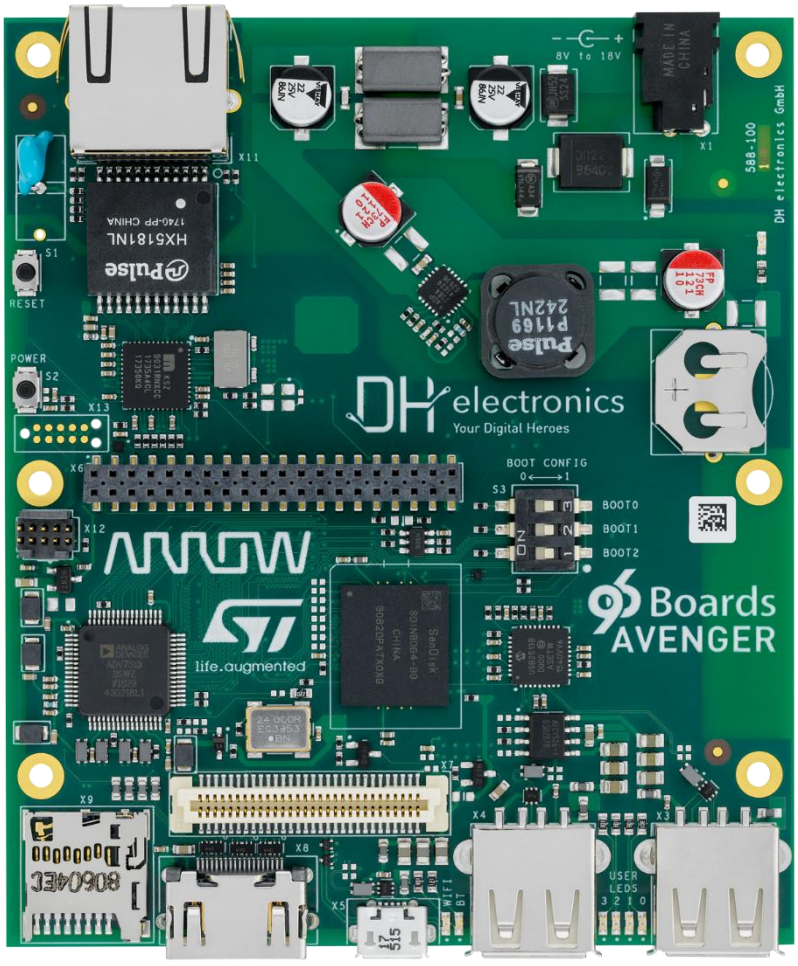
96Boards Avenger96 - Getting Started



DH electronics GmbH ▪ Am Anger 8 ▪ 83346 Bergen ▪ Germany

YOUR DIGITAL HEROES.

History

Revision	Date	Description / Change	Name
R01	13.08.2019	First release based on "DOC_Getting-Started-588-100_R06_2019-04-12".	MH
R02	08.10.2019	Added Settings to start glmark2 GPU Demo manually (2.5.4)	MH
R03	29.05.2020	Updated Link in Chapter 2.1 Building your own image	MH
R04	19.09.2023	Add chapter "Serial console connection" Updated "Creating your own image" to KAS project Mark pre-installed image as outdated Add some new software information's and update chapter "Set Up Guide" Add "OE / Yocto overview", kas-dhsom and "Image scope"	AG

Summary

1 Getting Started with the 96Boards Avenger96

➔ See chapter 1.2 Set Up Guide for "How to start Linux"

1.1 Hardware Overview

This chapter describes all features and key components of the Avenger96 board.

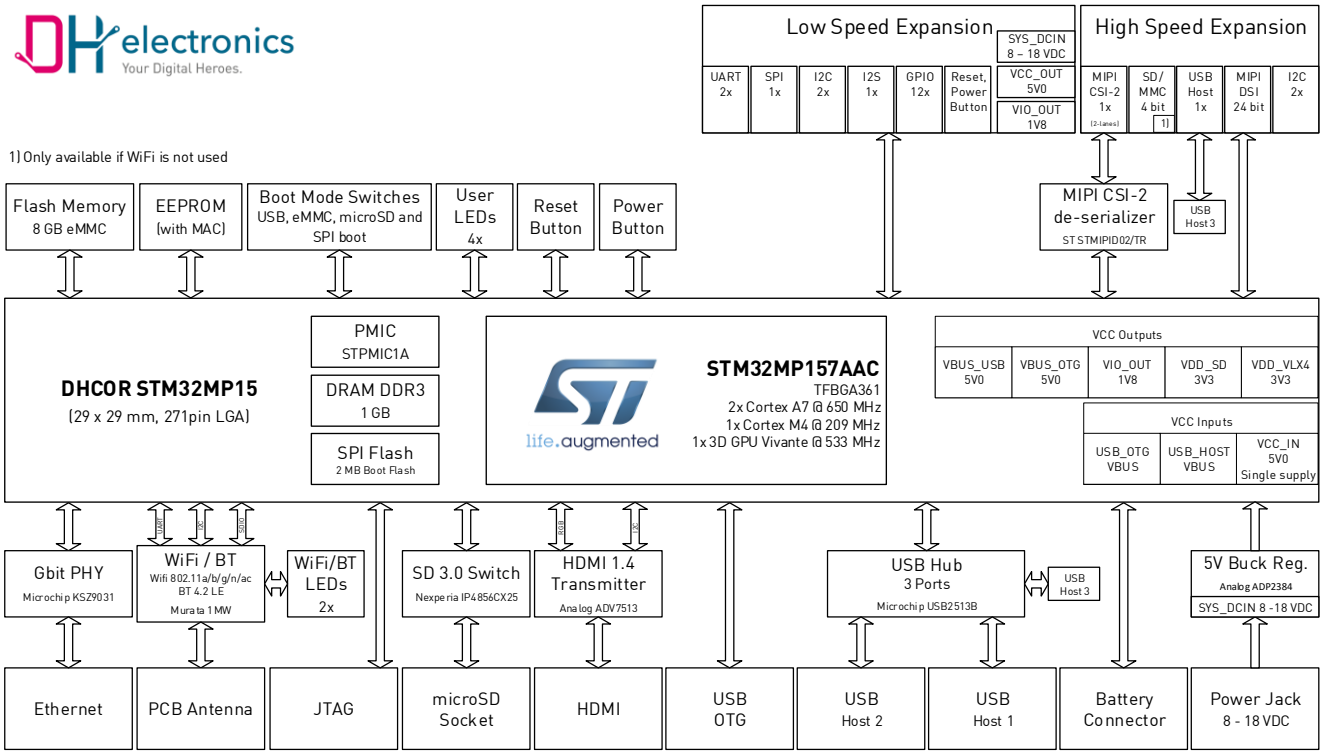
Where do you get the schematics? https://wiki.dh-electronics.com/index.php/Avenger96#Design_Files

1.1.1 Product Features

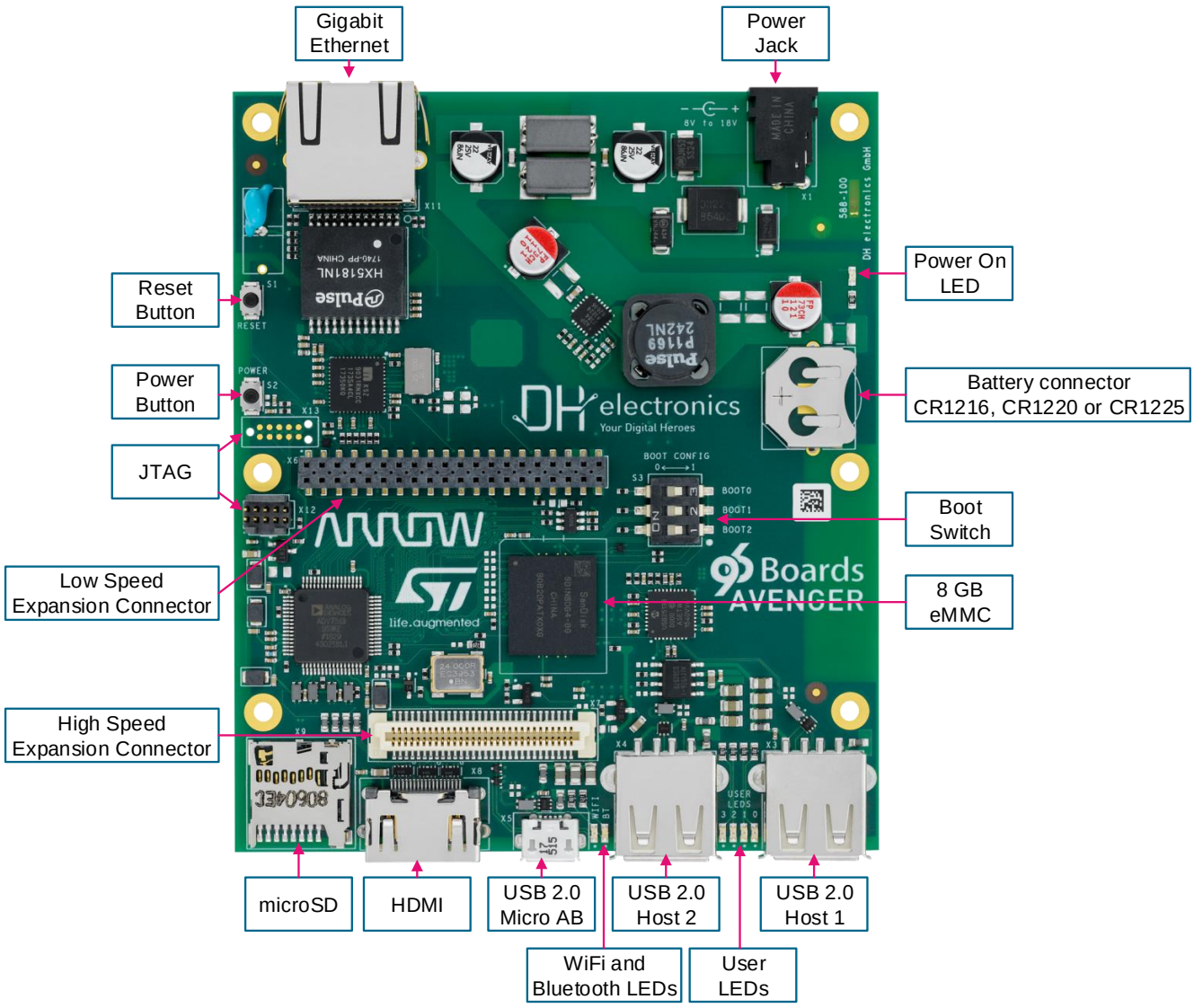
TECHNICAL DETAILS	
Processors	STM32MP157AAC 2x ARM® Cortex-A7 up to 650 MHz 1x ARM® Cortex-M4 up to 209 MHz
GPU	1x 3D GPU Vivante® @ 533 MHz - OpenGL® ES 2.0
PMIC	STPMIC1A
DDR3 DRAM	1024 Mbyte @ 533 MHz
eMMC Flash	8 Gbyte, v4.51 interface
NOR Flash	2 Mbyte, Quad SPI interface
E²Prom	128 byte
microSD Socket	UHS-I speed grade, v3.01
USB Host	2x type A, 2.0 high-speed
USB OTG	1x type micro-AB, 2.0 high-speed
HDMI	WXGA (1366x768) @ 60 fps, HDMI 1.4
WiFi / Bluetooth	WiFi 5 GHz & 2.4GHz IEEE 802.11a / b / g / n / ac Bluetooth® v4.2 (BR/EDR/BLE) PCB antenna
Ethernet	10 / 100 / 1000 Mbit/s, IEEE 802.3-compliant
Push-Buttons	Power and reset
Battery Socket	CR1216, CR1220 and CR1225
LEDs	4x green user controlled LEDs, 1x blue Bluetooth enabled, 1x yellow WiFi enabled, 1x red power supply okay
Boot Mode	3 bit boot mode switch
Debug Interface	JTAG interface via tag-connect
Supply (SYS_DCIN)	8 - 18 VDC

Temperature Range	0 - 40 °C
Dimensions	100 x 85 mm

1.1.2 Block diagram

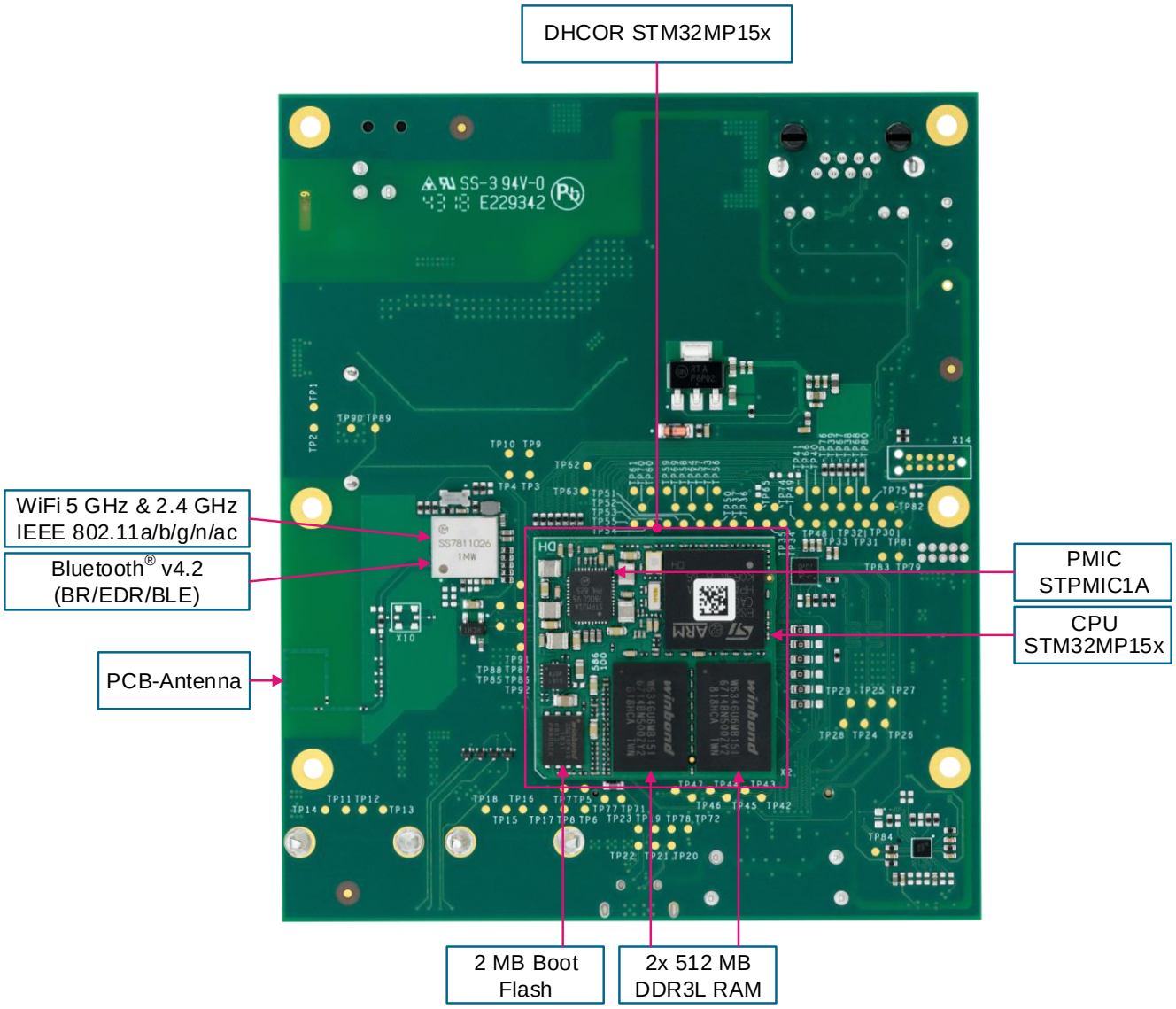


1.1.3 Key components



Please note:

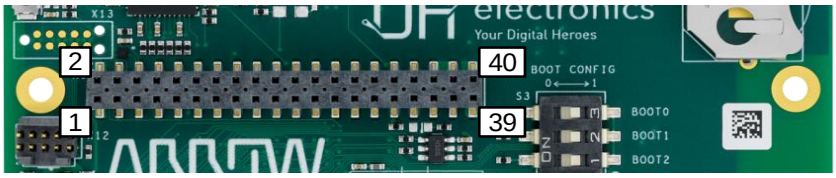
JTAG-connector X12 is not populated. See Chapter “1.1.7 JTAG-Connectors” for further information.


DHCOR STM32MP15x
WiFi 5 GHz & 2.4 GHz IEEE 802.11a/b/g/n/ac
Bluetooth® v4.2 (BR/EDR/BLE)
PCB-Antenna
PMIC STPMIC1A
CPU STM32MP15x
2 MB Boot Flash
2x 512 MB DDR3L RAM

1.1.4 Expansion Connectors

The Avenger96 has two expansion connectors which enables connection to a variety of Mezzanine boards. These boards allow to expand the interfaces of your Avenger96.

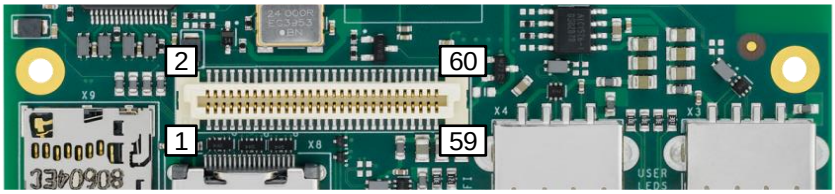
1.1.5 Low Speed Expansion Connector



TECHNICAL DETAILS	
UART 0	Rx / Tx / Rts / Cts
UART 1 (Standard interface to CPU)	Rx / Tx
SPI 0	Max. 52 Mbps
I2C 0	100 kHz, 400 kHz and 1000 kHz
I2C 1	100 kHz, 400 kHz and 1000 kHz
I2S	SAI interface in master mode
GPIOs	12 IOs
Supply	5 VDC, 1.8 VDS output, SYS_DCIN
RESET	Reset external request
POWER	Power on/off external request

GND	Pin 1	Pin 2	GND
UART0_CTS	Pin 3	Pin 4	PWR_BTN_N
UART0_TxD	Pin 5	Pin 6	RST_BTN_N
UART0_RxD	Pin 7	Pin 8	SPI0_SCLK
UART0_RTS	Pin 9	Pin 10	SPI0_DIN (MISO)
UART1_TxD	Pin 11	Pin 12	SPI0_CS
UART1_RxD	Pin 13	Pin 14	SPI0_DOUT (MOSI)
I2C0_SCL	Pin 15	Pin 16	PCM_FS
I2C0_SDA	Pin 17	Pin 18	PCM_CLK
I2C1_SCL	Pin 19	Pin 20	PCM_DO
I2C1_SDA	Pin 21	Pin 22	PCM_DI
GPIO-A	Pin 23	Pin 24	GPIO-B
GPIO-C	Pin 25	Pin 26	GPIO-D
GPIO-E	Pin 27	Pin 28	GPIO-F
GPIO-G	Pin 29	Pin 30	GPIO-H
GPIO-I	Pin 31	Pin 32	GPIO-J
GPIO-K	Pin 33	Pin 34	GPIO-L
+1V8	Pin 35	Pin 36	SYS_DCIN
+5V	Pin 37	Pin 38	SYS_DCIN
GND	Pin 39	Pin 40	GND

1.1.6 High Speed Expansion Connector



TECHNICAL DETAILS	
MMC/SD	4bit mode, v3.01 interface (only available if WiFi is not used)
MIPI CSI-2	2x Rx lanes
MIPI DSI	2x Tx lanes @ 1Gbit/s
I2C 0	100 kHz, 400 kHz and 1000 kHz
I2C 1	100 kHz, 400 kHz and 1000 kHz
USB Host	2.0 high-speed

SD_DAT0	Pin 1	Pin 2	CSI0_C+
SD_DAT1	Pin 3	Pin 4	CSI0_C-
SD_DAT2	Pin 5	Pin 6	GND
SD_DAT3	Pin 7	Pin 8	CSI0_D0+
SD_SCLK	Pin 9	Pin 10	CSI0_D0-
SD_CMD	Pin 11	Pin 12	GND
GND	Pin 13	Pin 14	CSI0_D1+
CSI_MCLK	Pin 15	Pin 16	CSI0_D1-
Not connected	Pin 17	Pin 18	GND
GND	Pin 19	Pin 20	Not connected
DSI_CLK+	Pin 21	Pin 22	Not connected
DSI_CLK-	Pin 23	Pin 24	GND
GND	Pin 25	Pin 26	Not connected
DSI_D0+	Pin 27	Pin 28	Not connected
DSI_D0-	Pin 29	Pin 30	GND
GND	Pin 31	Pin 32	I2C0_SCL
DSI_D1+	Pin 33	Pin 34	I2C0_SDA
DSI_D1-	Pin 35	Pin 36	I2C1_SCL
GND	Pin 37	Pin 38	I2C1_SDA
Not connected	Pin 39	Pin 40	GND
Not connected	Pin 41	Pin 42	Not connected
GND	Pin 43	Pin 44	Not connected
Not connected	Pin 45	Pin 46	GND
Not connected	Pin 47	Pin 48	Not connected
GND	Pin 49	Pin 50	Not connected
USB_D+	Pin 51	Pin 52	GND
USB_D-	Pin 53	Pin 54	Not connected
GND	Pin 55	Pin 56	Not connected
Not connected	Pin 57	Pin 58	GND
Not connected	Pin 59	Pin 60	+1V8 (Reserved)

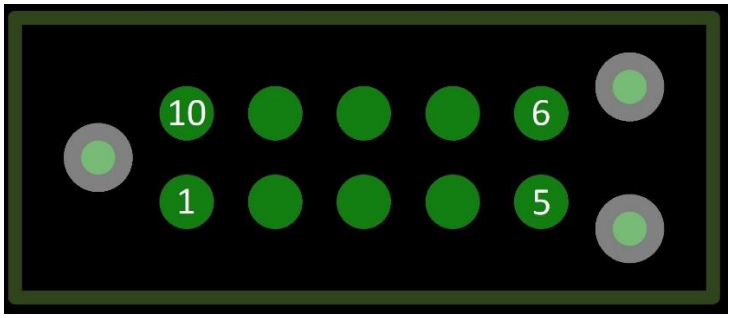
1.1.7 JTAG-Connectors

As a standard the JTAG-connector X12 is not populated on the Avenger96, but in case JTAG is needed, there are two ways to connect to it:

Tag-Connect (X13 and X14):

The Avenger96 has a standard Tag-Connect TC2050 footprint. With this design JTAG can be accessed with a standardized TC2050-IDC-NL-Cable. In order to hold the cable in place, there is also an TC2050-CLIP available, which locks the cable to the board.

The Tag-Connect footprint is placed on the top side of the Avenger96 as well as on the bottom side. This way the connector can be used on either side of the board.

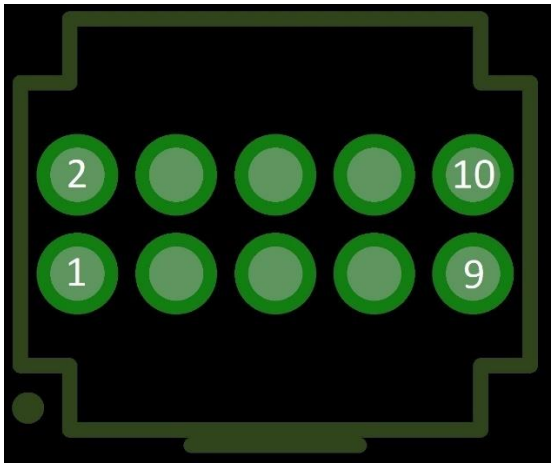


Pin	Net
1	+1V8
2	SYS_JTMS-SWDIO
3	GND
4	SYS_JTCK-SWCLK
5	GND
6	SYS_JTDO-SWO
7	SYS_JTRST
8	SYS_JTDI
9	Not Connected
10	Not Connected

JTAG-Connector (X12):

As an alternative to Tag-Connect there is also a footprint for a 10-Pin connector which uses the standard "Cortex 10-pin 0.05" JTAG/SWD Connector Pinout".

The footprint is designed for the Samtec FTSH-105-01-F-D-K, but it is also possible to use other pin headers with the same dimensions like Mouser 855-M50-3500542, Harwin M50-3500542 or similar.



Pin	Net
1	+1V8
2	SYS_JTMS-SWDIO
3	GND
4	SYS_JTCK-SWCLK
5	GND
6	SYS_JTDO-SWO
7	Not Connected
8	SYS_JTDI
9	GND (10k Pull-down)
10	SYS_JTRST

1.2 Set Up Guide

This short guide leads you through the first steps to start exploring your Avenger96.

In the box you can find the Avenger96 which comes with a pre-installed Linux on its eMMC storage as well as a bootloader installed on the NOR-Flash. **Unfortunately, the mounted version of the SoC (first engineering samples) can't boot directly from the eMMC.** In order to start the Avenger96 then, all you have to do is select the BOOT-Mode (chapter 1.2.2) "NOR-Flash", which loads the bootloader first. Once the bootloader is running, it starts up Linux from the eMMC. This way you can make use of the eMMC as a storage for your operating system.

Alternatively, the image including the bootloader U-Boot can also be started from the **microSD card**. This is the easiest way to start your first self-build image.

Note: The Avenger96 board comes with pre-installed (eMMC) demo image, but this image is outdated (Note: Avenger96 hardware was shipped years ago) and it is recommended to download the latest demo image (see chapter 1.2.1) or build your own up-to-date image (see chapter 2.1) and start from SD card.

1.2.1 Step by step guide to use a microSD card image

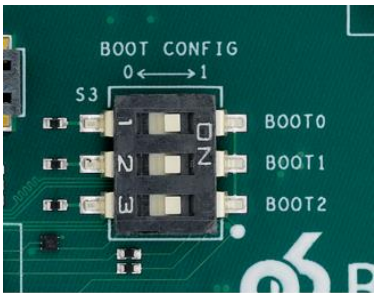
The board can be put in operation by following steps:

1. Download dh-image-demo binary: https://github.com/dh-electronics/kas-dhsom/releases
2. Flash the demo image to microSD card.
    - 8 Gbyte or larger capacity
    - Balena etcher tool could be used: https://etcher.balena.io/#download-etcher
3. Make sure the boot switch is set to boot from "SD-Card" (See chapter 1.2.2).

   Note: Select "NOR-Flash" boot to start pre-installed image.
4. Connect a display via the HDMI-connector.
5. Connect keyboard and mouse to the host USB ports.
6. Connect serial adapter to access Linux console (See chapter 1.2.4).
7. Connect a proper power supply (See chapter 1.2.3).
8. Plug the power supply into the power socket → Linux starts up!

1.2.2 Boot Mode

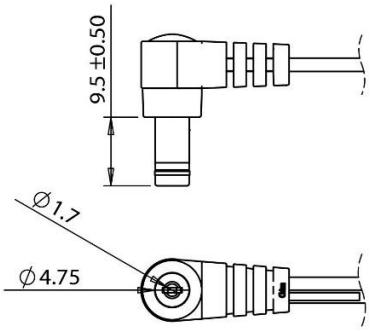
The Avenger96 supports multiple boot options which are selected by the DIP-switch S3. To select a logical "1" a switch needs to be pushed to the right. Therefore, a logical "0" is set by pushing the switch to the left. The numeration of these pins is printed next to the switch on the circuit board.



BOOT Mode	Comments	BOOT 2 (Switch 3)	BOOT 1 (Switch 2)	BOOT 0 (Switch 1)
UART and USB	USB high-speed Device	0	0	0
NOR-Flash (Standard)	On Quad SPI	0	0	1
eMMC (See „1.2 Set Up Guide“)	On SDMMC2	0	1	0
NAND-Flash (Not available)	SLC NAND Flash	0	1	1
Reserved (NoBoot)	Get boot access without boot from Flash memory	1	0	0
SD-Card	On SDMMC1	1	0	1
UART and USB	USB OTG	1	1	0
Serial NAND Flash (Not available)	NAND flash on Quad SPI	1	1	1

1.2.3 Power Supply

The Avenger96 is designed to work with a standard 96Boards power connector which can be found on the 96Boards website: https://www.96boards.org/product/power/



Dimensions shown in Millimeter

Suitable power supplies are:

- Avnet AES-ACC-U96-4APWR
- IDEAL POWER 25HK-AB-120A250-CP6

As an alternative to these power supplies, you can also use any other power cord which has the same dimensions like shown in picture above as long as it has an output voltage between 8 V DC to 18 V DC (typically 12 V DC are used) and provides a power consumption of up to 24 W.

In case you just need the connector only, you can use a cable like this one:

https://www.arrow.de/products/053-0198r/tensility-international

Note: The board contains reverse polarity protection.

1.2.4 Serial console connection

U-Boot and Linux serial access is available on the debug UART connected to pin 11 and 13 on the low-speed expansion connector (UART1 - see above). One can use the Tresor board from Arrow to connect to this UART via standard USB adapter:

https://www.arrow.com/en/products/tresor/arrow-development-tools

Any other USB-serial converter can be attached to the UART1 pins but care must be taken that signal levels are 1.8V CMOS.

1.2.5 HDMI Display

The Avenger96 supports a display resolution up to WXGA (1366 x 768) at a refresh rate of 60 fps.

Therefore, you can connect your Avenger96 to any HDMI display which supports such a resolution.

As an alternative, there is also a 7" HDMI display with a capacitive touch screen designed specific for the use with a 96Boards: https://www.arrow.de/products/96boards-display-7/linksprite-technologies-inc

2 Software

The Avenger96 board is completely upstreamed to mainline Linux and U-Boot.

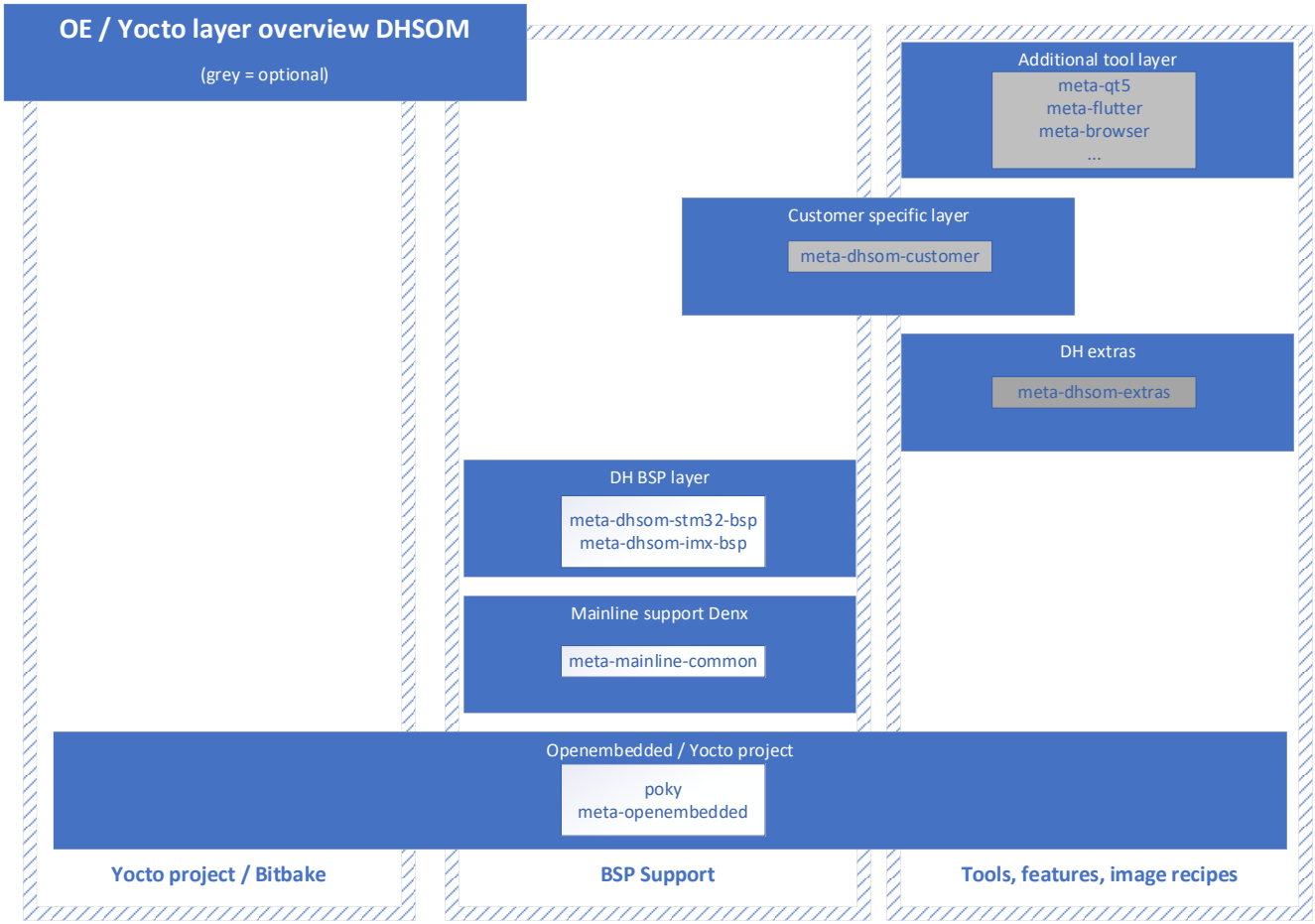
- Kernel Device Tree name: stm32mp157a-dhcor-avenger96.dts
- U-Boot Device Tree name: stm32mp15xx-dhcor-avenger96.dts

The Board Support Package (BSP) is always based on latest LTS OE / Yocto and LTS Linux Kernel version.

2.1 Building your own image

A pre-configured OE / Yocto is provided via the kas tool, which is available on GitHub. If you want to build your own Avenger96 image, you have to follow the steps from the README.md on GitHub:
https://github.com/dh-electronics/kas-dhsom

2.2 OE / Yocto layer overview



Yocto layers essentially consist of build instructions and definitions for devices and the entire software scope, as well as build instructions, configurations and patches for individual software packages. Build sources of the

software packages are usually not part of a Yocto layer. However, the Yocto layers reference the sources of supply repositories for the sources of the respective software packages. If configured for the build, these are downloaded within the Yocto build.

2.2.1 poky / meta-openembedded

All Yocto projects at DH electronics are based on the "poky" layer or the "meta-openembedded" layer collection. These layers determine the basic structure of a Yocto build. The "poky" layer provides the "bitbake" tool (build tool), as well as classes for the implementation of recipes/images/machine configs, busybox and licensing documentation. The "meta-openembedded" layer collection includes numerous basic tools/features that are already "pre-installed" in classic distributions such as Debian. (e.g. systemd, network tools, text editors, window managers, multimedia tools, benchmark tools, etc.)

2.2.2 meta-mainline-common

This layer is managed by Denx and made publicly available. This forms the basis for the implementation of the DHSOM BSPs. "meta-mainline-common" is the central adjusting screw for U-Boot, Linux Kernel and Mesa versions. For mainline-based projects, this determines which patch level of the versions is used for the build.

2.2.3 meta-dhsom-stm32-bsp

This layer provides BSP support specifically for DHSOM standard devices. It includes respective machine configs, additional module/device-specific patches/configurations for Kernel, U-Boot, etc. that is not (yet) available mainline. As well as defining the SD card layout for wic images.

2.2.4 meta-dhsom-extras

"meta-dhsom-extras" includes DH-specific rootfs adjustments, fine-tuning of software packages, distribution configuration and the recipe for the demo image "dh-image-demo". This layer includes support for all DHSOM modules.

2.2.5 meta-dhsom-customer

"meta-dhsom-customer" is representative of a customer-specific layer. All changes and extensions that deviate from the DHSOM standard are defined here. This can range from an extension of the Kernel to a full customer image.

2.2.6 Other layers

Often not all necessary tools are provided by a meta-openembedded layer. Additional open source layers are offered for appropriate support. Examples include QT5, QT6, Flutter and Chromium.

2.3 kas-dhsom

The “kas-dhsom” repository is not a Yocto layer, but an instance for defining specified Yocto configurations. “kas-dhsom” forms the central and simplified entry point for manual and CI-driven builds. It makes it possible to provide fixed configurations of Machine Config, Image Recipe and Layers in the form of YAML files (*.yml), which only need to be called with the kas tool (from Siemens).

2.4 Image scope

Which packages are explicitly implemented in the image is determined by the respective image recipe, as well as the configuration of the machine (board-specific) and the distribution guidelines. The layers are structured hierarchically and expand or overwrite each other. This enables precise adaptation to the requirements, but also requires a corresponding definition of all the tools that are required.
Since there are no package repositories for the Yocto project, subsequent installation via a package manager is difficult or even impossible.

2.5 Pre-installed image (outdated)

Note: The Avenger96 board comes with pre-installed (eMMC) demo image, but this image is outdated (Note: Avenger96 board hardware was shipped years ago) and it is recommended to download the latest demo image (see chapter 1.2.1) or build your own up-to-date image (see chapter 2.1) and start from SD card.

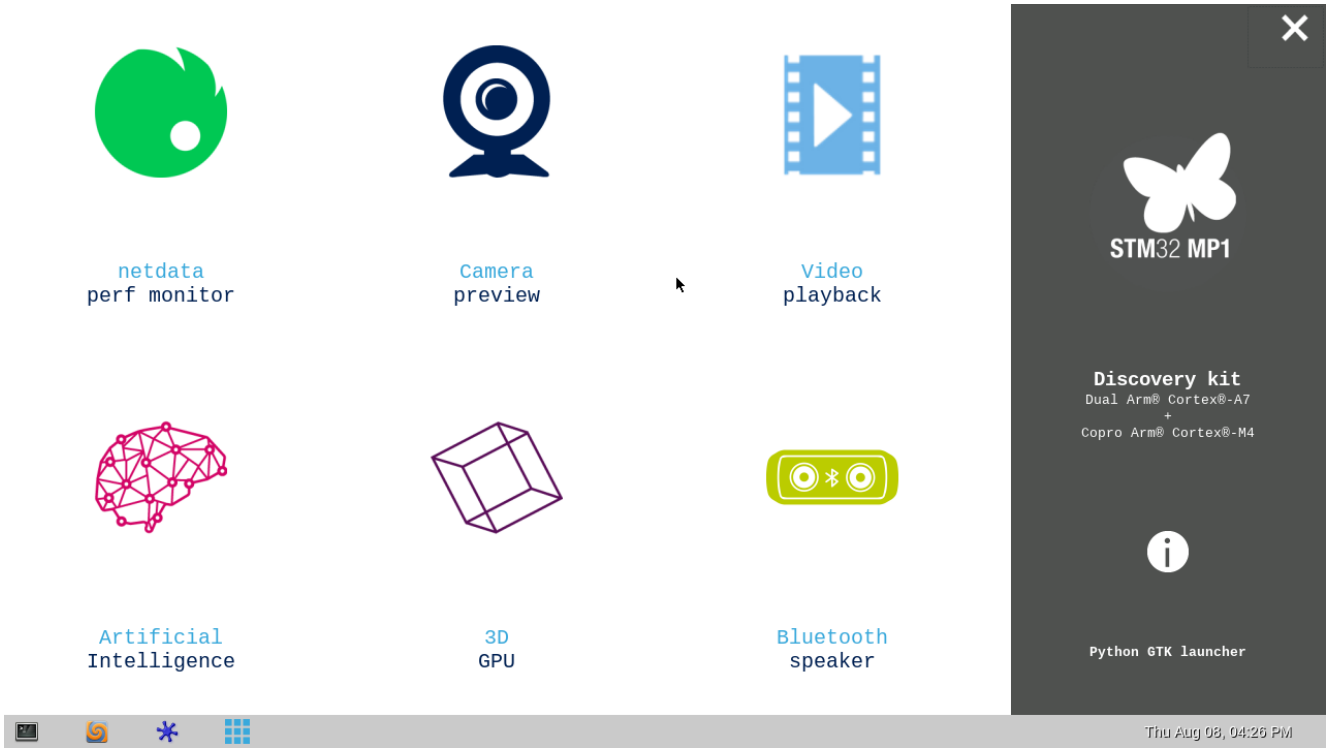
The current Avenger96 "Avenger96_v3.2_2019-08-09" image is based on OpenSTLinux package.

U-Boot version: U-Boot 2018.11-stm32mp-r2

Linux version: Linux stm32mp1-av96 4.19.9

Linux Distribution: ST OpenSTLinux - Weston - (A Yocto Project Based Distro)

When Linux starts up the standard Weston desktop will appear on the display. A terminal window can be opened by clicking on the terminal icon in the down left corner:



Known image limitations / missing features:

- SAI support
- MIPI DSI and MIPI CSI are not yet supported
- HDMI CEC
- Battery and power button

2.5.1 Configuring ethernet

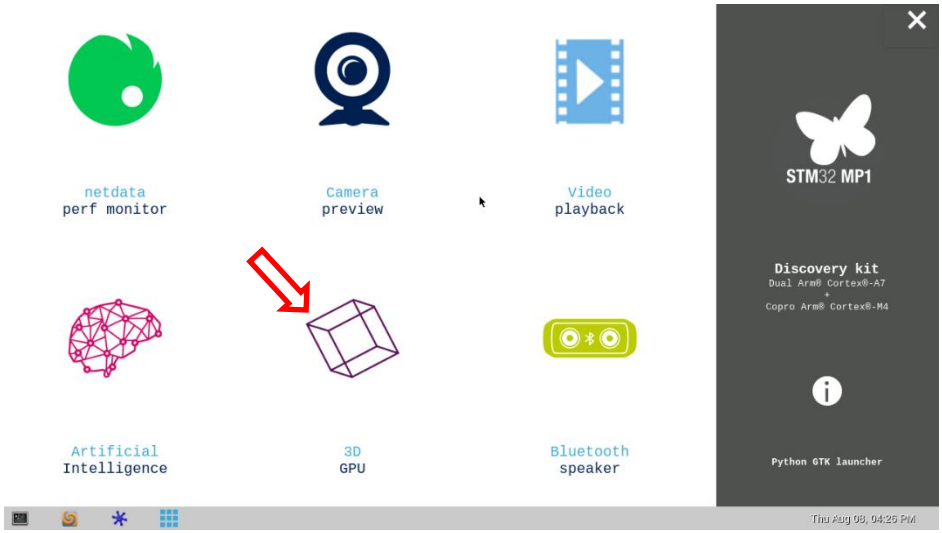
When an ethernet cable is plugged into the RJ45 connector, autonegotiation will start automatically and the network service will try to get IP address over DHCP. The progress of network card configuration can be checked by executing:

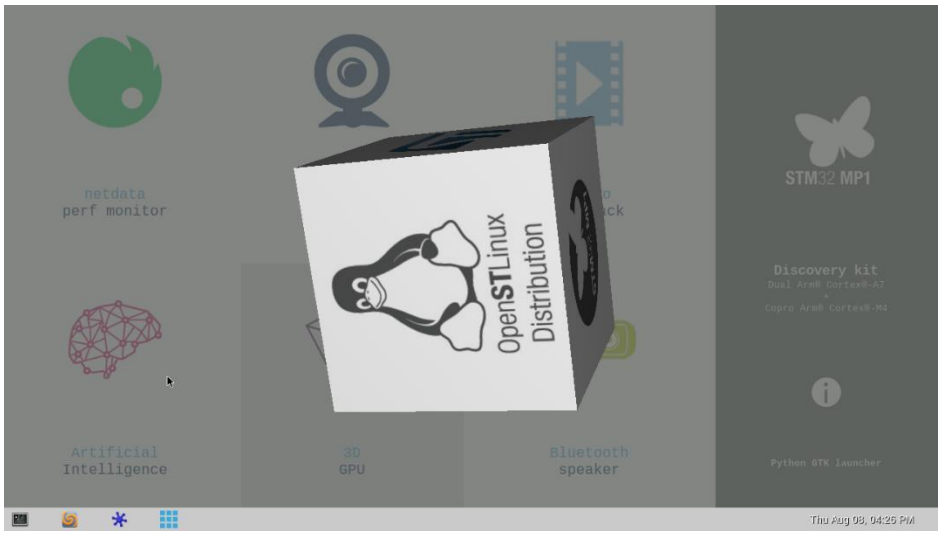
```
# ifconfig eth0
```

When eth0 has already received IP address the network connection can be tested by

```
# ping google.com
```

2.5.2 Starting ST GPU Demo





2.5.3 Starting glmark2 GPU Demo

The GPU demo glmark2 can be started and stopped with the marked button.

2.5.4 Starting glmark2 GPU Demo manually

In case more options (e.g. different screen size) are needed, the demo can also be started manually with different settings:

First open the Terminal with the Icon on the bottom left.

Then use the following command for showing all the options for "glmark2":

```
# glmark2-es2-wayland -h
```

Finally, all you have to do is starting it with the desired settings, like for instance in windowed mode with annotation turned on:

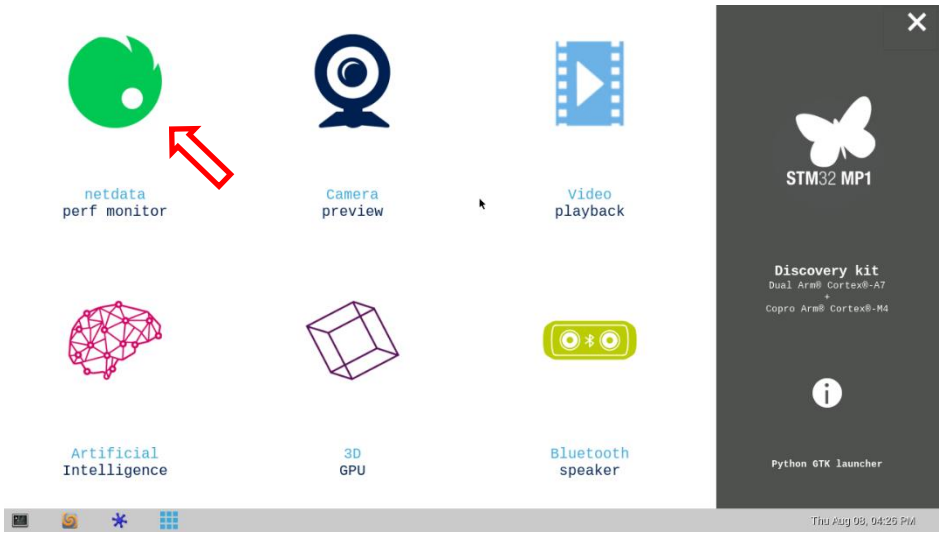
```
# glmark2-es2-wayland --size 800x600 --annotate
```

For displaying it in full screen, use this command:

```
# glmark2-es2-wayland --fullscreen --annotate
```

To close glmark2, all you have to do is clicking in the Terminal and cancel the current command by pressing "CTRL + C" on your keyboard. In order to cancel it while it is running in full screen, you have to switch back to the Terminal first by using the keystroke combo "CTRL + TAB" followed by "CTRL + C" to quit it.

2.5.5 Configuring WiFi by Weston desktop



2.5.6 Configuring WiFi manually

Before configuring WiFi it is best to shut down other interfaces:

```
# ifconfig eth0 down
# ifconfig usb0 down
```

Now let's start WiFi:

```
# ifconfig wlan0 up
# iw dev wlan0 scan | grep ssid -i
```

The last command will list the WiFi hot spots in range. If a WiFi network uses WPA authentication the following commands will connect to it:

```
# cd
# cp /etc/wpa_supplicant.conf .
# wpa_passphrase [SSID] [passphrase] >> ./wpa_supplicant.conf
```

Please insert the name of the WiFi network and the passphrase in the command above without quotes or the brackets.

Now connect to the WiFi network.

```
# wpa_supplicant -B -Dnl80211 -iwlan0 -cwpa_supplicant.conf
```

The connection can be checked by:

```
# iw dev wlan0 link
```

When connected to the network IP address can be requested by:

```
# dhclient wlan0
```

2.5.7 SSH server

The pre-installed Linux automatically starts SSH server on all interfaces. One can connect to the board by executing the following command on the PC (which has access to the same network as the Avenger96):

```
# ssh root@[IP address of board]
```

2.5.8 Show Image Information

Show image information:

```
# cat /etc/issue
```