

„Linuxsensorik“

„Wie wir Beschleunigungssensoren auslesen“

Ein Vortrag im Rahmen von nichts

„Wie wir Beschleunigungssensoren auslesen“

Der DTB ...

„Wie wir Beschleunigungssensoren auslesen“

Der DTB ...

..nein, nicht der Deutsche Turnerbund!

„Wie wir Beschleunigungssensoren auslesen“

Device Tree im Binaryformat

Der Device Tree ist eine Zusammenstellung von Angaben zur Hardware eines Endgerätes.

„Wie wir Beschleunigungssensoren auslesen“

Device Tree im Binaryformat

Der Device Tree ist eine Zusammenstellung von Angaben zur Hardware eines Endgerätes.

Unter Linux wird dies für nicht-findbare Hardwarekomponenten benutzt, wie sie zum Beispiel auf dem SOC eines Handies zu finden sind.

Geräte die z.B. per PCI- oder USB-Bus gefunden werden können, brauchen keine Einträge im Device Tree.

„Wie wir Beschleunigungssensoren auslesen“

Device Tree im Binaryformat

Der Device Tree ist eine Zusammenstellung von Angaben zur Hardware eines Endgerätes.

Unter Linux wird dies für nicht-findbare Hardwarekomponenten benutzt, wie sie zum Beispiel auf dem SOC eines Handies zu finden sind.

Geräte die z.B. per PCI- oder USB-Bus gefunden werden können, brauchen keine Einträge im Device Tree.

Menschen beschreiben den Device Tree in einer „`dtsti`“ – Datei in lesbarer Form.

Die finale Version für das Endgerät wird in Binärformat `.dtb` ausgeliefert, welches durch Kompilieren der `.dtsti` – Datei erzeugt wird.

„Wie wir Beschleunigungssensoren auslesen“

Beispiel: **sun50i** Chipsatz für einen Allwinner 64 SOC

```
aliases {  
    ethernet0 = &rtl8723cs;  
    serial0 = &uart0;  
};
```

```
backlight: backlight {  
    compatible = "pwm-backlight";  
    pwms = <&r_pwm 0 50000 PWM_POLARITY_INVERTED>;  
    enable-gpios = <&pio 7 10 GPIO_ACTIVE_HIGH>; /* PH10 */  
    power-supply = <&reg_ps>;  
    /* Backlight configuration differs per PinePhone revision. */  
};
```

Quelle: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/arch/arm64/boot/dts/allwinner/sun50i-a64-pinephone.dtsi?h=v5.12-rc2>

„Wie wir Beschleunigungssensoren auslesen“

Beispiel: **sun50i** Chipsatz für einen Allwinner 64 SOC

Natürlich ist dies nur ein kleiner Ausschnitt einer kompletten DTSI Datei.

Wir wissen aber jetzt, daß dieser **SOC** (**S**ystem **o**n a **C**hip) einen **Realtek 8723cs Chip** benutzt.

Mit dieser Information kann der Kernel dann diesen Treiber laden und das Ethernet, hier in Form einer WLAN Anbindung, bereitstellen.

„Wie wir Beschleunigungssensoren auslesen“

Beispiel: **sun50i** Chipsatz für einen Allwinner 64 SOC

Im weiteren Verlauf des DT's bekommen wir mitgeteilt, daß dieser SOC einen **Beschleunigungssensor** hat.

```
&i2c1 {  
    status = "okay";  
    /* Accelerometer/gyroscope */  
    accelerometer@68 {  
        compatible = "invensense,mpu6050";  
        reg = <0x68>;  
        interrupt-parent = <&pio>;  
        interrupts = <7 5 IRQ_TYPE_EDGE_RISING>; /* PH5 */  
        vdd-supply = <&reg_dldo1>;  
        vddio-supply = <&reg_dldo1>;  
    };  
};
```

„Wie wir Beschleunigungssensoren auslesen“

Wo finden wir denn jetzt diesen MPU6050 Sensor im System?

„Wie wir Beschleunigungssensoren auslesen“

Da es sich bei dem Chip um einen per I2C Bus angebundenes Gerät handelt, suchen wir doch am besten im I2C Interface:

```
/sys/bus/iio/devices/
```

„Wie wir Beschleunigungssensoren auslesen“

Auf einem PinePhone A64 SOC bekommen hier z.B. diese Geräte angezeigt:

```
lrwxrwxrwx 1 root root 0 16. Mär 15:39 iio:device0 -> ../../../../devices/platform/soc/1f03400.rsb/sunxi-rsb-3a3/axp813-adc/iio:device0
lrwxrwxrwx 1 root root 0 16. Mär 15:39 iio:device1 -> ../../../../devices/platform/soc/1c2b000.i2c/i2c-2/2-0048/iio:device1
lrwxrwxrwx 1 root root 0 16. Mär 15:39 iio:device2 -> ../../../../devices/platform/soc/1c2b000.i2c/i2c-2/2-0068/iio:device2
lrwxrwxrwx 1 root root 0 16. Mär 15:39 iio:device3 -> ../../../../devices/platform/soc/1c2b000.i2c/i2c-2/2-001e/iio:device3
lrwxrwxrwx 1 root root 0 16. Mär 15:39 trigger0 -> ../../../../devices/platform/soc/1c2b000.i2c/i2c-2/2-001e/trigger0
lrwxrwxrwx 1 root root 0 16. Mär 15:39 trigger1 -> ../../../../devices/platform/soc/1c2b000.i2c/i2c-2/2-0068/trigger1
```

iio:device2 ist hier das gesuchte Magnetometer. Wenn man das von Hand bestimmen möchte, muß man einfach nur mal reinsehen ;)

„Wie wir Beschleunigungssensoren auslesen“

Die Sensorwerte

in_accel_x_raw	X-Achsenwert	-17000 – 17000
in_accel_y_raw	Y-Achsenwert	-17000 – 17000
in_accel_z_raw	Z-Achsenwert	-17000 – 17000

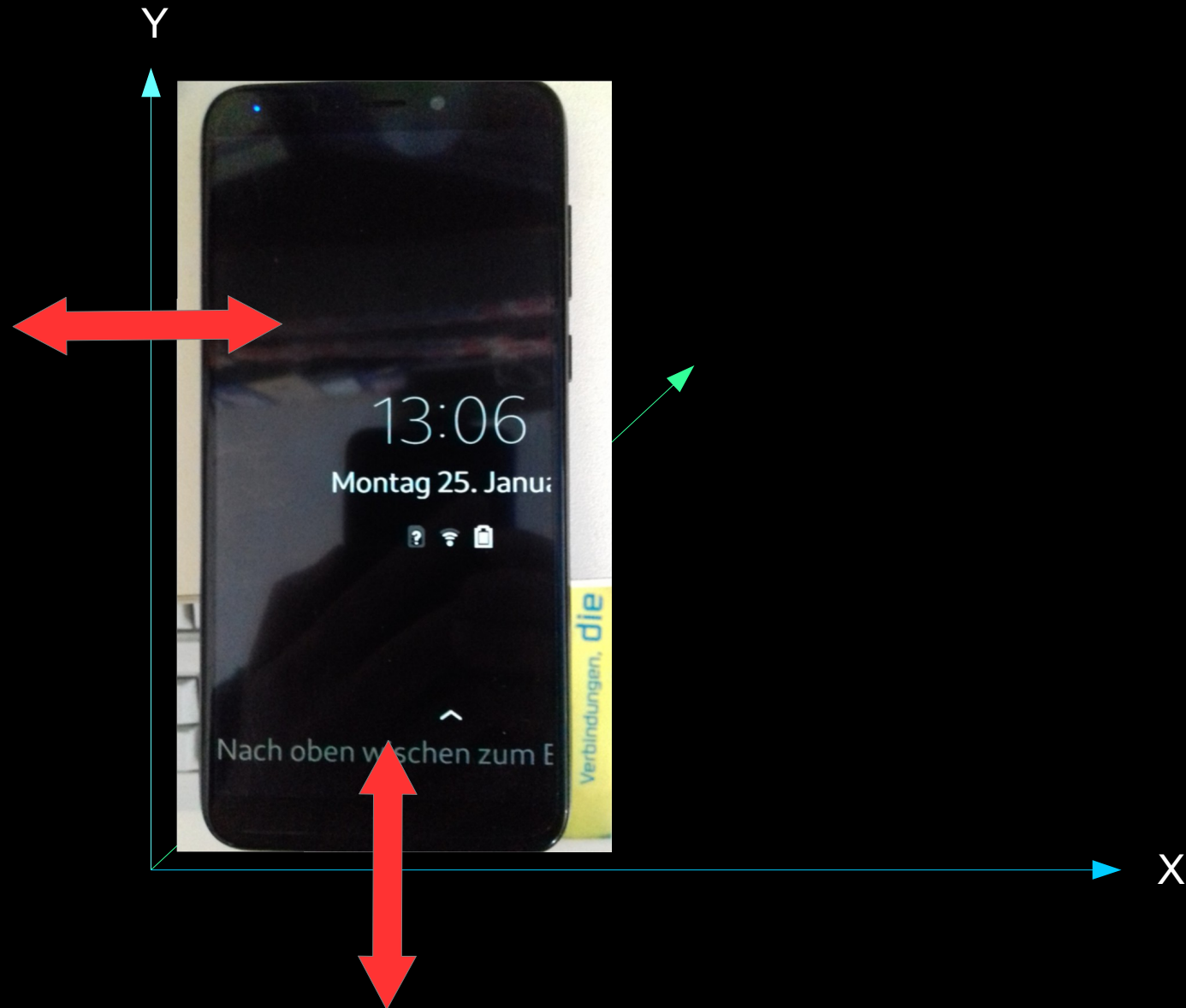
Beispiele:

- X > 15000 Handy steht Senkrecht mit Kamera oben
- X < -15000 Handy steht Senkrecht mit Kamera unten
- Y > 15000 Handy liegt auf rechter Kante
- Y < -15000 Handy liegt auf linker Kante
- Z > 15000 Handy liegt auf der Rückseite
- Z < -15000 Handy liegt auf dem Display

„Wie wir Beschleunigungssensoren auslesen“

Ein Beschleunigungssensor misst den „Zug“ an der z -Achse.

„Wie wir Beschleunigungssensoren auslesen“



„Wie wir Beschleunigungssensoren auslesen“

Das Auslesen der Daten ist der einfachste Teil:

„cat in_accel_x_raw“

„Wie wir Beschleunigungssensoren auslesen“

Wie nutzt man diese Infos jetzt sinnvoll?

„Wie wir Beschleunigungssensoren auslesen“

Bildschirmrotation \o/

„Wie wir Beschleunigungssensoren auslesen“

```
OLD=""
while :
do
  X=$(cat /sys/bus/iio/devices/iio:device2/in_accel_x_raw)
  Y=$(cat /sys/bus/iio/devices/iio:device2/in_accel_y_raw)
  Z=$(cat /sys/bus/iio/devices/iio:device2/in_accel_z_raw)

  if [ $X -gt 15000 ] && [ "$OLD" != "N" ]; then
    # portray mode 0 Grad
    wlr-randr --output DSI-1 --transform normal
    OLD="N"
  fi
  if [ $Y -gt 15000 ] && [ "$OLD" != "90" ]; then
    #Landscape 90 Grad
    wlr-randr --output DSI-1 --transform 90
    OLD="90"
  fi
  if [ $Y -lt -15000 ] && [ "$OLD" != "270" ]; then
    #Landscape 270 Grad
    wlr-randr --output DSI-1 --transform 270
    OLD="270"
  fi
  sleep 0.5
done
```

„Wie wir Beschleunigungssensoren auslesen“

Zunächst lesen wir in einer Endlosschleife X, Y, Z aus. Z braucht wir hier nicht, aber auch damit ließen sich nette Sachen umsetzen.

```
X=$(cat /sys/bus/iio/devices/iio:device2/in_accel_x_raw)
Y=$(cat /sys/bus/iio/devices/iio:device2/in_accel_y_raw)
Z=$(cat /sys/bus/iio/devices/iio:device2/in_accel_z_raw)
```

Am Beispiel Portraymode setzen wir die Logik um:

```
if [ $X -gt 15000 ] && [ "$OLD" != "N" ]; then
    # portray mode 0 Grad
    wlr-randr --output DSI-1 --transform normal
    OLD="N"
fi
```

Wenn $X > 15000$ ist, steht das Handy auf der Ladebuchse, also senkrecht. Das bedeutet, wir drehen den Screen auf Normalposition. Die OLD Variable verhindert, daß wir alle 0.5 Sekunden den Bildschirm drehen, auch wenn sich das Handy gar nicht bewegt.

`wlr-randr` führt für uns die eigentliche Bildschirmdrehung des Wayland-Bildschirms durch. Nach dem Drehen merken wir uns in OLD, in welcher Position der Bildschirm nun ist, den in diese Position brauchen wir ihn nicht nochmal bringen ;)

„Wie wir Beschleunigungssensoren auslesen“

Mehr zur technischen Umsetzung

<https://marius.bloggt-in-braunschweig.de/2021/03/14/pinephone-automatische-screenrotation-einschalten/>