

StampA5D3x/PortuxA5/PanelA5

Quickstart Guide

StampA5D3x/PortuxA5/PanelA5: Quickstart Guide

Copyright © 2015 taskit GmbH

All rights to this documentation and to the product(s) described herein are reserved by taskit GmbH.

This document was written with care, but errors cannot be excluded. Neither the company named above nor the seller assumes legal liability for mistakes, resulting operational errors or the consequences thereof. Trademarks, company names and product names may be protected by law. This document may not be reproduced, edited, copied or distributed in part or in whole without written permission.

This document was generated on 2016-10-24T17:19:13+02:00.

Table of Contents

Introduction	v
1. Powering up the Board	1
1.1. The Debug Interface	1
1.2. The Network Connection	3
1.3. Setting a Password	3
2. Starting Development	4
2.1. Installing the Compiler Toolchain	4
2.2. Compiling the Kernel	4

List of Figures

1.1. PortuxA5 DBGU Connection	1
1.2. PanelA5 DBGU Connection	2

Introduction

This document describes the first steps in setting up the PanelA5/PortuxA5. After completing this guide, you will be able to:

- Power up the device
- Get access to the debug UART
- Configure basic settings
- Set up the cross-compiler toolchain
- Compiled the Linux kernel

This guide assumes, that you have PanelA5-, PortuxA5- or StampA5-Starterkit and a computer with Debian Wheezy installed. Other distributions might also work, especially Debian based ones like Ubuntu, but there were not tested during the writing of this guide. A general familiarity with Linux is helpful.

1. Powering up the Board

Before powering up the board, you should connect the USB FTDI cable to the debug interface (for details see Section 1.1, “The Debug Interface”) and the RJ45 ethernet interface to your development computer or a network switch. After that, connect the power supply to the device. In case of the PanelA5, connect the 24V power supply to connector X3. Make sure the polarity is correct. In case of the PortuxA5, connect the micro USB power supply to the micro USB port.

1.1. The Debug Interface

It is recommended to setup the connection to the serial debug interface (DBGU). Use the supplied FTDI USB cable to connect both the PC and the DBGU interface of your board. On PanelA5, this is connector X20, on PortuxA5 it is X6. The pins are marked BL (black), OR (orange) and YE (yellow). Connect the color coded pins of the FTDI cable to the corresponding pins of the DBGU interface. See Figure 1.1, “PortuxA5 DBGU Connection” and Figure 1.2, “PanelA5 DBGU Connection” for reference.

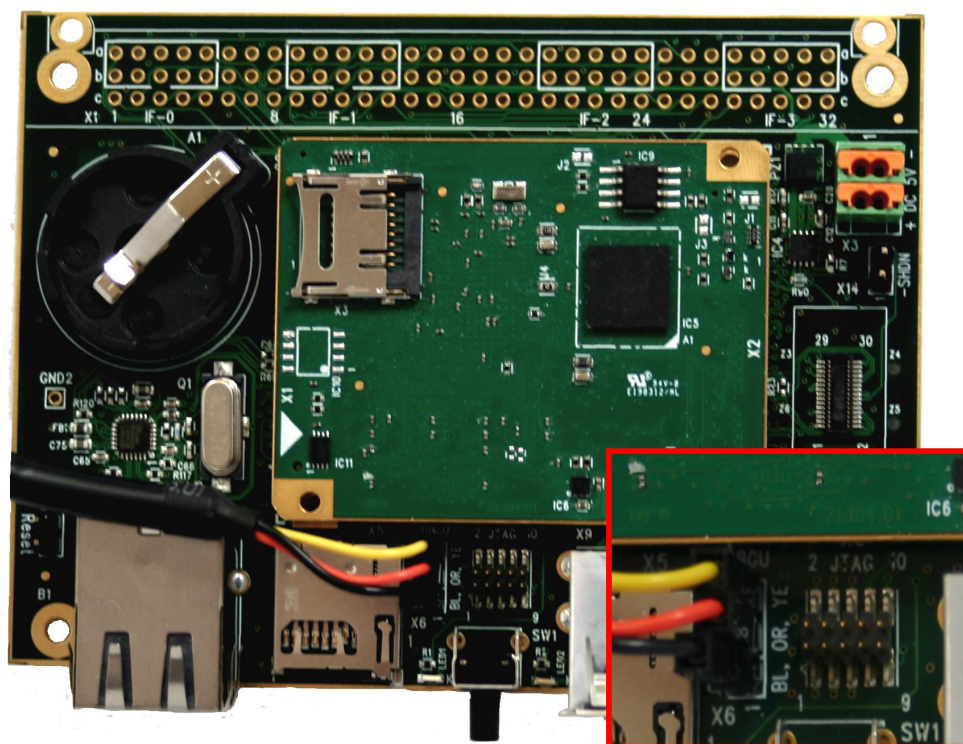


Figure 1.1. PortuxA5 DBGU Connection

Powering up the Board

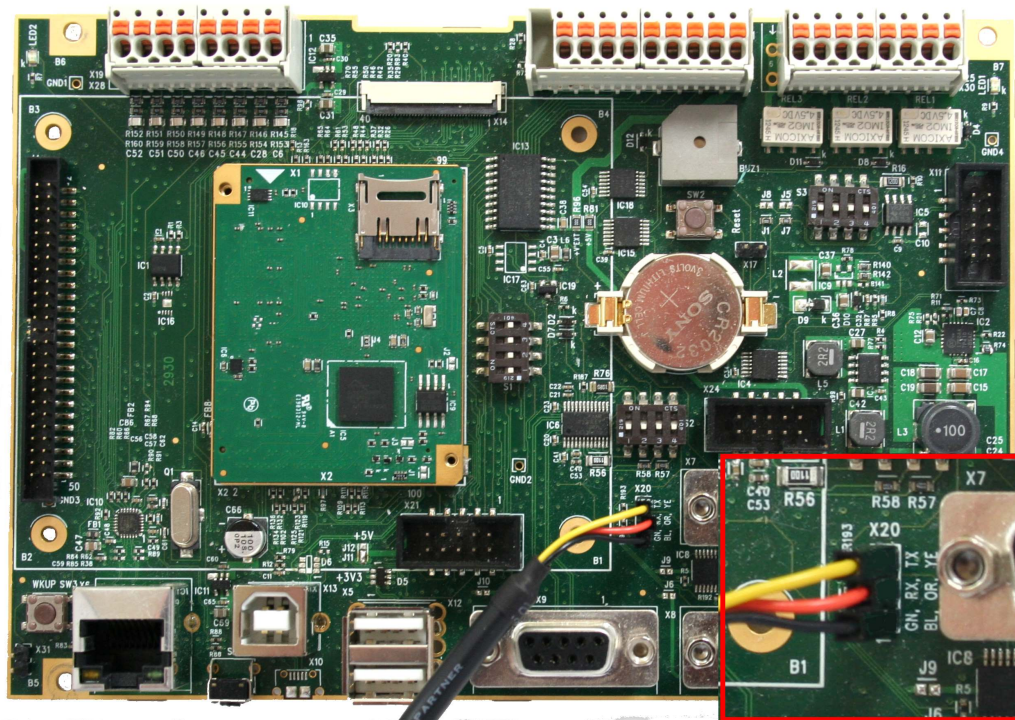


Figure 1.2. PanelA5 DBGU Connection

On the development host, the FTDI cable should normally be recognized as device `/dev/ttyUSB0`. The number can be different depending on how many USB serial converters are currently in use on the host.

To make use of the serial connection, we need serial terminal program. We will use `picocom` here, because it is relatively simple. To install it, open an X terminal program, e.g. `gnome-terminal` and enter the following command:

```
sudo apt-get install picocom
```

It will most probably ask for your password to continue.

You can now start the serial terminal with the following command:

```
picocom -b 115200 /dev/ttyUSB0
```

If you get a permission denied message, you will have to add your user to the `dialout` group. You can do that with the following command (it is assumed, that your user account is called `developer`; change the command, if you used another name):

```
sudo adduser developer dialout
```

Again, you might be asked to provide your password. For this change to be effective, you have to logout and relogin again. `Picocom` should now start without problems.

If you had already plugged in the power supply, you can now press the enter key in the terminal, where you started `picocom`, and a login prompt should become visible.

```
stampa5d36 login:
```

You can now login with username **root**. No password will be needed. You should now be on the command prompt.


```
root@stampa5d36:~#
```

1.2. The Network Connection

During the development, you will need to have access to the device via the network interface. Therefore you should set up the network to have the correct settings. If you have a DHCP server and have no problem with checking the currently used IP address, you can skip the network configuration step. You can discover the IP address using the `ifconfig` command.

```
root@stampa5d36:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr CE:29:C2:BB:D9:A6
          inet addr:192.168.0.50  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::cc29:c2ff:febb:d9a6/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2726 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:275050 (268.6 KiB)  TX bytes:32214 (31.4 KiB)
          Interrupt:42 Base address:0xc000
```

For a static configuration, you need to edit the file `/etc/network/interfaces`.

```
root@stampa5d36:~# vi /etc/network/interfaces
```

Add the following section to the file. Adjust the values to match your network.

```
iface eth0 inet static
    address 192.168.0.50
    netmask 255.255.255.0
    gateway 192.168.0.1
```

Now comment out the following line to disable DHCP for `eth0`.

```
# iface eth0 inet dhcp
```

To finish the configuration, save the file and reinitialize the interface.

```
root@stampa5d36:~# ifdown eth0
root@stampa5d36:~# ifup eth0
```

If you connect the device directly to your development computer, set up its network configuration accordingly.

1.3. Setting a Password

By default, no password is set for the root account. That is not only dangerous for a final product, but also prevent you from logging into the system via SSH. As SSH/SCP is later needed, you should set a password for the root account on the device.

```
root@stampa5d36:~# passwd
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
passwd: password changed.
```


2. Starting Development

In case that this has not already been done, install a Linux-PC as your development PC, preferably using Debian Linux or a derivate distribution.

2.1. Installing the Compiler Toolchain

Install the toolchain from the DVD to you development PC. The DVD contains a shell script within the directory "toolchain" called

```
oecore-x86_64-cortexa5t2hf-vfp-toolchain-nodistro.0.sh
```

which will unpack the tar archive of the toolchain to the PC. Default directory for the unpacked files is

```
/usr/local/oecore-x86_64
```

After installing the toolchain, call the following shell script using "source":

```
source environment-setup-cortexa5t2hf-vfp-oe-linux-gnueabi
```

This will set important environment variables like

```
ARCH=arm  
CROSS_COMPILE=arm-oe-linux-gnueabi-
```

necessary for the ARM-Linux cross compiler. As these setting are not persistent, the script has to be run once for every compiling session.



Note

The "menuconfig" program for the Linux kernel configuration will not work any more after running the script. This can be "repaired" by unsetting the variables `PKG_CONFIG_PATH` and `PKG_CONFIG_SYSROOT_DIR`.

```
unset PKG_CONFIG_PATH PKG_CONFIG_SYSROOT_DIR
```

Don't do this, when you actually compile applications, as they actually may need these variables to detect and link libraries.

2.2. Compiling the Kernel

1. Download the Linux 3.16 kernel sources from the Internet, e.g. from kernel.org.
2. Download all the Linux patches from

```
https://gitlab.com/taskit/meta-taskit/tree/master/recipes/linux/linux-3.16
```

Use the program "patch" to install the patches into the Linux source files.

3. You may wish to copy the taskit default configuration for the Linux kernel. To do so, copy the file

```
https://gitlab.com/taskit/meta-taskit/blob/master/recipes/linux/linux-3.16/stampa5d36/defconfig
```

to the file.config in the source directory of Linux. In order to get all the other default configurations of Linux without having to enter them by hand, call

```
make olddefconfig
```

thereby completing the file ".config".

4. You can now compile the kernel with make.