Hochschule Furtwangen
Fakultät Informatik
Prof. Dr. Lothar Piepmeyer

Workshop ‚Plattformen für mobile Systeme'

Aufgabenblatt 8

Im Rahmen dieser Aufgaben konstruieren Sie Kernel-Module für Ihr Target.

Aufgabe 1:

Melden Sie sich an Ihrem *Host*-System an. Das Target wird zunächst nicht benötigt. Prüfen Sie Ihr Environment. Stellen Sie *unbedingt* sicher, dass die Umgebungsvariablen

```
ARCH
CROSS_COMPILE
CC
```

richtig für das Cross-Development gesetzt sind! Weisen Sie *außerdem* der Umgebungsvariablen `KERNEL` den Wert `kernel7` zu.

Aufgabe 2:

a. Wechseln in das Verzeichnis ~/`mobile-project/modules` .

b. Rufen Sie `make` mit den richtigen Parametern auf. Insbesondere muss der Wert des Parameters –C auf das Verzeichnis mit den Sourcen Ihrer Linux-Kernels angepasst werden

c. Fügen Sie das Kernel-Modul ein und stellen Sie sicher, dass es erfolgreich eingefügt wurde.

d. Ordnen Sie dem Modul einen Geräteknoten zu.

e. Testen Sie das Gerät, indem Sie wie in der Vorlesung beschrieben auf das Character-Device schreiben und anschließend etwa mit `cat` vom Gerät lesen.

f. Entfernen Sie das Modul und stellen Sie sicher, dass es erfolgreich entfernt wurde.

Aufgabe 3:

Nutzen Sie `simple.c` als Vorlage für ein weiteres Kernel-Modul für ein Gerät, auf das man nur schreiben kann. Wenn dem Modul etwa der Gerätknoten `/dev/blinker` zugeordnet wurde, soll

```
echo –n 3 > /dev/blinker
```

bewirken, dass die Status-LED dreimal blinkt. Das soll analog für alle Werte zwischen 1 und 9 funktionieren.

Dazu müssen Sie noch folgendes wissen:

- Wenn der Standard-Trigger der Status-LED auf `none` gestellt ist - und so haben wir den Kernel ja extra gebaut - dann kann man die LED als GPIO 47 ansprechen.
- GPIO-Pins weist man Werte mit Hilfe der Funktion `gpio_set_value(pin, value)` zu. Der Prototyp der Funktion steht in `linux/gpio.h`.
- Im Kernel lässt man den ausführenden Thread mit Hilfe von `msleep(n)` n Millisekunden ruhen. Der Prototyp der Funktion steht in `linux/delay.h`.