Betriebssoftware für eine Fahrplattform unter besonderer Berücksichtigung der Echtzeitbedingungen

Christoph Peltz

Institut für Betriebssysteme und Computernetze
Technische Universität Braunschweig

26. Oktober 2009

Aufgabenstellung und Betreuung:
Prof. Lars Wolf und Dipl.-Ing. Dieter Brökelmann

Übersicht

Einführung

Hardware

Design und Implementierung

Untersuchung

Fazit

- Praktikum "Programmieren eingebetteter Systeme"
- Fahrzeug wird bewegt
- Teilnehmer sollen sich auf **ihre** Projekte konzentrieren
- Bisherige Software hat einige Kritikpunkte

- Verbesserung des Protokolls
- Erweiterbarkeit
- Performanz
- Echtzeit-Aspekt
- Umfassende Dokumentation

- AtMega 2561 (16 MHz, 8 KiB SRAM, 64 KiB Flash)
- PWM, I2C, UART, LCD

- Seite der Studierenden
- Anschluss an die Motorplatine über I2C
- Anschluss des WLAN-Moduls über UART

Design-Prinzipien

- Code in Module organisiert
- kurze, übersichtliche Funktionen
- generelle Funktionen
- wenige globale Variablen
- weniger inter-Modul Abhängigkeiten

Grundlegender Aufbau

- Endlos-Schleife als Hauptschleife
- Periodisches Aktualisieren

Protokoll

Die order_t-Struktur

- einfache Struktur
- flexibel Einsetzbar
- abstrakt, kein Wissen in der Struktur

Datenpfad der Befehle

Das Aktive-Brems-System

- de-/aktivierbar und konfigurierbar
- Verhindert, dass sich das Fahrzeug ungewollt bewegt

Das IO-Framework

- abstrahiert die Ein-/Ausgabe über UART und I2C
- es werden die selben Funktionen benutzt

Bibliothek

- Praktikumsplatine muss mit der Motorplatine kommunizieren
- Befehle erstellen
- Befehle direkt senden

Untersuchung

- Pins setzen/zurücksetzen (Makros)
- auslesen mit Logik-Analyser
- pin_set() und pin_clear() benötigen 250 ns
- pin_toggle() benötigt 250 ns

Verzögerungs-Problem

LCD-Problem

Vorher:

- Busy-Waiting benötigt 3250 us

Nachher:

- max. 1 Zeichen pro Iteration benötigt 60 us

Compiler-Optimierung

- Wechsel von -Os auf -O2
- Inlining von Funktionen (ca. 32% Verbesserung)

Modulo-Operatoren

- Benötigen viel Zeit (ca. 13 us pro %)
- Einfach zu ersetzen mithilfe von -, / und *
- Verbraucht dann bis zu 90% weniger Zeit

Echtzeit

- Ist es möglich Interrupts zu verpassen/verlieren?
- Nein. Interrupts werden zwischengespeichert und später ausgeführt.
- Worst Case: 8 Ints * 10 us = 80 us (schnellster Int tritt alle 138 us auf)

Vergleich

Ziele erreicht

- gleiche Funktionalität
- Erweiterbar
- schneller als Vorläufer
- Protokoll neu implementiert

Ideen für die Zukunft

- bessere Fehlererkennung des Parsers
- erhöhte Robustheit durch einen Sanity-Checker
- Verwendung der Energie-Spar-Funktionen