

TECHNISCHE HOCHSCHULE MITTELHESSEN

# Gerätetreiber in Embedded Systems

---

Schriftliche Ausarbeitung

Mario Weber

26.06.2011

# 1. Inhalt

|       |  |    |
|-------|--|----|
| 1.    | Inhalt .....                                 | 1  |
| 2.    | Software-Aufbau von Embedded Systems .....   | 2  |
| 2.1   | Definition .....                             | 2  |
| 2.2   | Systemaufbaumodell .....                     | 2  |
| 2.3   | Einordnung in das Modell .....               | 3  |
| 3.    | Aufgaben von Gerätetreibern .....            | 4  |
| 4.    | Arten von Gerätetreibern .....               | 5  |
| 4.1   | Architekturspezifische Gerätetreiber .....   | 5  |
| 4.2   | Generische Gerätetreiber .....               | 5  |
| 5.    | Typische Funktionen von Gerätetreibern ..... | 6  |
| 5.1   | Startup und Shutdown .....                   | 6  |
| 5.2   | Enable und Disable .....                     | 6  |
| 5.3   | Acquire und Release .....                    | 6  |
| 5.4   | Read und Write .....                         | 7  |
| 5.5   | Install und Uninstall .....                  | 7  |
| 6.    | Zustände von Gerätetreibern .....            | 8  |
| 6.1   | Zustandsmodell .....                         | 9  |
| 6.1.1 | Inactive .....                               | 9  |
| 6.1.2 | Busy .....                                   | 9  |
| 6.1.3 | Finished .....                               | 9  |
| 7.    | Interner Aufbau .....                        | 10 |
| 9.    | Quellenangabe .....                          | 11 |

## **2. *Software-Aufbau von Embedded Systems***

### **2.1 *Definition***

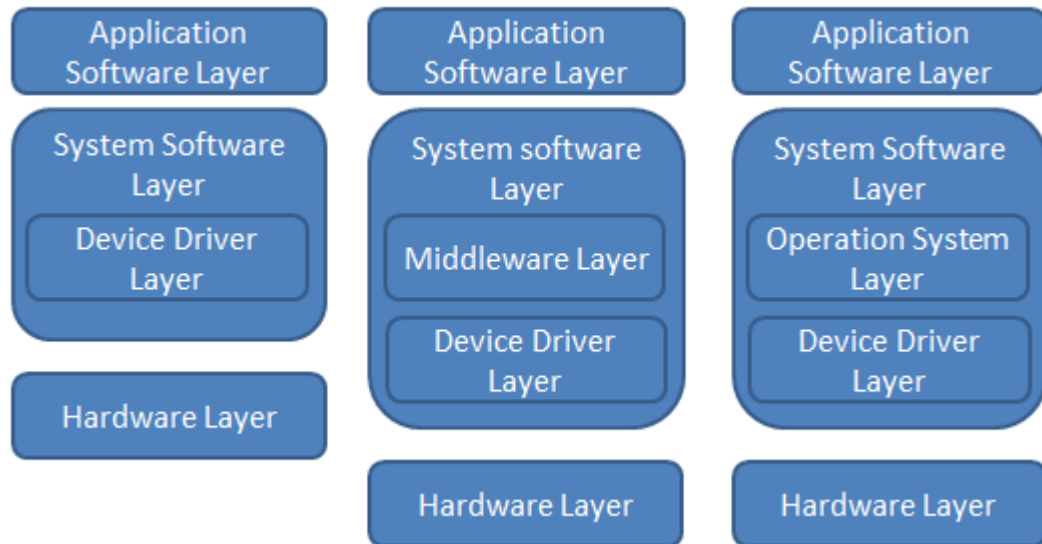
Die Hardwarekomponenten in Embedded Systems benötigen Software, die diese initialisiert und verwaltet. Die Software, die diese Aufgabe erfüllt, nennt man Gerätetreiber.

### **2.2 *Systemaufbaumodell***

Im untenstehenden Diagramm sieht man ein Modell, das den Aufbau eines Embedded Systems darstellt. Unterste Schicht ist die Hardwareerschicht. Darüber befindet sich die Systemsoftwareschicht. Welche Komponenten diese Schicht enthält unterscheidet sich im Einzelfall je nachdem wie aufwändig der Aufbau des konkreten Embedded Systems ist. Zu oberst befindet sich die Anwendungssoftwareschicht.

Die Gerätetreiber für Embedded Systems befinden sich in der Systemsoftwareschicht. Sie stellen die minimale Konfiguration dieser Schicht dar, das heißt sie sind in jedem Fall vorhanden, müssen aber nicht die einzige Komponente dieser Schicht sein.

In der linken Spalte sieht man einen Systemaufbau, bei dem die Systemsoftwareschicht lediglich aus der Gerätetreiberschicht besteht. In der mittleren Spalte wird der Aufbau der Systemsoftwareschicht um die Middlewareschicht ergänzt. In der rechten Spalte sieht man einen Systemaufbau, bei dem ein Betriebssystem vorhanden ist. Diese Betriebssystemschicht befindet sich ebenfalls in der Systemsoftwareschicht.



### 2.3 *Einordnung in das Modell*

Die Gerätetreiberschicht stellt somit je nach konkretem Systemaufbau die Schnittstelle zwischen Anwendungsschicht und Hardwareschicht, Middlewareschicht und Hardwareschicht oder Betriebssystemschicht und Hardwareschicht dar.

### **3. *Aufgaben von Gerätetreibern***

Die Aufgaben von Gerätetreibern im Allgemeinen sind die Hardware zu initialisieren und eine Zugriffsmöglichkeit für Software höherer Schichten bereitzustellen. Welche Aufgaben dies im speziellen sind und wie diese Aufgaben erfüllt werden hängt natürlich davon ab, welche Art Hardwarekomponente der Gerätetreiber verwalten soll.

## **4. Arten von Gerätetreibern**

Gerätetreiber werden typischerweise in zwei Kategorien eingeteilt. Diese sind architektur-spezifische Gerätetreiber und generische Gerätetreiber.

### **4.1 Architekturspezifische Gerätetreiber**

Architekturspezifische Gerätetreiber verwalten die Hardwarekomponenten, die in den Hauptprozessor integriert sind. Beispiele für Komponenten, die architektur-spezifische Gerätetreiber benötigen sind on-chip memory, integrierte MMUs (memory management units) und Gleitkommahardware.

### **4.2 Generische Gerätetreiber**

Generische Gerätetreiber verwalten Hardware die sich auf dem Board befindet und nicht in den Hauptprozessor integriert ist. Innerhalb eines generischen Gerätetreibers befindet sich gewöhnlicherweise Code, der architektur-spezifisch ist. Dies liegt daran, dass alle Komponenten auf dem Board in irgendeiner Weise über den Hauptprozessor angesprochen werden müssen. Generische Treiber verwalten aber Hardware die sich auf dem Board befindet und somit unabhängig von dem Hauptprozessor ist. Dieser Anteil des Code ist architektur-unabhängig. Das heißt generische Gerätetreiber müssen auf die jeweilige Architektur angepasst werden, können dann aber die Hardwarekomponente auf dem Board für die sie geschrieben sind auf einer Vielzahl von verschiedenen Architekturen verwalten.

## **5. *Typische Funktionen von Gerätetreibern***

Unabhängig davon welche Art von Hardware ein Gerätetreiber verwaltet beinhaltet er im Allgemeinen alle oder eine Teilmenge der folgenden Funktionalitäten.

### **5.1 *Startup und Shutdown***

Die Startup-Funktion initialisiert die Hardware wenn der Strom eingeschaltet wird oder ein Reset durchgeführt wurde.

Die Shutdown-Funktion versetzt die Hardware in den Zustand in dem sie ausgeschaltet werden kann.

### **5.2 *Enable und Disable***

Die Enable-Funktion erlaubt anderer Software die Hardware on-the-fly anzuschalten.

Die Disable-Funktion erlaubt anderer Software die Hardware on-the-fly auszuschalten.

### **5.3 *Acquire und Release***

Die Acquire-Funktion erlaubt anderer Software exklusiven Zugriff auf die Hardware zu erlangen.

Mittels der Release-Funktion kann andere Software exklusiven Zugriff auf die Hardware wieder freigeben.

## **5.4      *Read und Write***

Die Read-Funktion erlaubt anderer Software Daten von der Hardware zu lesen.

Die Write-Funktion erlaubt anderer Software Daten auf die Hardware zu schreiben.

## **5.5      *Install und Uninstall***

Die Install-Funktion erlaubt anderer Software Hardware on-the-fly zu installieren.

Die Uninstall-Funktion erlaubt anderer Software Hardware on-the-fly zu deinstallieren.



## 6. *Zustände von Gerätetreibern*

Dass Gerätetreiber genau diese gemeinsamen Funktionen haben liegt daran, dass die Software davon ausgeht, dass sich das Verhalten der Hardware durch ein Zustandsmodell beschreiben lässt. Die Hardware befindet sich immer in einem der Zustände inaktiv, beschäftigt oder fertig. Ein Übergang von einem Zustand in einen anderen wird ausgelöst durch das Aufrufen einer der Funktionen des Gerätetreibers.

## **6.1 Zustandsmodell**

### **6.1.1 Inactive**

Hardware im Zustand inaktiv wird angesehen entweder als disconnected, ohne Strom oder abgeschaltet. Hardware die disconnected ist benötigt die Install-Funktion. Hardware die ohne Strom ist benötigt die Startup-Funktion zur Initialisierung. Und Hardware die abgeschaltet ist benötigt die Enable-Funktion um von anderer Software eingeschaltet zu werden.

Die Zustände beschäftigt und fertig sind aktive Zustände. In diesen beiden Zuständen werden die Counterparts der oben erwähnten Funktionen benötigt, und zwar Uninstall, Shutdown und Disable.

### **6.1.2 Busy**

Hardware im Zustand beschäftigt ist gerade dabei Daten zu verarbeiten. In diesem Zustand wird die Release-Funktion benötigt um die Hardware wieder freizugeben.

### **6.1.3 Finished**

Befindet sich die Hardware im Zustand fertig verarbeitet sie zu diesem Zeitpunkt keine Daten. In diesem Zustand ist es möglich die Hardware mit der Acquire-Funktion zu beanspruchen oder die Read- oder Write-Funktion zu benutzen.

## **7. *Interner Aufbau***

Die einzelnen Funktionen eines Gerätetreibers können zu größeren Funktionen mit mächtigerer Funktionalität zusammen gefasst sein. Die Funktionen des Treiber haben Code der direkt die Hardware anspricht und Code der die Schnittstelle zu Software höherer Schichten implementiert. Diese Schichten können im Code des Treibers sowohl vermischt sein als auch im Code selbst in Schichten aufgeteilt sein.

## 9. *Quellenangabe*

Embedded Systems Architecture, Tammy Noergaard

Designing Embedded Hardware, John Catsoulis

[http://en.wikipedia.org/wiki/Embedded\\_system](http://en.wikipedia.org/wiki/Embedded_system)

[http://microcontroller.com/wp/DeviceDrivers/device\\_drivers5.htm](http://microcontroller.com/wp/DeviceDrivers/device_drivers5.htm)

[http://en.wikibooks.org/wiki/Embedded\\_Systems](http://en.wikibooks.org/wiki/Embedded_Systems)