

Einleitung

Es gibt in der derzeitigen Fachliteratur einige Ausführungen über Embedded-Systeme. Ihre Schwerpunkte liegen oft auf den dafür verwendeten Betriebssystemen oder auf Applikationen in kompakter, nicht vernetzter Umgebung. Beide Autoren haben in den letzten Jahren die Einführung von modernen SW-Strukturen ins Auto miterlebt und mitgestaltet.

Die Ausführungen beziehen sich deshalb im Folgenden beispielhaft auf vernetzte Systeme im Auto. Im Moment ist der europäische Standard dafür ein optischer Bus, MOST-Bus genannt (media oriented system transport), [MOST].

Die Ausführungen sind aber allgemein genug, um auch für andere Vernetzungen verwendbar zu sein. Das Buch ist in zwei Teile gegliedert.

Teil 1:

Welche Bausteine zu einem Framework gehören, welche Dinge festgelegt werden müssen und welche Anforderungen an ein Framework gestellt werden, wird in diesem ersten Teil des Buches besprochen.

Insbesondere der erste Teil lässt sich auf viele Domänen der Embedded Software anwenden.

Da es vielen, auch erfahrenen SW-Ingenieuren nicht bekannt ist, wie sich ihre Programme im Speicher auswirken, beginnt der Teil 1 mit einigen Grundlagen dazu. Anschließend werden Grundkonzepte und Grundbausteine von Embedded Systemen erörtert.

Ein Schwerpunkt der Diskussion ist dabei das Speicherverhalten und die Startperformance. Grundmechanismen der Betriebssysteme werden erläutert und ausgemessen, um Speicherverwendung, Kommunikation und Synchronisationen effizient zu gestalten.

Dabei wird darauf geachtet, dass Implementierungen unter Linux, QNX und Windows lauffähig sind. Dieser Teil ist sowohl an Studierende als auch an Embedded Experten gerichtet.

Teil 2:

Im zweiten Teil des Buches wird eine prototypische objektorientierte Framework-Implementierung besprochen. Diese diente zur Erprobung besonders effizienter Synchronisations- und Kommunikationsansätze. Außerdem sollte das Framework besonders klein sein und gute Startzeiten haben. Die Vorsätze sind in dem kleinen Rahmen recht gut geglückt, ein Framework entsprechend der vorgestellten Erkenntnisse mit mittlerer Applikationsausstattung bewegte sich im Rahmen von 2,5 s Startzeit und einer Größe von einigen 100 kByte. Noch viel wichtiger betrug die Umsetzungszeit nur wenige Mannmonate.

Neue Ideen

In dieses Buch sind einige Konzepte eingeflossen, die den Autoren aus der Literatur noch nicht bekannt waren. Es sind dies:

- Flexible, aber einheitliche Konfiguration für Threads und Prozesse,
- dafür intensive Nutzung des Shared-Memories in Verbindung mit dem `fork`-Konstrukt und von Thread-specific-Data;
- Queue-Konstrukte mit Vorrichtungen gegen Prioritäts-Inversion;
- Komponenten-Klassen mit festem Set von Queues und Services;
- Hintergrundsysteme: Watchdog, Timer;
- spezielle FSM-Architekturen für Embedded Systeme;
- Lauffähigkeit auf QNX, Linux, Windows;
- const-Objekte im Code-Segment zum Dekodieren der MOST-Nachrichten zur Performance-Optimierung.

Eine Schwierigkeit bei der Erstellung war das Ziel, im Text verallgemeinert genug zu bleiben, um noch lesbar zu sein und doch gleichzeitig auch Quellcodes zu zeigen, damit sich der Leser ein substanzielles Bild machen kann, wie die Erklärungen gemeint sind und umgesetzt werden könnten.

Ein Versuch, dieses Dilemma zu lösen, war, die Quelltexte auf das Nötigste zu kürzen. Sämtliche Includes wurden weggelassen. Nach dem Vorbild von [Stev] wurden alle wichtigen Includes in einen Abschnitt im Anhang verlagert. Weiterhin wurden viele Programmzeilen durch „...“ ersetzt.

Alle Sourcen laufen prinzipiell auf QNX, Linux und Windows. Allerdings wurden auch hier einige `ifdefs` herausgekürzt, die nötig waren, um auf die Eigenheiten der Betriebssysteme einzugehen. Im Anhang finden sich dafür Hinweise.

Ohne dass ein ganzes Framework in allen Einzelheiten vorgestellt wird, kann der Leser sicher manche Erkenntnisse aus den Beispielen gewinnen, sie umsetzen oder eigene Implementierungen an den vorgestellten Implementierungen messen. Unter dem Link

www.fbi.fh-darmstadt.de/~jwietzke/Das.Buch

können viele Quellen angesehen und heruntergeladen werden. Auch ein aktueller Stand eines studentischen Framework-Projektes wird dort ab Juli 2005 zu finden sein.

Anmerkungen zur Domäne und zur Sprache:

Ein Framework ist immer domänenspezifisch, auch viele SW-Muster (Pattern) sind domänenspezifisch, in unseren Beispielen spezifisch für Embedded Automotive Systeme. In jeder Domäne gibt es Fachbegriffe, die sich nur schwer übersetzen lassen.

Wir werden auf die Eindeutschung von englischen Begriffen verzichten, ein Thread wird nicht als Laufzeitfaden übersetzt, sondern bleibt, nach einer inhaltlichen Erläuterung, ein Thread. Auch die Frage der Groß- oder Kleinschreibung englischer Begriffe, die eventuell im Deutschen schon bekannt sind, wird systematisch ignoriert. Wir wählen, wann immer wir darauf achten, die Großschreibung oder die Kleinschreibung.

Auch die Frage, welche Sprache in den Kommentaren der Implementierung gewählt wird, soll ohne Bedeutung sein, wesentlich ist nur das Vorhandensein hoffentlich aufschlussreicher Kommentare.

Automotive Embedded Systeme
Effizientes Framework - Vom Design zur
Implementierung

Wietzke, J.; Tran, M.T.

2005, XXIII, 445 S., Hardcover

ISBN: 978-3-540-24339-7