

# Programmierung

in diesem Abschnitt habe ich ein paar von mir häufig verwendete Code-Schnipsel hinterlegt. vielleicht ist der eine oder andere hilfreich ;)

[GIT SVN](#)

[bash](#)

[javascript](#)

[mysql](#)

[Python](#)

[PHP](#)

[VSCode](#)

## Batch

## C/C++

## Makefile

```
$(warning $$src is [${src}])
```

string-manipulation:

[https://ftp.gnu.org/old-gnu/Manuals/make-3.79.1/html\\_chapter/make\\_8.html#SEC77](https://ftp.gnu.org/old-gnu/Manuals/make-3.79.1/html_chapter/make_8.html#SEC77)

## Linux

### Fixes (für backports)

```
Cc: stable@vger.kernel.org
```

in den Signed-Off-Bereich

Quelle: <https://www.kernel.org/doc/html/latest/process/stable-kernel-rules.html>

## debugging

## Using dynamic debug

Add options:

```
CONFIG_DYNAMIC_DEBUG=y
CONFIG_DEBUG_FS=y
```

And enable it from userspace (here all for network drivers):

```
echo "file drivers/net/* +p"> /sys/kernel/debug/dynamic_debug/control
```

## Manual by adding new messages

```
printk(KERN_ALERT "DEBUG: Passed %s %d
val:0x%x\n", __FUNCTION__, __LINE__, (unsigned int)val);
```

```
dev_err(dev, "DEBUG: Passed %s %d val:0x%x\n", __FUNCTION__, __LINE__, val);
```

## cmdline

```
//register params
static int startreg=-1;
module_param( startreg, int, S_IRUGO );
static int regcount=10;
module_param( regcount, int, S_IRUGO );

//look for module-name
static struct mdio_driver realtek_mdio_driver = {
    .mdiodrv.driver = {
        .name = "realtek-mdio",//<<<<

//pass cmdline arguments
realtek-mdio.startreg=0x1300 realtek-mdio.regcount=50
```

- <http://embeddedguruji.blogspot.com/2018/12/passing-parameters-to-linux-device.html>
- <https://ezs.kr.hsnr.de/TreiberBuch/html/sec.treiberparam.html>

## Links

- [elixir@bootlin](#)
- Patchwork
  - [Mediatek](#)
  - [Rockchip](#)
  - [dt-bindings](#)
- GIT:
  - [stable torvalds linux-next](#)

- [net net-next](#)
- [drm mtk next](#)[drm mtk dts next](#)
- [Rockchip SoC/DTS](#)
- [dt-bindings](#)
- [pinctrl-next](#)

## U-Boot

### debugging

```
#define DEBUG 1
```

vor "#include <common.h>" um vorhandene/zusätzliche debug() anzuzeigen

```
debug("## %s:%d...\n", __FILE__, __LINE__);
```

oder manuell (wird immer angezeigt):

```
printf("%s:%d val:0x%0x\n", __FUNCTION__, __LINE__, (u32)val);
```

### Links

- [elixir@bootlin](mailto:elixir@bootlin)
- [Uboot-Patchwork Mailinglist Archiv](#)
- [uboot gitlab](#)

## DTC

<http://thesoulofamachine.blogspot.com/2016/07/linux-dts-compilation-using-dtc-compiler.html>

DTB dekompileieren:

```
sudo apt-get install device-tree-compiler
```

```
dtc -I dtb -O dts sun8i-h3-orangepi-pc.dtb -o /tmp/tmp.dts  
less /tmp/tmp.dts
```

#Alternative ohne temporäre Datei

```
dtc -I dtb -O dts ../build/arch/arm64/boot/dts/rockchip/rk3568-bpi-r2-  
pro.dtb | less
```

#aus sysfs

```
dtc -I fs -O dts /sys/firmware/devicetree/base | less
```

## dtbs\_check

```
sudo apt install swig yamllint
```

```
pip3 install dtschema --upgrade  
pip3 show dtschema
```

```
ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- make dt_binding_check  
DT_SCHEMA_FILES=Documentation/devicetree/bindings/ata/ahci-platform.yaml  
ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- make defconfig #dtbs_check need  
kernel-config  
ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- make dtbs_check  
DT_SCHEMA_FILES=Documentation/devicetree/bindings/ata/ahci-platform.yaml  
ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- make multi_v7_defconfig  
ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- make dtbs_check  
DT_SCHEMA_FILES=Documentation/devicetree/bindings/ata/ahci-platform.yaml  
  
#maybe need to add dir to path  
PATH=$PATH:~/local/bin
```

## DTS

Mapping wifi Eeprom data to pcie slot:

<https://forum.banana-pi.org/t/802-11ac-module-gives-max-6dbi-transmitter-power/14238/23>

## CSS

## HTML

From:  
<https://wiki.fw-web.de/> - **FW-WEB Wiki**

Permanent link:  
<https://wiki.fw-web.de/doku.php?id=programming:start&rev=1692983467>

Last update: **2023/08/25 19:11**

