

NEUE ARM32 SOCS

EIN SUBSYSTEM-RUNDGANG AUS DER HOBBY-PERSPEKTIVE

HEIKO STÜBNER

- Kernel-Entwickler bei BQ (eReader, Smartphones, Tablets)
- embedded-Einstieg 2008 mit Openmoko Freerunner
- erste Kernelschritte 2010 auf dem Thalia Oyo
- Maintainer des Rockchip SoC supports im Mainline-Kernel

INTRO: AUSGANGSPUNKT

- Gerät mit serieller Schnittstelle
- Kernel-Drop des Herstellers
- entspricht in den 99,9% der Fälle nicht Mainline-Standards
- oft kleine "Community" die am Hersteller-Kernel/Android bastelt
- keine Dokumentation für den Prozessor
- eventuell Schaltplan des Boards

INTRO: BLOCK DIAGRAMM

- SoC: ARM-Kern + herstellerspezifische Peripherie
- ARM32-Kern: ARM9, Cortex-A9 .. Cortex-A17
- Peripherie: Controller für I2C, SPI, MMC, GPU/LCDC, etc
- daran angeschlossen: Sensoren, Wifi, Buttons, etc

INTRO: MULTIARCH UND DEVICETREE

- früher: Kernel wurde für bestimmten Prozessortyp kompiliert
- heute: keine SoC-spezifischen Kernel mehr, sondern multiarch
- Umstellung alter Architekturen immernoch im Gange
- multi_v7_defconfig als Basis
- Devicetrees statt Board-Files
- arch/arm/mach-foo optional und nur wenn hardspezifischer Code nötig

INTRO: KERNEL-IMAGE AUF DAS GERÄT

- verschiedene herstellerspezifische Wege
- Fastboot + Android Boot.img
- U-Boot Boot-Image
- Voodoo - aber oft bereits reverse-Engineered
- Hilfe auf xda-developer oder ähnlichen Foren/Listen
- Vorteil: Hersteller bleiben ihrem Flash-Verfahren meist lange treu
- Rockchip: Prüfsumme um Kernel-Image + spezielles USB-Protokoll
- rkrcrc -k und rkflashtool als freie Implementierung

LOG: ERSTER STARTVERSUCH - STILLE

DDR Version 1.04 20130517

In

DDR3

300MHz

Bus Width=32 Col=10 Bank=8 Row=15 CS=2 Die Bus-Width=16 Size=2048MB

Memory OK

OUT

BUILD=====5

F:32 1061 2 0 40

GetRemapTbl flag = 0

OK! 66400

unsigned!

SecureBootEn = 0 0

Boot ver: 2013-05-18#1.20

start_linux=====78620

508144 Starting kernel...@0x60408000

DEBUGLL: ERSTE LEBENSZEICHEN

- Serielle Ausgabe während des zeitigen Boots
- kommt ohne Interrupts und Clocks aus
- nutzt UART-Einstellungen des Bootloaders weiter
- nicht multiplatform-fähig
- Alternative echte Earlycon im seriellen Treiber
- in arch/arm/mach-foo/include/mach/debug.S bzw. arch/arm/include/debug/*
- meist bereits eine Implementierung im Herstellerkernel

DEBUGLL: EINFACHES INTERFACE

```
.macro  addruart, rp, rv, tmp
ldr    \rp, =CONFIG_DEBUG_UART_PHYS
ldr    \rv, =CONFIG_DEBUG_UART_VIRT
.endm

        .macro  senduart, rd, rx
strb   \rd, [\rx, #UART01x_DR]
.endm

        .macro  waituart, rd, rx
1001:  ldr    \rd, [\rx, #UART01x_FR]
      tst   \rd, #UART01x_FR_TXFF
      bne   1001b
      .endm

        .macro  busyuart, rd, rx
1001:  ldr    \rd, [\rx, #UART01x_FR]
      tst   \rd, #UART01x_FR_BUSY
      bne   1001b
      .endm
```

DEBUGLL: WO EINBINDEN

- arch/arm/Kconfig.debug
- DEBUG_LL_INCLUDE - Auswahl des zu nutzenden debug includes
- DEBUG_UART_PHYS - physikalische Adresse des UART
- DEBUG_UART_VIRT - virtuelle Adresse des UART
- DEBUG_UNCOMPRESS - Meldungen des Decompressors aktivieren
- EARLY_PRINTK - Early-Console auf dem DebugLL

ADDRESS-MAPPING

- 4GB Adressraum
- Adressraum des Arbeitsspeichers
- Adressraum der Peripheriekomponenten
- Quelle 1: Prozessor-Handbuch
- Quelle 2: arch/arm/mach-foo/include/mach/io-irgendwas.h
- Quelle 3: arch/arm/boot/dts/foo.dtsi

MACH-RK3188/INCLUDE/MACH/IO.H

```
#define RK30_IPP_PHYS      0x10110000
#define RK30_IPP_SIZE     SZ_16K
#define RK30_RGA_PHYS     0x10114000
#define RK30_RGA_SIZE     SZ_8K
#define RK30_I2S1_2CH_PHYS 0x1011a000
#define RK30_I2S1_2CH_SIZE SZ_8K
#define RK30_SPDIF_PHYS   0x1011e000
#define RK30_SPDIF_SIZE   SZ_8K
#define RK30_UART0_PHYS   0x10124000
#define RK30_UART0_SIZE   SZ_8K
#define RK30_UART1_PHYS   0x10126000
#define RK30_UART1_SIZE   SZ_8K
#define RK30_L2C_PHYS     0x10138000
#define RK30_L2C_SIZE     SZ_16K
#define RK30_SCU_PHYS     0x1013c000
#define RK30_SCU_SIZE     SZ_256
#define RK30_GICC_PHYS    0x1013c100
#define RK30_GICC_SIZE    SZ_256
#define RK30_GTIMER_PHYS  0x1013c200
#define RK30_GTIMER_SIZE  SZ_1K
#define RK30_PTIMER_PHYS  0x1013c600
#define RK30_PTIMER_SIZE  (SZ_2K + SZ_512)
#define RK30_GICD_PHYS    0x1013d000
#define RK30_GICD_SIZE    SZ_2K
```

LOG: WIR KOMMEN NOCH NICHT WEIT

[...]

Boot ver: 2013-05-18#1.20

start_linux=====78620

508144 Starting kernel...@0x60408000

Uncompressing Linux... done, booting the kernel.

Error: unrecognized/unsupported machine ID (r1 = 0x00000bfa).

Available machine support:

ID (hex) NAME

ffffffff Generic DT based system

Please check your kernel config and/or bootloader.

DEVICETREE: WAS IST DAS?

- Datenstruktur um Hardware zu beschreiben
- ursprünglich aus dem PowerPC-Bereich (OpenFirmware)
- heute heißt standalone als "Flattened Devicetree"
- unabhängig von der Implementierung (Linux, BSDs, ...)
- Binding-Beschreibung in `Documentation/devicetree/bindings/*`
- Devicetree als ABI, neue Kernel müssen alte Devicetrees unterstützen
- Einführung auf www.devicetree.org
- Spezifikation im "Standard for embedded power architecture platform requirements (ePAPR)"

DEVICETREE: GRUNDGERÜST

```
/dts-v1/;
/ {
    model = "Firefly-RK3288";
    compatible = "firefly,firefly-rk3288", "rockchip,rk3288";
    memory {
        reg = <0 0x80000000>;
    };
    cpus {
        #address-cells = <1>;
        #size-cells = <0>;
        cpu0: cpu@500 {
            device_type = "cpu";
            compatible = "arm,cortex-a12";
            reg = <0x500>;
        };
        cpu@501 {
            device_type = "cpu";
            compatible = "arm,cortex-a12";
            reg = <0x501>;
        };
        [...]
    };
};
```

DEVICETREE + BOOT: STOLPERFALLEN

- Herstellerbootloader können oft nicht mit DT umgehen
- Alternativ: Hersteller hat sich eigenes DT-Schema ausgedacht
- DT an Kernel anhängen `CONFIG_ARM_APPENDED_DTB`
- Bootloader kann unnütze Kernel-Commandline übergeben
- Abhilfe durch `CONFIG_CMDLINE` und `CONFIG_CMDLINE_FORCE`

LOG: ZWEI SCHRITTE WEITER

```
Uncompressing Linux... done, booting the kernel.
Booting Linux on physical CPU 0x500
Linux version 4.0.0-rc2-next-20150306+ [...]
CPU: ARMv7 Processor [410fc0d1] revision 1 (ARMv7), cr=10c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine model: Firefly-RK3288
bootconsole [earlycon0] enabled
Memory policy: Data cache writealloc
PERCPU: Embedded 11 pages/cpu @ee5ac000 s13376 r8192 d23488 u45056
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 522578
Kernel command line: console=fb0 console=ttyS2,115200 earlyprintk [...]
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
Memory: 2062296K/2097152K available (4914K kernel code, 235K rwdata, [...])
Virtual kernel memory layout:
   vector   : 0xffff0000 - 0xffff1000   (   4 kB)
   fixmap   : 0xffc00000 - 0xfff00000   (3072 kB)
   vmalloc  : 0xf0000000 - 0xff000000   ( 240 MB)
   lowmem   : 0xc0000000 - 0xef800000   ( 760 MB)
[...]
clocksource_of_init: no matching clocksources found
sched_clock: 32 bits at 100 Hz, resolution 10000000ns, wraps every 214748[...]
]
Console: colour dummy device 80x30
Calibrating delay loop...
```

INTERRUPTS

- Unterbrechung des Prozessors um Hardware-Events zu verarbeiten
- ARM9: meist Eigenbau-Interrupt-Controller
- ARM Generic Interrupt Controller (GIC)
- Ausnahme: Raspberry Pi2 kein GIC obwohl Cortex-A7

TIMER: HERDING CATS [*]

[*] tglx in »[GIT pull] timer changes for 3.18«:

- A new ARM SoC timer abomination. One should expect that we have enough of them already, but they insist on inventing new ones.

- The usual bunch of ARM SoC timer updates. That feels like herding cats.

- Clocksource - einfacher Zähler
- Clockevents - Interrupt nach x Zählerdurchläufen
- ARM9, Cortex-A8: nur soc-spezifische Timer
- Cortex-A9: Global- und Local-Timer
- Cortex-A15 und später: Architected Timer
- trotz generischer Timer auch immer SoC-spezifische

TIMER + INTERRUPTS: DEVICETREE

```
/ {
    [...]
    interrupt-parent = <&gic>;
    timer {
        compatible = "arm,armv7-timer";
        interrupts = <GIC_PPI 13 (GIC_CPU_MASK_SIMPLE(4) | IRQ_TYPE_LEVEL_HIGH)>,
                    <GIC_PPI 14 (GIC_CPU_MASK_SIMPLE(4) | IRQ_TYPE_LEVEL_HIGH)>,
                    <GIC_PPI 11 (GIC_CPU_MASK_SIMPLE(4) | IRQ_TYPE_LEVEL_HIGH)>,
                    <GIC_PPI 10 (GIC_CPU_MASK_SIMPLE(4) | IRQ_TYPE_LEVEL_HIGH)>;
        clock-frequency = <24000000>;
    };

    gic: interrupt-controller@ffc01000 {
        compatible = "arm,gic-400";
        interrupt-controller;
        #interrupt-cells = <3>;
        #address-cells = <0>;
        reg = <0xffc01000 0x1000>,
              <0xffc02000 0x1000>,
              <0xffc04000 0x2000>,
              <0xffc06000 0x2000>;
        interrupts = <GIC_PPI 9 0xf04>;
    };
};
```


LOG: KEINE KONSOLE

```
.data : 0xc06b4000 - 0xc06eccc8 ( 236 kB)
.bss : 0xc06eccc8 - 0xc0ef5f20 (8221 kB)
Architected cp15 timer(s) running at 24.00MHz (phys).
sched_clock: 56 bits at 24MHz, resolution 41ns, wraps every 2863311519744ns
Switching to timer-based delay loop, resolution 41ns
Console: colour dummy device 80x30
Calibrating delay loop (skipped), value calculated using timer frequency.. 48
.00 BogoMIPS (lpj=240000)
CPU: Testing write buffer coherency: ok
CPU0: thread -1, cpu 0, socket 5, mpidr 80000500
CPU1: failed to boot: -6
CPU2: failed to boot: -6
CPU3: failed to boot: -6
Brought up 1 CPUs
SMP: Total of 1 processors activated (48.00 BogoMIPS).
CPU: All CPU(s) started in SVC mode.
[...]
Switched to clocksource arch_sys_counter
[...]
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
usbcore: registered new interface driver r8188eu
usbcore: registered new interface driver rtl8723au
NET: Registered protocol family 17
Registering SWP/SWPB emulation handler
bootconsole [earlycon0] disabled
```

SYSCON: REGISTER-SAMMELSURIUM

- Ermöglicht Zugriff auf System Controller Register
- Registerblöcke mit Einstellungen zu vielen grundverschiedenen Komponenten, die sich nicht als einzelner Treiber darstellen lassen
- Rockchip: General Register Files mit 230 32-bit Registern zu IOMUX Einstellungen, Ethernet, HDMI, eDP, PLL Lock Status, USB Phy uvm.
- syscon sichert den parallelen Zugriff auf diese Register und ermöglicht syscons im Devicetree zu referenzieren
- Aber: keine Allzweckwaffe, d.h. Einsatz sollte wohlüberlegt sein

SYSCON: RK3288

```
/ {  
    [...]  
    grf: syscon@fff770000 {  
        compatible = "rockchip,rk3288-grf", "syscon";  
        reg = <0xff770000 0x1000>;  
    };  
  
    pmu: power-management@ff730000 {  
        compatible = "rockchip,rk3288-pmu", "syscon";  
        reg = <0xff730000 0x100>;  
    };  
  
    sgrf: syscon@ff740000 {  
        compatible = "rockchip,rk3288-sgrf", "syscon";  
        reg = <0xff740000 0x1000>;  
    };  
  
    /* Beispiel */  
    hdmi: hdmi@ff980000 {  
        [...]  
        rockchip,grf = <&grf>;  
    };  
};
```

PINCTRL: IO-MUXING UND GPIOS

- Datenpins der Prozessoren meist mehrfach belegt
- Pinmux legt fest zu welchem Controllerblock Daten geleitet werden
- aber auch: oft können nicht alle Peripherieblöcke gleichzeitig genutzt werden
- GPIO - general purpose Input/Output, d.h. low/high Status lesen (Input) oder setzen (Output)
- Input GPIOs können Interrupts generieren - z.B. Buttons oder Card-Detect

COMMON-CLOCK-FRAMEWORK: IM TAKT

- externer Oscillator (24MHz)
- Phase-Locked-Loops (PLLs)
- Multiplexer und Teiler
- Gates
- Clock-Struktur extrem gewachsen seit ARM9
- RK3188 clocks: Registermap aus 150kb Code gebastelt

CLOCKS: DEVICETREE

```
/ {  
    [...]  
    cru: clock-controller@ff760000 {  
        compatible = "rockchip,rk3288-cru";  
        reg = <0xff760000 0x1000>;  
        rockchip,grf = <&grf>;  
        #clock-cells = <1>;  
        #reset-cells = <1>;  
        assigned-clocks = <&cru PLL_GPLL>, <&cru PLL_CPLL>,  
            <&cru ACLK_CPU>, <&cru HCLK_CPU>,  
            <&cru PCLK_CPU>, <&cru ACLK_PERI>,  
            <&cru HCLK_PERI>, <&cru PCLK_PERI>;  
        assigned-clock-rates = <594000000>, <400000000>,  
            <300000000>, <150000000>,  
            <750000000>, <300000000>,  
            <150000000>, <750000000>;  
    };  
};
```

SERIELLER TREIBER: VERBINDUNG ZUR AUSSENWELT

- DebugLL wird während des Boots beendet
- mit Glück heutzutage meist 8250 oder PL011 basiert
- neuer UART-Treiber schwierig, ggf. möglich den Treiber aus dem Vendor-Kernel aufzuräumen
- temporäre Alternative: `keep_bootcon` cmdline Option

SERIELLER TREIBER: DEVICETREE

```
/ {  
    [...]  
  
    uart2: serial@fff690000 {  
        compatible = "rockchip,rk3288-uart", "snps,dw-apb-uart";  
        reg = <0xff690000 0x100>;  
        interrupts = <GIC_SPI 57 IRQ_TYPE_LEVEL_HIGH>;  
  
        reg-shift = <2>;  
        reg-io-width = <4>;  
  
        clocks = <&cru SCLK_UART2>, <&cru PCLK_UART2>;  
        clock-names = "baudclk", "apb_pclk";  
        pinctrl-names = "default";  
        pinctrl-0 = <&uart2_xfer>;  
    };  
};
```

LOG: KEIN ROOT-DATEISYSTEM

```
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
ff180000.serial: ttyS0 at MMIO 0xff180000 (irq = 30, base_baud = 1500000)
ff190000.serial: ttyS1 at MMIO 0xff190000 (irq = 31, base_baud = 1500000)
console [ttyS2] disabled
ff690000.serial: ttyS2 at MMIO 0xff690000 (irq = 32, base_baud = 1500000)
console [ttyS2] enabled
console [ttyS2] enabled
bootconsole [earlycon0] disabled
bootconsole [earlycon0] disabled
[...]
VFS: Cannot open root device "(null)" or unknown-block(0,0): error -6
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0
)
CPU: 0 PID: 1 Comm: swapper/0 Not tainted 4.0.0-rc3-next-20150311+ #13
Hardware name: Rockchip (Device Tree)
[<c0014e38>] (unwind_backtrace) from [<c00116ec>] (show_stack+0x10/0x14)
[<c00116ec>] (show_stack) from [<c04a21d4>] (dump_stack+0x84/0xb4)
[<c04a21d4>] (dump_stack) from [<c049f98c>] (panic+0x88/0x1f0)
[<c049f98c>] (panic) from [<c067c298>] (mount_block_root+0x238/0x284)
[<c067c298>] (mount_block_root) from [<c067c588>] (prepare_namespace+[...])
[<c067c588>] (prepare_namespace) from [<c067bf08>] (kernel_init_freeable+[...])
[<c067bf08>] (kernel_init_freeable) from [<c049e514>] (kernel_init+0x8/0xe4)
[<c049e514>] (kernel_init) from [<c000eab0>] (ret_from_fork+0x14/0x24)
---[ end Kernel panic - not syncing: VFS: Unable to mount root fs on unkno[.
.]
```

INITRAMFS

- auf Busybox-Basis in wenigen MB
- z.B. durch Entpacken des Debian busybox-static Paketes
- Busybox bringt alle wesentlichen Tools mit
- ermöglicht es erste Tests und schnelle Bootzyklen
- kann mittels Kconfig INITRAMFS_SOURCE an den Kernel angehängt werden

INITRAMFS: INITSKRIPT

- busybox --install -s legt symlinks für sh, tar, etc an
- mdev arbeitet ähnlich udev und legt Geräte an

```
#!/bin/sh

mount proc /proc -t proc
mount sysfs /sys -t sysfs
mount none /debug -t debugfs

#Create all the symlinks to /bin/busybox
/bin/busybox --install -s

#Create some device nodes
mknod /dev/null c 1 3
mknod /dev/tty c 5 0
mdev -s

/bin/sh
```


MASSENSPEICHER

- NAND - oftmals binärer Blob wegen patentbelasteter FTLs
- MMC/EMMC (z.B. Designware dw_mmc bei Rockchip, Exynos und anderen)
- USB (Rockchip: Designware dwc2)

DMA

- ermöglicht Peripheriekomponenten direkten Speicherzugriff
- ARM9 meist spezielle DMA-Controller (z.B. S3C24xx)
- ARM PL080
- ARM PL330
- in aktuellen Prozessoren nur noch PL330

I2C UND PMIC

- I2C: Master-Slave-Bus um mit anderen Chips zu kommunizieren
- z.B. PMIC, Audio-Codec, Sensoren, Touchscreen-Controller
- gelegentlich auch zur Versorgung eines VGA-Anschlusses (DDC)
- PMIC: in modernen Systemen große Anzahl an unterschiedlichen individuell kontrollierbaren Spannungen nötig
- Regulator-Framework sorgt dafür, dass nur die nötigen Spannungen aktiviert sind

SYMMETRIC MULTIPROCESSING (SMP)

- CPU-Kern 2-x aktivieren und deaktivieren (hotplug)
- Wunschenken: PSCI (Power State Coordination Interface)
- Realität: Prozessor-spezifische Methoden um Cores an- und auszuschalten
- Rockchip: wenn Core Powerdomain aktiviert wird, führt der Core den Code aus, der an einer bestimmten Adresse liegt (entweder im SRAM oder Bootrom)

SYSTEM-SUSPEND

- bringt System in einen Schlaf-Zustand um den Energieverbrauch zu minimieren
- kann durch Interrupts wieder aufgeweckt werden (wake-up sources)
- Geräte-Treiber betreiben ihre eigenen Suspend-Aktionen (pm_ops)
- System-Teil sehr Prozessorabhängig (RAM in self refresh, etc)
- Rockchip: RK3288 kann self-refresh selber auslösen, ältere benötigen SRAM Code

GRAFIKSTACK

- fbdev (deprecated, aber weit verbreitet auf Android)
- Direct Rendering Manager (drm, kms) als Standard-Schnittstelle
- ADF (Atomic display framework, Android Neuentwicklung, Warum?)

GRAFIKSTACK: DRM

- Konzept einer "Grafikkarte" (2D, 3D, Ausgabekomponenten in einem) passt nicht auf embedded SoCs
- CRTC - Auflösung, Refreshrate, was wird angezeigt
- Encoder - analoges Ausgabesignal generieren
- Connector - physischer Port, Verbindungsstatus, verfügbare Modi
- Component-Framework ermöglicht Initialisierung aus separaten Komponenten

GRAFIKSTACK: COMPONENT-FRAMEWORK

- Treiber-Probe legt nur Komponenten-Hierarchie an
- Einzelkomponenten werden vom Master gebunden, wenn alle Komponenten da sind
- Beschreibung der Verbindungen zwischen Komponenten durch OF-Graph Elemente im Devicetree

GRAFIKSTACK: KOMPONENTEN

```
display-subsystem {
    compatible = "rockchip,display-subsystem";
    ports = <&vopl_out>, <&vopb_out>;
};
vopl: vop@ff940000 {
    compatible = "rockchip,rk3288-vop";
    vopl_out: port {
        vopl_out_hdmi: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&hdmi_in_vopl>;
        };
    };
};
hdmi: hdmi@ff980000 {
    compatible = "rockchip,rk3288-dw-hdmi";
    ports {
        hdmi_in: port {
            hdmi_in_vopb: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&vopb_out_hdmi>;
            };
        };
    };
};
```

GRAFIKSTACK: GPU VS. LCD-CONTROLLER

- LCDC kontrolliert Planes + Auflösung + HDMI, LVDS und andere Ausgabe-Blöcke
- Mali / PowerVR sind RenderNodes, d.h. Rendern in Speicherbereiche
- Qualcomm Adreno, NVidia Tegra können integrierte Treiber verwenden

GRAFIKSTACK: 3D-BESCHLEUNIGUNG

- Binary-Treiber weitverbreitet
- offene Treiber unterschiedlich weit fortgeschritten
- MALI - Lima/Tamil
- PowerVR - ???
- Vivante - Etnaviv
- Adreno - sogar Qualcomm scheint auf Freedreno umzuschwenken
- Tegra - K1 verwendet Nouveau, Patches von NVidia selbst

FRAGEN?