

Vorwort

Design is not just what it looks like and feels like. Design is how it works.

[Steve Jobs]

Wir haben eine Idee oder sogar schon eine SW-Lösung. Nun muss sie noch auf einer Hardware realisiert werden. Übliche Boards laufen in einer Hauptschleife ohne Betriebssystem oder sie nutzen Linux oder QNX. Wenn unsere Lösungen auf existierenden fertigen Geräten installiert werden sollen, könnten wir es auch mit Android oder iOS zu tun haben. Nun wollen wir aber nicht Betriebssysteme studieren und uns auch nicht in sieben verschiedene Frameworks einarbeiten, deren Speicher- und Performance-Merkmale undokumentiert sind, sondern wir möchten nur die erforderlichen Kenntnisse erwerben, um das Board und unsere Applikation inklusive der erforderlichen Treiber und Interrupt-Routinen zum Laufen zu bringen.

Auch wollen wir unsere Lösung nicht noch einmal schreiben, diesmal angepasst an ein beliebiges Fenster-Framework, welches davon ausgeht, dass die ganze Welt als Sammlung von Fenstern beschreibbar ist. Eigentlich wollen wir nur die vorliegende Hardware und die verfügbaren Bedienoberflächen auf großen Displays nutzen. Möglicherweise schreiben wir auch eine HMI in Java und binden sie zu unserer SW hinzu. Dann braucht es geschickte Übergänge zwischen den Welten und für die Java-Welt braucht es ein paar Hinweise, wie wir mit Speicher, Garbage-Collection und Performance umgehen sollten. Vielleicht sollten wir auch verschiedene SW-Komponenten auf verschiedene Cores des Prozessors verteilen, eventuell virtualisieren wir dafür auch Cores, um die Komponenten und ihre Konfiguration voneinander unabhängig zu realisieren. Alle diese Kenntnisse sollten aber so fundiert sein, dass wir nicht, wie in der Informatik-Gemeinde so oft der Fall, qualifizierten Aber-

glauben walten lassen müssen, welcher Knopf zu welcher Zeit zu drücken ist, damit „es läuft“ (das „Ein Huhn schlachten-Phänomen“¹).

Und wenn dann die Applikation das erste Mal läuft, kann es sein, dass wir einen viel zu langsamen Start konstatieren, ein Speicherleck oder ein Performance-Problem des Gesamtsystems erkennen. Schlimmer noch, nach einiger Zeit bleibt das System vollständig stehen, sogar der Prozessor wird kalt. Auch hier könnten Kapitel über übliche Probleme mit Speichermedien oder über das Tracen und Debuggen eines Systems hilfreich sein. Viel zu oft wird die Methode „mit der Kapuze über den Kopf und der Taschenlampe in der Hand in den dunklen Keller gehen und das Problem suchen“ angewendet, das 30 Jahre alte sogenannte „printf-Debugging“. Einige vorgestellte kleine Tools könnten helfen, Fehler zu finden oder auch neue Ideen für eigene Tools zu entwickeln.

Noch ein Wort zu den verwendeten Begriffen. Die Kombination aus englischen und deutschen Begriffen ist nicht immer schön, gelegentlich aber unvermeidlich. Die Bezeichnungen „eingebettete“ oder „technische“ Systeme geben nicht wirklich die Vorstellung wieder, die man von embedded Systemen hat, die sich durch knappe und effizient eingesetzte Ressourcen auszeichnen. Auch die Diskussionen um Bindestriche oder Groß-/Kleinschreibung lassen sich hier nicht widerspruchsfrei führen (man kann root-Verzeichnis wirklich nicht groß schreiben).

Schon möglich, dass einige der eingestreuten Geschichten zur Motivation nach Benzin riechen, das könnte an den letzten 20 Jahren Arbeit für Bosch, Blaupunkt, Harman/Becker, Johnson Controls und andere europäische Automobilhersteller liegen. Diese Geschichten sollen aber immer auch die Relevanz des Themas über die Anekdote und auch über den automotiven Bereich hinaus zeigen.

Ich hoffe, dies gelingt. Viel Spaß beim Lesen.

Karlsruhe, Januar 2012

Joachim Wietzke

¹ In einem malaysischen Kurzgeschichtenbuch las ich einmal über einen Eingeborenenstamm auf einer Insel, der noch nie mit der Zivilisation zu tun hatte und gerade unter einer großen Hungersnot litt. In letzter Verzweiflung schlachtete man als Opfergabe an die Götter ein Huhn. In der Folgenacht strandete ein vollbeladenes Schiff, das man plündern konnte. Wann immer in den späteren Jahren wieder eine Hungersnot aufkam, wurde ein Huhn geschlachtet, in der Hoffnung, dass das wieder funktionieren würde. Diese Vorgehensweise steht im Widerspruch zu embedded Vorgehensweisen.



<http://www.springer.com/978-3-642-23995-3>

Embedded Technologies

Vom Treiber bis zur Grafik-Anbindung

Wietzke, J.

2012, XXVI, 321 S. 114 Abb., 20 Abb. in Farbe.,

Hardcover

ISBN: 978-3-642-23995-3