

Entwicklung eines Bootloader für über CAN verbundene Mikrocontroller



Jörg Diederich

Doktoranden- und Diplomandenseminar
Institut für Verteilte Systeme
AG Eingebettete Systeme und Betriebssysteme
Sommersemester 2007

Gliederung

Einleitung

Ausgangssituation

Problem und Lösungsmöglichkeit

Anforderungen

AVR

CAN-Verbindung

Vorgehen

Verwandte Arbeiten

Bootloader

avrdude

Gliederung

Einleitung

Ausgangssituation

Problem und Lösungsmöglichkeit

Anforderungen

AVR

CAN-Verbindung

Vorgehen

Verwandte Arbeiten

Bootloader

avrdude

Gliederung

Einleitung

Ausgangssituation

Problem und Lösungsmöglichkeit

Anforderungen

AVR

CAN-Verbindung

Vorgehen

Verwandte Arbeiten

Bootloader

avrdude

In-System-Programming



- Verwenden existierender und standardisierter I/O-Schnittstellen auf beiden Seiten
- Operationen umfassen Löschen, Schreiben und Lesen von bzw. in unterschiedliche Speicherbereiche



Motivation

Problem:

- minimale Anzahl von Kommunikationspartnern bei bisher verwendeten Kommunikationswegen
- Aufwand für ISP steigt mit zunehmender Anzahl zu bearbeitender Mikrocontroller

Lösungsmöglichkeit:

- Ausnutzen der existierenden Verbindung der Mikrocontroller über Bussysteme

CAN, TWI

Anforderungen



AVR AT90CAN Mikrocontroller

- Bootloader-Support mit 8kByte Programmspeicher
- integrierter CAN-Controller nach CAN 2.0B

CAN-Adapter



- Anbindung eines PC an einen CAN-Bus über Dongle oder Steckkarte
- Linux-Unterstützung mit Treiber und Library

Verwandte Arbeiten



avrdude

- (Standard-)Client mit verschiedenen Programmieradaptoren
- Verwendung der seriellen (RS232) und parallelen Schnittstellen des PC

Atmel AN914

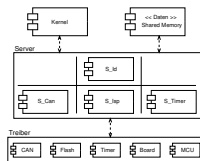
- Protokoll für CAN-Bus
- Geeignet für genau 1 Server
- Implementation für Windows

Entwicklung des Bootloaders

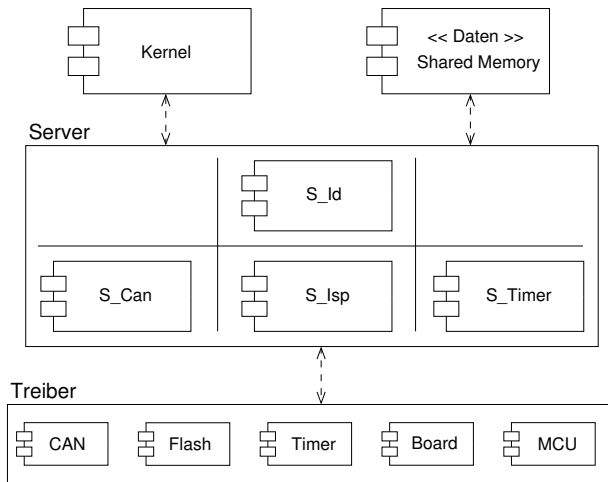
- Variation des in AN914 vorgeschlagenen Protokolls
Identifier einer CAN-Nachricht kennzeichnet sendenden Knoten



- lediglich lose Kopplung aller beteiligten Komponenten des Bootloaders angestrebt



Entwicklung des Bootloaders

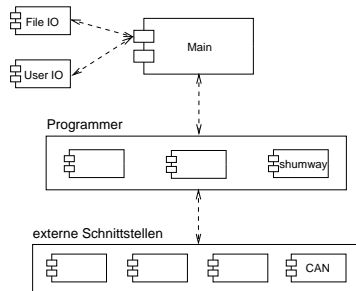


Erweiterung von avrdude

- neuer Programmieradapter:
shumway

Implementierung von
call-backs

- neue Schnittstelle: CAN
Verwendung der libpcan





Zusammenfassung

- Verwendung eines Protokolls für mehrere reaktive Knoten
- Entwicklung und Implementierung eines Bootloaders
 - jede Instanz besitzt eindeutigen Identifier
 - Abbruchbedingung nach Reset
 - Ansprung aus Applikation vorbereitet
- Fortgesetzte Anwendung von avrdude als Programmiersoftware
 - Geschwindigkeit Schreiben/Verify: ca. 1,5kB/s

Schwierigkeiten

1. Codesize
2. Asynchronität
3. Uabhngige Weiterentwicklung von avrdude

Vielen Dank für die Aufmerksamkeit.