



Konzeptperspektive

Sensorik- und Mikrocontrollerplattform im Formula Student Rennwagen

Johann-Nikolaus Andreae

1. Dezember 2008

AW1



Agenda

1. Ziele

- Formula Student
- Telemetrie
- Hardwaredesign

2. Ist-Zustand

- Sensorik gibt es im FS-Rennwagen
- Erforderliche Hardwarechnittstellen werden benötigt
- Einschränkungen der aktuellen Architektur

3. Lösungsansätze

4. Projektausblick

Projektziele Formula Student



- Schneller Rennwagen
- Geringe Produktionskosten [1]
 - Max. 25.000\$ für bei einer Produktion von 1000 Stück im Jahr
- Zuverlässige Konstruktion
 - 50% der Teams fallen im Ausdauerstest aus.
- Gutes Geschäftskonzept

Ziele der Telemetrie



Fahrzeugsverstnis erhöhen

- Erfassung der dynamischen Betriebseigenschaften
- Messdaten grafisch aufbereiten
- Warnmeldungen bei zu hohen Werten

Fahrverhalten verbessern

- Fahrer assistieren
 - Automatisieren der Schaltung
 - Antriebsschlupfregelung
 - Informieren über den Zustand des Wagens
- Leistungssteigerung
 - Optimale Steuerung des Motors
 - Veränderung der Fahreigenschaften
 - Energieverbrauch optimieren

Ziele für die Hardware



- Geringer Stromverbrauch
 - Leistung Motor 63kW
 - Leistung Lichtmaschine 1kW
- Geringes Gewicht
- Reduzierung des Hardwareaufwandes
 - Wenige kurze Leitungen
 - Weniger Gehäuse
- Kurze Abstände zu den Sensoren
 - Geringere Störungen der Messwerte

Unterschied zwischen PKW und FS-Rennwagen



Audi R8

- Motorleistung 309kW
- Höchstgeschwindigkeit: 301km/h
- Beschleunigung auf 100km/h: 4,6s
- Gewicht: 1560kg
- Radstand: 265cm

HAWK03

- Motorleistung 63kW
- Höchstgeschwindigkeit: 160km/h
- Beschleunigung auf 100km/h: 4,0s
- Gewicht: 283kg
- Radstand: 180cm

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Dehnungsmesstreifen
- Federweg

- Kühlwassertemperatur
- Lüftersteuerung

- Dehnungsmesstreifen

- Gangschaltung

- Motorsteuerung
- ASR
- Ölschnorchel
- Öltemperatur
- Öldruck
- Benzinverbrauch
- Tankfüllstand

- Beschleunigung
- Drehrate

- Display f

- WLAN
- Datenlogger
- Stromverbrauch
- Sicherung

- Pedalwinkel
- Bremsdruck
- Lenkwinkel
- Federweg

- Bremslicht

- Luftdruck vom Unterboden

- Dehnungsmesstreifen
- Federweg

- Kühlwassertemperatur
- Lüftersteuerung

- Dehnungsmesstreifen

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

Sensoren
Aktoren
Datenlogger
Sensor / Aktoren

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Dehnungsmesstreifen
- Federweg

- Kühlwassertemperatur
- Lüftersteuerung

- Dehnungsmesstreifen

- Gangschaltung

- Motorsteuerung
- ASR
- Ölschnecke
- Öltemperatur
- Öldruck
- Benzinverbrauch
- Tankfüllstand

- Beschleunigung
- Drehrate

- Display f

- WLAN
- Datenlogger
- Stromverbrauch
- Sicherung

- Pedalwinkel
- Bremsdruck
- Lenkwinkel
- Federweg

- Bremslicht

- Luftdruck vom Unterboden

- Dehnungsmesstreifen
- Federweg

- Kühlwassertemperatur
- Lüftersteuerung

- Dehnungsmesstreifen

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

Analyse
Auswertung in den Aktoren
Aktoren
Warnsignale
Aktor mit eigenen Sensoren

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Dehnungsmesstreifen
- Federweg

- Kühlwassertemperatur
- Lüftersteuerung

- Dehnungsmesstreifen

- Bremslicht

- Gangschaltung

- Motorsteuerung
- ASR
- Ölschnorchel
- Öltemperatur
- Öldruck
- Benzinverbrauch
- Tankfüllstand

- Beschleunigung
- Drehrate

- Display f

- WLAN
- Datenlogger
- Stromverbrauch
- Sicherung

- Pedalwinkel
- Bremsdruck
- Lenkwinkel
- Federweg

- Dehnungsmesstreifen
- Federweg

- Luftdruck vom Unterboden

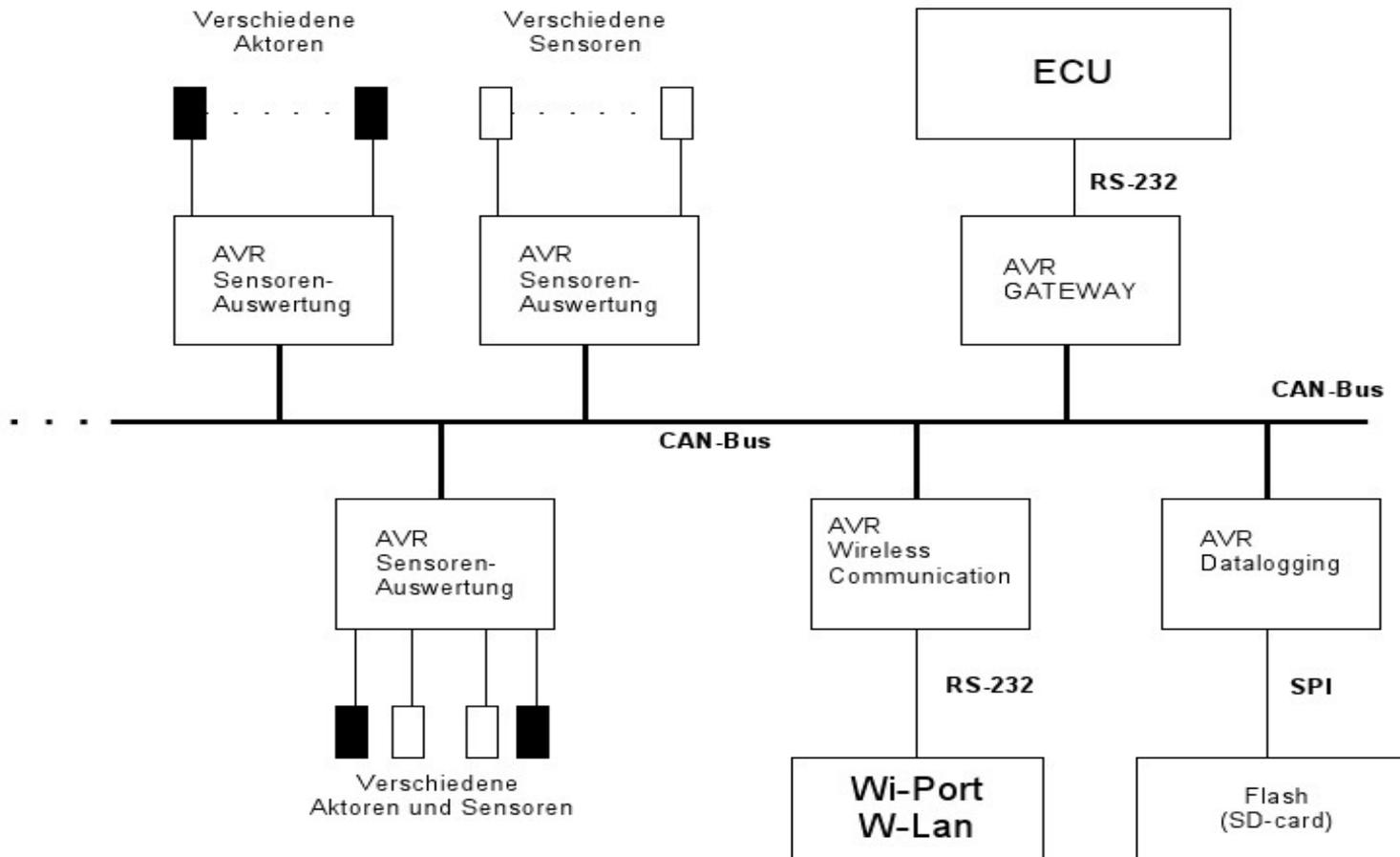
- Dehnungsmesstreifen

- Kühlwassertemperatur
- Lüftersteuerung

- Raddrehzahl
- Reifentemperatur
- Reifendruck

- Raddrehzahl
- Reifentemperatur
- Reifendruck

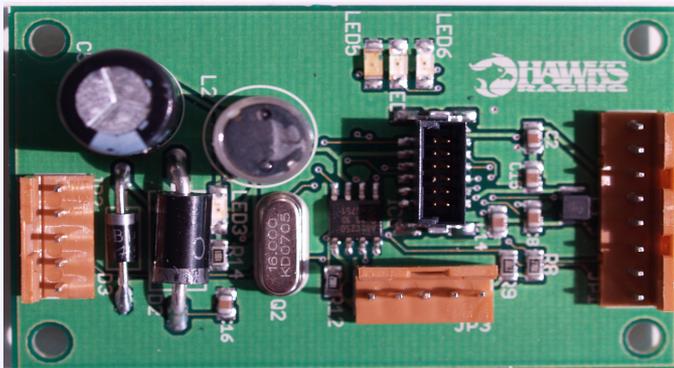
Aktuelles Sensor / Aktor Konzept



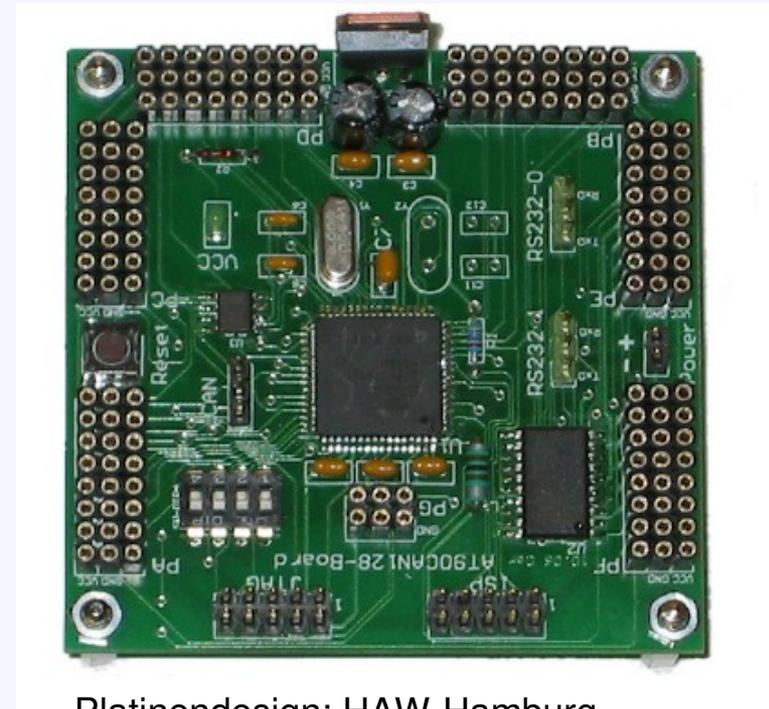
Sensor- / Aktormodule mit AT90CAN



- Sensor-Module mit AT90CAN 16MHz [3]
 - AVR8 8-Bit RISC
 - 53 x I/Os
 - 8 x Multiplexed 10-Bit AD-Eingänge
 - 1 x SPI, 1 x I2C
 - 1 x 8-Bit, 2 x 16-Bit Timer
 - Energie 30mA bei 5V => 0.15W
 - Kosten Stück 8€ (+500 Stück)



Platinendesign: HAWKS Racing



Platinendesign: HAW-Hamburg

Hardwareaufwand



- ~ 90 Sensoren im Zielkonzept
 - ~ 55 Analoge Messwerte
 - ~ 5 serielle Sensoren (SPI, I2C, RS232)
 - ~ 50 digitale I/Os
- In der aktuellen Architektur min. 25 AVR8
 - Hängt von der Auslastung der Module durch die Software ab.

Einschränkungen in der aktuellen Architektur



- TimeTriggert Softwarearchitektur
- TimeTriggert CAN-Bus
- Nur 1 Timer-Interrupt für Scheduler.
- Scheduler gibt Arbeitsrytmus vor
 - Jede Task hat ihren festen Zeitslot
 - Es sind keine Aktionen aufgrund von lokalen Sensoren mit hoher Geschwindigkeit möglich
 - Sensorereignisse mit anderer Datenrate lassen sich nicht verarbeiten

Lösungsansätze



- Parallelisierung durch zwei statt einem AVR pro Aufgabe
 - Einer für die Steuerungsfunktion
 - Einer für die Datenübertragung
 - Kommunikation zwischen AVRs notwendig
- Prozessor mit höherer Taktfrequenz
 - Prozessor ist zu bestimmten Zeiten von den Kommunikation belegt.
- Multicore Mikrocontroller
 - Wenige Hersteller: Parallax Propeller[4], Freescale MPC5668G [5]
- Verlagern von Funktionen in parallele Hardware
 - Einsatz von FPGAs: Hardware / Software Codesign
 - Vorgänge in der Hardware laufen parallel zur Software ab.
 - Hardwaremodule lassen sich den Sensoranforderungen anpassen.

Überprüfung des Einsatz von FPGAs



- Zusammenfassung von Modulen zu einer Einheit
 - Weniger Platinen
 - Mehr fehlende Funktionen im Störfall
- Komplexere Hardware
 - Höhere Herstellungskosten
- Analyse des Stromverbrauch
 - Rentiert sich nur durch Zusammenfassung von Modulen
- 32bit MicroBlaze Prozessorarchitektur
 - Überarbeitete Softwarearchitektur
 - Neues Hardwaredesign

Projektausblick



- Entwurf der Hardware / Software Partitionierung
 - Kriterien für die Algorithmen recherchieren
- Entwicklung von Kriterien für die Zusammenfassung von Sensor-Modulen
- Recherche über die Verfügbarkeit von IP-Blocks
- Implementierung auf dem Evaluation-Board
 - Digilent Nexus-2: 4 x 8 I/Os + 1 x 40 I/Os, fertige I/O-Module
 - Xilinx XAS1600E: Optimiert für Automotiv
- Entwicklung eines Prototypen

Zusammenfassung



- Telemetriesystem besteht aus Sensoren und Aktren
- Aktuelle Architektur kann keine Sensorevents außerhalb der Timeslots bearbeiten
- Einsatz von von Hardware / Software Codesign als Lösung muss evaluiert werden.



- [1] SAE: 2008 Formula SAE Rules.
<http://students.sae.org/competitions/formulaseries/rules/rules.pdf>
- [2] SCHUCKERT, Simon M.: Microcontrollerbasierte Telemetrie und Echtzeitauswertung von Sensordaten im Formula Student Rennwagen, Hochschule für Angewandte Wissenschaft Hamburg, Bachelorarbeit, 2007
- [3] Atmel: Datasheet AT90CAN. 2008. – URL
http://www.atmel.com/dyn/resources/prod_documents/doc7679.pdf
- [4] Propeller™ P8X32A Datasheet
<http://www.parallax.com/Portals/0/Downloads/docs/prod/prop/PropellerDatasheet-v1.1.pdf> Rev. 1.1 vom 2008-09-12
- [5] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MPC5668G&webpagelid=121120349534072559427E
Zugriff: 2008-12-01