature	Changes
1	23.04.2018/Ha	Initial release
2	13.07.2018/Ha	Add SW in flash support

1 Table of contents

Revision: 02 / 13.07.2018	2
2 Introduction.....	4
3 Preconfigured image.....	5
3.1 Device Network Setup	6
3.1.1 Ethernet connection	6
3.1.2 Wi-Fi connection.....	6
4 Debian Linux Distribution	6
5 GPIOs available for prototyping	7
6 Install Debian Root File System in eMMC	7
7 U-Boot.....	8
7.1 Basic U-Boot operation	8
7.2 Using U-Boot to change boot device or update parts of the system.....	8
7.2.1 Booting.....	9
7.2.2 Install Uboot in SPI Nor Flash	9

2 Introduction

Welcome to emSTAMP-Helium documentation. This manual will give you a startup software guideline of our developer kit. It will describe how to use the preconfigured image. Further it will give you a quick introduction on how to use your Debian Linux, and to conclude a quick view of the bootloader.

It is assumed that users of emtrion Linux developer kits are already familiar with Linux. General Linux and programming knowledge are out of the scope of this document. emtrion will gladly assist you in acquiring this knowledge. If you are interested in training courses or getting support, please contact the emtrion sales department.

The examples in this manual are demonstrated on specific hardware but if not mentioned otherwise they all work on all supported emtrion devices.

Please refer to the “Hardware Description” of to emSTAMP-Helium available on the emtrion support website (<http://support.emtrion.de>) for more detailed info of the capability of the product.

3 Preconfigured image

You can download the latest preconfigured image from our support pages (<http://support.emtrion.de>) and burn into a micro SD card (4GBytes minimum) using etcher tool (www.etcher.io) for example or using dd command line in Linux.

Etcher tool:



Linux dd command line example with /dev/sdX your micro SD card device:

```
hico@halinux:~/work/SAMA5D2x$ sudo dd if=emtrion.img of=/dev/sdX bs=512
```

Once your SD card ready you can plug it in your emSTAMP-Helium micro SD card socket.

You may plug the FTDI USB/Serial to your computer in order to get the Terminal output.

You may now power on the developer kit. You should see it booting U-Boot and Linux from Flash.

After the developer kit booted you are prompted for login:

- user: **root**
- password: **hico**

3.1 Device Network Setup

3.1.1 Ethernet connection

If your network offers a DHCP server the developer kit can be triggered to use it by using following command in a terminal:

```
#dhcpcd eth0
```

You can check the result with the command **ifconfig**.

3.1.2 Wi-Fi connection

In the rootFS of your device there is a script file provided as example to set-up quickly your Wi-Fi.

The example is for a WPA/WPA2 WiFi access with DHCP server example.

You may want to edit the file **wlan-setup.sh** on target (located in /root) or directly on your computer with the rootFS of your micro SD card mounted, and change to your SSID and your PASSWORD:

Change YOUR_SSID in this line:

```
Wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid '"YOUR_SSID"'
```

Change YOUR_PASSWORD in this line:

```
wpa_cli -p/var/run/wpa_supplicant set_network 0 psk '"YOUR_PASSWORD"'
```

Then run the script:

```
#!/wlan-setup.sh
```

Once finished, you can check the result with the command **"ifconfig wlan0"**.

4 Debian Linux Distribution

Since Debian is a long established Linux distribution you find good and detailed documentation on the internet. A good starting point to learn about Debian is the Debian wiki:

<https://wiki.debian.org>

There also exists an online book about Debian as extensive documentation:

<http://debian-handbook.info/>

Emtrion provide an extended explanation of the Debian BSP in the documentation called:

"Emtrion Developer Kit - Debian based BSP Manual".

5 GPIOs available for prototyping

All the pins around the modules are not connected to a peripheral. This chapter described the free GPIOs that can be used and their possibilities:

Signal	GPIO Linux	PIO Periph A	PIO Periph B	PIO Periph C	PIO Periph D	PIO Periph C	PIO Periph F
PB2	34	NRD/NANDOE	PWMFIO	x	x	x	CLASSD_R1
PB3	35	URXD4	D8	IRQ	PWMEXTRG1	x	CLASSD_R2
PB4	36	UTXD4	D9	FIQ	x	x	CLASSD_R3
PD25	121	SPI1_SPCK	x	FLEXCOM4_IO4	x	x	x
PD26	122	SPI1_MOSI	x	FLEXCOM2_IO0	x	x	x
PD27	123	SPI1_MISO	TCK	FLEXCOM2_IO1	x	x	x
PD31	127	ADTRG	NTRST	IRQ	TCLK3	PCK0	x
PD4	100	TWD1	URXD2	x	GCOL	ISC_D10	NCS0
PD5	101	TWCK1	UTXD2	x	GRX2	ISC_D9	NCS1

Please refer to the HW manual the pin signal location on the module!

6 Install Debian Root File System in eMMC

Once you have a preconfigured image booting Debian Linux on your system, you can install the RootFS into the eMMC by following those steps:

- Copy the RootFS archive in your SD card RootFS partition somewhere in /home
- Boot your emStamp-Helium with the SD card prepared like above
- First we need to format the eMMC:

```
root@SAMStamp:~# mkfs.ext4 /dev/mmcblk0
```

- Mount the eMMC:

```
root@SAMStamp:~# mount /dev/mmcblk0 /media/
```

- Extract the RootFS archive in the eMMC and unmount the RootFS:

```
root@SAMStamp:~# tar -xmf /home/rootfs.tar -C /media/
```

```
root@SAMStamp:~# sync
root@SAMStamp:~# umount /media/
```

Now you have the RootFS install in the eMMC you can proceed to the installation of the bootloader in the SPI Nor Flash (chapter 7.2.2) and the emStamp-Helium will boot without SD card.

7 U-Boot

The basic task of U-Boot is to load the operating system from bulk memory into RAM and then start the kernel. You can also use it to initiate an update of the kernel, the root file system and of U-Boot itself. Furthermore you can configure directly from the medium the operating system is to be booted from, for example MMC, NFS or a serial terminal.

7.1 Basic U-Boot operation

To work with U-Boot, first use a terminal program to connect to the serial line of the board. As soon as the U-Boot prompt appears in the terminal, U-Boot is ready to receive commands. To interrupt the automatic boot mechanism press a keyboard key while using the terminal. The general U-Boot documentation can be found here: <http://www.denx.de/wiki/U-Boot/Documentation>

U-Boot has a set of environment variables which are used to store information needed for booting the operating system. Variables can contain information such as IP addresses, but they can also contain a whole script of actions to perform sequentially. The following commands explain the basic handling of environment variables:

U-Boot command	Explanation
printenv [variable]	This shows the value of the specified variable. If no variable is specified, the whole environment is shown.
setenv [variable] [value]	Set a variable to a specific value. If no value is specified, the variable gets deleted.
saveenv	Make your changes permanent, so they remain after power off or reboot.

7.2 Using U-Boot to change boot device or update parts of the system

This chapter describes how U-Boot has to be setup for updating and booting. The exact variable assignment is board specific. The variable serverip has to be set with the IP address of the virtual machine. You can get the IP address by entering **sudo ifconfig** in the terminal of your virtual machine.

Take the IP address of the corresponding network adapter and assign it to the variable serverip in the U-Boot console. The format of [IP address] is dot decimal notation.

```
U-Boot # setenv serverip [IP address]
```


7.2.1 Booting

The default boot device in U-Boot is determined by the variable "bootcmd". If you want to set up one of the following boot options as a default you have to set "bootcmd" to the command mentioned below.

Boot from on-board flash

This is the default boot option configured when you receive the developer kit from emtrion. To start it manually simply use this command:

```
U-Boot # boot
```

Boot via network using a NFS share

Make sure you exported a directory via NFS in your virtual machine. Of course the directory must contain an extracted Root File System of the Debian BSP for the board. Updated Root File System tar balls can be downloaded from our support pages (<http://support.emtrion.de>):

```
U-Boot # setenv serverip [ip-address of virtual machine]
U-Boot # setenv nfsroot [path to the root file system]
U-Boot # setenv ip-method [dhcp or static]
U-Boot # setenv ipaddr [ip-address for device, only needed for static ip]
U-Boot # setenv netmask [netmask for device, only needed for static ip]
U-Boot # saveenv
U-Boot # run net_boot
```

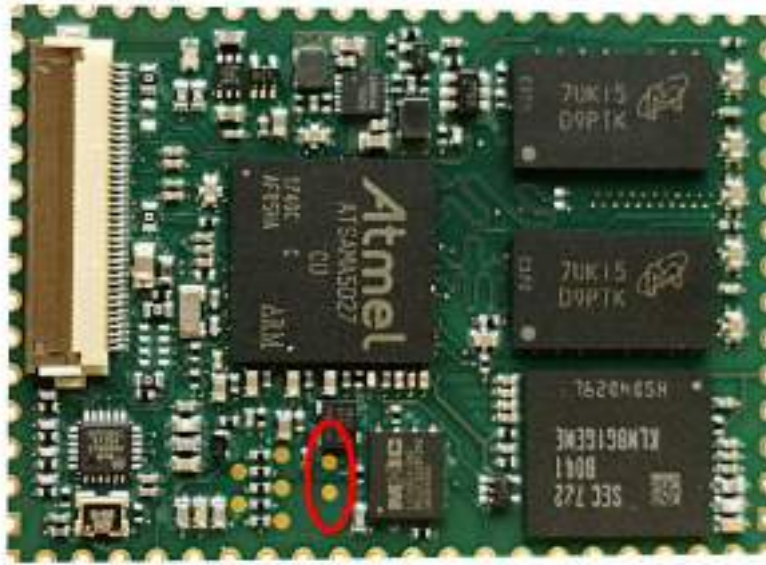
Now the board should boot via network using the NFS share in the virtual machine.

7.2.2 Install Uboot in SPI Nor Flash

First of all you need to install the Microchip SAM-BA tools on your computer. This is a free and open set of tools provided by Microchip for programming their SAM SoC family.

(<https://www.microchip.com/developmenttools/ProductDetails/Atmel%20SAM-BA%20In-system%20Programmer>)

In order to boot the SAM-BA monitor on the emStamp-helium, you need to make a short cut between the two test points (encircled in red below) and reset the board with the USB device cable attached to your PC.



Once booted a serial interface will show up on your computer and you will be able to perform the following command in this example script:

```
#!/bin/bash

SAMBA="sudo /home/hico/work/SAMA5D2x/sam-ba_3.2.1/"
UBOOT_DIR=$(pwd)

# ERASE THE SPI NOR FLASH
$SAMBA/sam-ba -p serial -d sama5d2 -a serialflash:0:1:0:10 -c erase

# WRITE THE BOOT REGISTER, EXTERN MEMORY BOOT
$SAMBA/sam-ba -p serial -d sama5d2 -a bootconfig -c
writecfg:bscr:valid,bureg0
$SAMBA/sam-ba -p serial -d sama5d2 -a bootconfig -c
writecfg:bureg0:ext_mem_boot,sdmmc1_disabled,sdmmc0_disabled,nfc_disabled,s
pil_disabled,spi0_ioaset1,qspi1_disabled,qspi0_disabled

# FLASH UBOOT SPL IN SPI NOR
$SAMBA/sam-ba -p serial -d sama5d2 -a serialflash:0:1:0:10 -c
writeboot:${UBOOT_DIR}/u-boot-spl.bin

# FLASH UBOOT IN SPI NOR
$SAMBA/sam-ba -p serial -d sama5d2 -a serialflash:0:1:0:10 -c
write:${UBOOT_DIR}/u-boot.img:0x20000
```