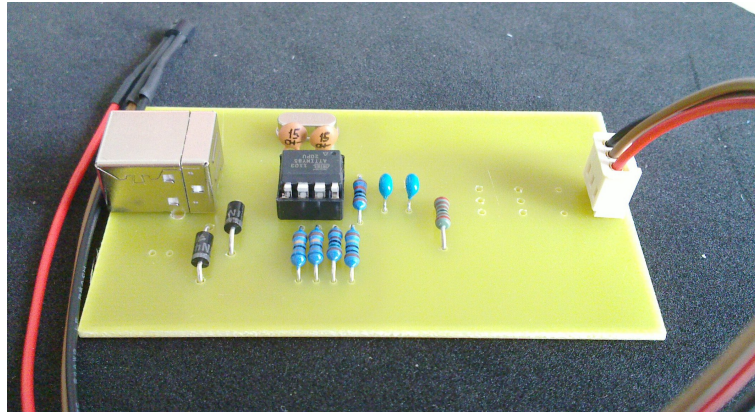


4 Übungsaufgabe #4: Entwicklung eines Linux-Kernel-Moduls

Im Rahmen von Aufgabe 4 soll für den im Seminar vorgestellten USB-Temperatursensor ein Linux-Gerätetreiber entwickelt werden. Dieser soll über Einträge im `Sysfs` die gemessene Temperatur der jeweiligen Sensoren ausgeben.



4.1 Grundlagen

Im Rahmen des Seminars wurden einige Informationsquellen vorgestellt, welche die für diese Aufgabe benötigte Funktionalität beschreiben. Über diese solltet ihr euch zuerst einen Überblick verschaffen, da die Entwicklung eines Treibers viele Teile des Kerns involviert.

Einen guten Ausgangspunkt zur Programmierung bietet das *Writing USB Device Drivers-How-To*, das man sukzessive um die benötigte Funktionalität erweitern kann.

4.2 Funktionsumfang des Treibermoduls

Der Treiber soll als Kernel-Modul implementiert werden. Es soll möglich sein mehrere angeschlossene Instanzen des Gerätes zu verwalten.

Im `Sysfs`-Baum eines angeschlossenen Gerätes soll für jeden am USB-Gerät vorhandenen Sensor eine eigene Datei angelegt werden, die beim Lesen den gemessenen Temperaturwert des jeweiligen Sensors liefern soll. Dabei bietet es sich an die ID des Sensors als Dateinamen zu verwenden.

Darüberhinaus gibt es die Befehle `rescan` bzw. `reset`. Diese sollen ebenfalls durch eine entsprechend benannte Datei im `Sysfs`-Baum repräsentiert werden. Durch Schreiben von 1 in die Gerätedatei soll der korrespondierende Befehl auf dem USB-Gerät ausgeführt werden.

Wird `reset` ausgeführt, so ist das aus Sicht der Software vergleichbar mit einem Entfernen und Wiederanschließen des USB-Gerätes. Bei `rescan` hingegen wird nur der 1-Wire-Bus neu initialisiert; das USB-Gerät bleibt dabei mit dem PC verbunden. Dessen Zustand kann sich jedoch zwischenzeitlich ändern: Es können Temperatursensoren verschwinden oder neu erscheinen. Dies soll sich entsprechend in der Struktur des `Sysfs`-Baums widerspiegeln.

4.3 Zu Beachten

Um mehrere Instanzen des USB-Gerätes gleichzeitig zu unterstützen müssen die pro Gerät benötigten Datenstrukturen dynamisch allokiert werden. Wird das Gerät vom PC entfernt, müssen die damit assoziierten Datenstrukturen auch wieder freigegeben werden, um ein Speicherleck im Linux-Kern zu vermeiden.

Aufgaben:

- Platine des USB-Gerätes bestücken
- Lesen der Dokumentation, um die für den Treiber notwendigen Teile des Linux-Kerns zu verstehen und benutzen zu können
- Treiber gemäß obiger Anforderungen implementieren

4.4 Abgabe: am 11.07.2011