

**Fakultät Maschinenbau**

**Projektaufgabe Fahrzeugmechatronik**

**Konzeption intelligenter Hardware (smart Sensor und smart Aktuator)**

im Studiengang

Maschinenbau

vorgelegt von: **Tim Dang, 70452663**

**Kai Bergmann, 70455883**

Betreuer: Jie Zhang

Abgabedatum: 19. Juli 2019

**Eidesstattliche Erklärung**

Wir, Kai Bergmann und Tim Dang, versichern durch unsere Unterschriften, dass wir die vorliegende Projektarbeit selbstständig und ohne Benutzung anderer, als der angegebenen Quellen und Hilfsmittel angefertigt, sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet haben.

Wolfenbüttel, 8. Juli 2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kai Bergmann Tim Dang

Inhalt

[1 Einleitung 1](#_Toc11843953)

[2 Konventionelle Sensoren 2](#_Toc11843954)

[3 Konventionelle Aktoren 3](#_Toc11843955)

[4 Smart Transducer 4](#_Toc11843956)

[4.1 Sensoren 4](#_Toc11843957)

[4.2 Aktoren 4](#_Toc11843958)

[4.3 Vorteile 4](#_Toc11843959)

[5 Aufbau Smart Transducer 5](#_Toc11843960)

[6 Bussysteme 6](#_Toc11843961)

[6.1 CAN 6](#_Toc11843962)

[6.2 Ethernet 7](#_Toc11843963)

[6.2.1 EtherCAT 7](#_Toc11843964)

[6.3 Profibus 7](#_Toc11843965)

[6.4 AS-Interface 7](#_Toc11843966)

[6.5 Bluetooth 7](#_Toc11843967)

[6.6 WiFi 7](#_Toc11843968)

[7 Bestehende Standards 8](#_Toc11843969)

[8 Industrial Internet of Things 9](#_Toc11843970)

[8.1 Möglichkeiten Smarter Module 9](#_Toc11843971)

[9 Konzipierung 10](#_Toc11843972)

[9.1 Anwendungsszenario 10](#_Toc11843973)

[9.2 Anforderungen 10](#_Toc11843974)

[9.3 Auswahl Sensorik und Aktorik 10](#_Toc11843975)

[9.4 Systemarchitektur 10](#_Toc11843976)

[9.5 Auswahl Bussystem 10](#_Toc11843977)

[10 Integration in höhere Systeme 11](#_Toc11843978)

[11 Fazit und Ausblick 12](#_Toc11843979)

[12 Literaturverzeichnis 13](#_Toc11843980)

[13 Abbildungsverzeichnis 14](#_Toc11843981)

# Einleitung

# Konventionelle Sensoren

# Konventionelle Aktoren

# Smart Transducer

## Sensoren

## Aktoren

## Vorteile

# Aufbau Smart Transducer

# Bussysteme

## CAN

Der CAN-Bus, von Bosch in den 1980er Jahren zur Kommunikation zwischen Steuergeräten in Kraftfahrzeugen entwickelt, basiert auf dem Grundprinzip der Nutzung einer zentralen Datenleitung zur Übertragung sämtlicher Signale. CAN steht für Controller Area Network, im Vergleich zu anderen Techniken kann aufgrund der zentralen Datenleitung Kabelmasse gespart werden, was insbesondere im Fahrzeugbau vorteilhaft ist.

Der einfachste Aufbau eines CAN-Systems besteht aus einer CAN-High und einer CAN-Low Leitung. Die Übertragung eines Signals findet auf den beiden Leitungen mit gegensinniger Potentialänderung statt, was das System äußerst störunempfindlich macht. Weitere Leitungen können der Erdung oder Stromversorgung dienen. In der folgenden Abbildung werden die Kommunikationseinheiten, auch Knoten genannt, als SG bezeichnet. Die beiden CAN-Leitungen werden über einen festgelegten Widerstand von 120Ω kurzgeschlossen.

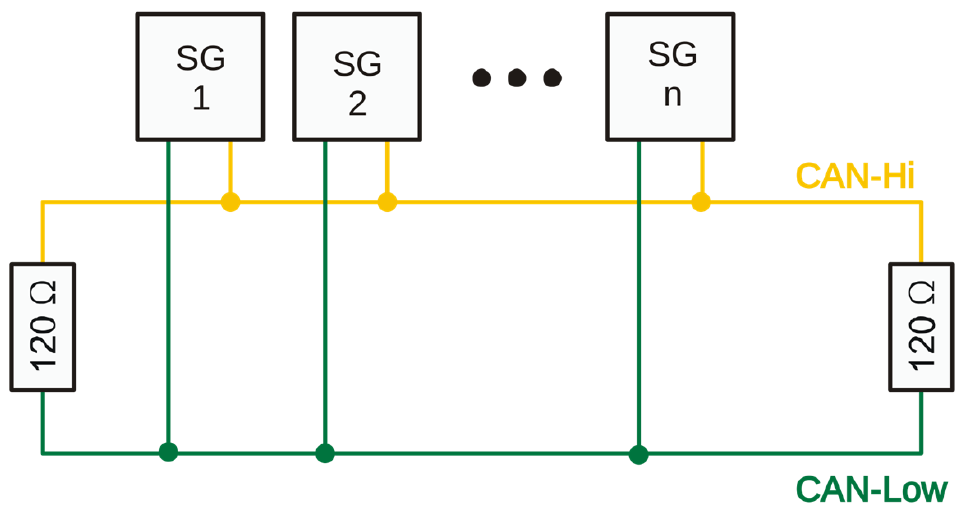


Abbildung 4: Schematischer Aufbau CAN-Bus

Grundlegend für die Funktion ist die Festlegung, dass zu einem Zeitpunkt stets nur ein Gerät sendet und ein Gerät empfängt. Bei langen Kabelwegen müssen daher niedrigere Abtastraten, die sogenannte Baudrate, verwendet werden. Typische Baudraten liegen zwischen 125 kBit/s und 1 Mbit/s.

Bei der Nachrichtenübertragung wird zwischen dem Standardformat und dem Erweiterten Format, die sich in der Länge des Identifiers unterscheiden. Das Standardformat verfügt über einen 11 Bit Identifier und das Erweiterte Format über einen 29 Bit Identifier. Dieser dient zur Schlichtung von Konflikten, falls mehrere Einheiten gleichzeitig mit der Datenübertragung beginnen. Somit wird der Verlust von Zeit und Information verhindert.

Jede Einheit verfügt über einen Identifier und kann somit gezielt angesprochen werden, die Auslesung erfolgt über Hexadezimalstellen ihres „Most significant Bits“ (MSB) und „Least significant Bits“ (LSB). [8]

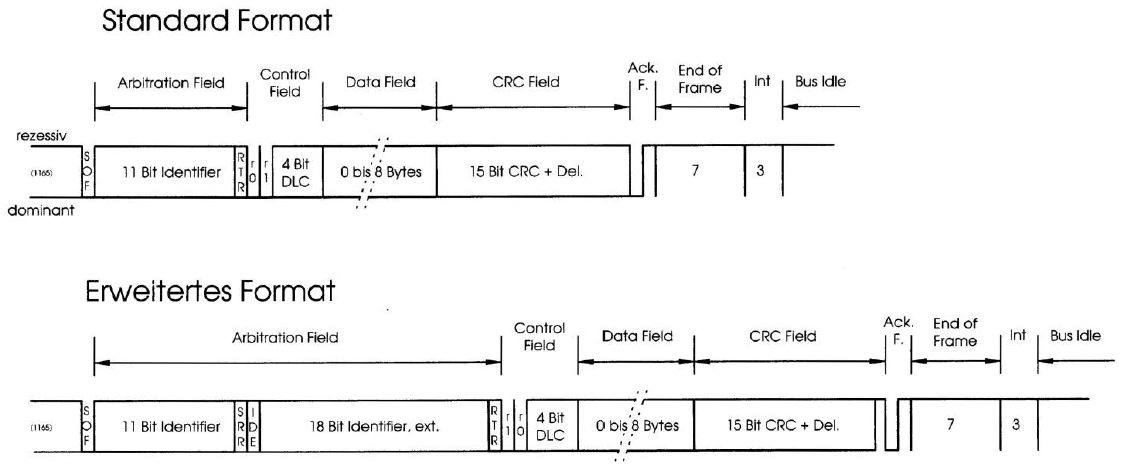


Abbildung 5: Aufbau der Identifier beim CAN-Bus [8]

## Ethernet

Ethernet ist eine von Robert Metcalfe Ende der 1970er Jahre konzipierte Übertragungstechnologie. Da Ethernet nicht die Darstellung der übertragenen Daten, sondern lediglich die Übertragung der Bytes definiert, ist es universell einsetzbar und weltweit weit verbreitet. Als Ethernet-Standard wird das physikalische Layer und zugehörige Zugriffverfahren sowie das Frame-Format beschrieben. Heutzutage sind unterschiedliche Spezifikationen verbreitet, die Geschwindigkeiten von 25 Gbit/s von bis zu 400 Gbit/s ermöglichen.

Netzwerke auf Ethernetbasis werden als sogenannte geswitchte Netzwerke ausgeführt. Dabei werden nicht alle Teilnehmer am gleichen Medium (Kollisionsdomäne) angeschlossen, sondern sie werden über Vermittlungsknoten (switche) und Punkt-zu-Punkt Verbindungen miteinander vernetzt. Man spricht auch von Vollduplex-Kommunikation, welche keine Kollisionen im Netzwerk zulässt.

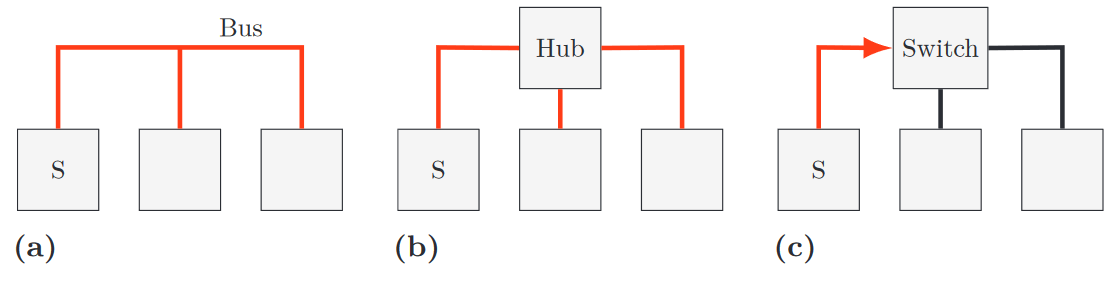


Abbildung 6: Kollisionsdomänen beim Senden (S): Bus(a), Hub (b), Switch (c) [9]

Wie in Abbildung 3 zu erkennen, liegt das Signal bei einem Bus immer bei allen Teilnehmern im Netzwerk an. Bei einem Netzwerk mit Hub ist die Charakteristik gleich, der Hub sorgt im sternförmigen Aufbau für eine Verstärkung des Signals an den Anschlüssen. Abweichend dazu sorgt der Switch dafür, dass die Kollisionsdomäne lediglich zwischen zwei Teilnehmern liegt. Die gesendete Botschaft wird vom Switch mit Hilfe der Zieladresse des Frames nur dahin weitergeleitet, wo der Empfänger erreicht wird. Infolgedessen ist die Effizienz bei switch basierten Netzwerken höher.

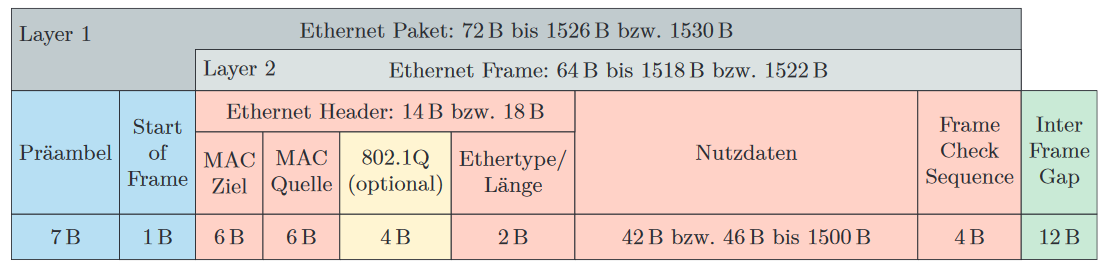


Abbildung 7: Ethernet Frame-Format Standard [9]

Bei der Darstellung eines Ethernet-Pakets unterscheidet man die des physikalischen Layers und das Ethernet Frame auf Layer 2. Das Paket beginnt mit einer 7 B Präambel und einem 1 B großen SFD (Start of Frame). Daraufhin beginnt der eigentliche Ethernet Frame. Zu Beginn wird die Ziel-MAC-Adresse und die Quell-MAC-Adresse übertragen, die jeweils 6 B groß sind. Bei diesen Adressen handelt es sich um eindeutige 48 bit lange IDs, welche ein Hardware Interface bezeichnen. Das letzte Feld beinhaltet das Typen- oder Längenfeld (2B). Vergebene Protokolltypen wie IPv6 (Internetprotokoll Version 6) sind im IEEE definiert. [10]

Der Datensatz muss mindestens 64 B groß sein, im Standard ist eine maximale Größe von 1500 B vorgesehen, sogenannte Jumbo-Frames von Netzwerkkomponentenherstellern können aber auch größere Frames unterstützen. [11]

Der gesamte Frame schließt mit der Frame Check Sequence (FCS) ab, mit der die Richtigkeit der übertragenen Daten überprüft werden kann. Nach jedem Datenpaket folgt mit der Inter Frame Gap (IFG) eine Ruhephase von 12 B. [9]

### EtherCAT

## PROFIBUS

PROFIBUS legt verschiedene technische Merkmale eines seriellen Feldbussystems fest, um Geräte von der Feld- bis zur Zellebene zu vernetzen.

## AS-Interface

## Bluetooth

## WiFi

# Bestehende Daten-Kommunikations-Standards

## IEEE 1451 Standard

## Object Management Group (OMG) Standard

## Serielle Schnittstellen

### Inter-IC Interface (I2C)

Ein I2C-Bus ermöglicht die Vernetzung von Mikrocontrollern, Sensoren und anderen Bausteinen wie A/D-, D/A-Wandler, RAM-Speicher etc. [16] Der I2C-Bus funktioniert nach dem Master-Slave Prinzip, wobei der Master der Knoten ist, der die I2C-Kommunikation initiiert und beendet, den Takt für die Synchronisation der Datenübertragung bereitstellt und Kommunikationsprobleme erkennt und löst. Jeder Slave besitzt eine Adresse, die 7 oder 10 Bit groß sein kann, sodass 128 bzw. 1024 unterschiedliche Slaves im Bus verbunden sein können. Zusätzlich bieten einige I2C-Bauteile (Speicher oder Sensoren) neben der fest programmierten Device Type Adresse eine über externe Anschlüsse einstellbare Device Chip Adresse an, was einen Anschluss und die Identifikation von mehr Geräten erlaubt. [17] Die Datenrate des I2C-Bus kann bis zu 5 Mbit/s betragen. [3]

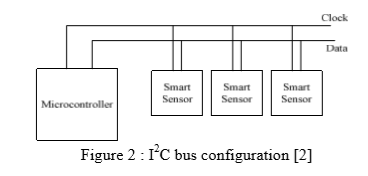


Abbildung 9: I2C-Bus Konfiguration [16]

Der Master wird mit dem Slave über bidirektionale Busleitungen, der Datenleitung (auch: SDA) und der Taktleitung (auch: SCL), verbunden. Die Busleitungen werden über pull-up Widerstände an die Versorgungsspannung angeschlossen, so ,dass eine UND-Verdrahtung der Geräte entsteht.

Ein neues Gerät kann an den Bus angeschlossen werden, indem die entsprechenden Anschlüsse eines Gerätes mit der Daten- bzw. Taktleitung verbunden werden. Dadurch, dass der Bus als synchroner Multi-Master Bus konzipiert wurde, kann jedes angeschlossene Gerät als Master oder Slave fungieren. [17]

Für ein System aus intelligenten Geräten, wäre die Schnittstelle bzw. der Bus dahingehend sinnvoll, dass jeder Teilnehmer durchzuführende Aktionen als Master vermitteln könnte.

### Serial Periphal Interface (SPI)

Bei dem SPI handelt es sich um ein Master-Slave System mit drei Leitungen für eine 8-Bit Datenkommunikation und einer Leitung um die Slaves anzusteuern. [16] Sie ist eine häufig genutzte Schnittstelle, um auf Sensoren auf der Leiterplatte oder außerhalb des Gehäuses zuzugreifen und ermöglicht wesentlich höhere Datenraten als bei asynchronen Schnittstellen. Ein Problem ist jedoch die Erhöhung des EMV-Risikos mit zunehmender Länge der Zuleitung, sodass Abschirmmaßnahmen erforderlich sind. [17]

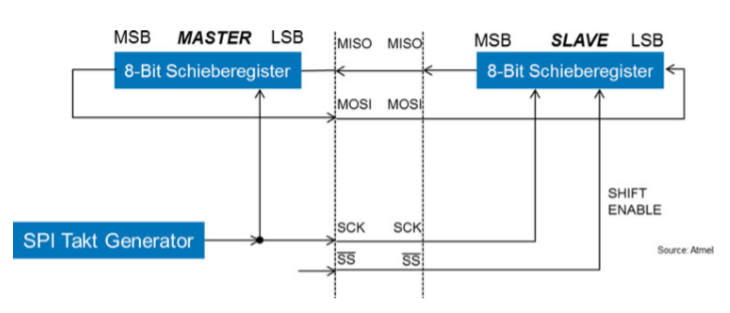


Abbildung 10: Schemazeichnung der SPI Schnittstelle [17]

Die SPI Schnittstelle wird genutzt, wenn einer Master CPU mehrere Knoten (Slaves) untergeordnet sind. Der Takt wird vom Master angegeben und der Slave über die Slave Select (SS) Verbindung angesteuert. Der Slave wird dabei erst aktiv, wenn SS ein Low-Signal gibt. Der Master schiebt die Daten aus dem Master-Schieberegister in das Slave-Schieberegister. Zeitgleich schiebt der Slave mit dem selben Takt Daten zum Master. Die beiden sind dabei über die Leitungen MISO (Master In Slave Out) und MOSI (Master Out Slave In) in einer Ringtopologie verbunden. Die Leitung Slave Select verhindert bei mehreren Slaves im Ring, dass mehrere Ausgänge parallel auf den Bus schalten. [17]

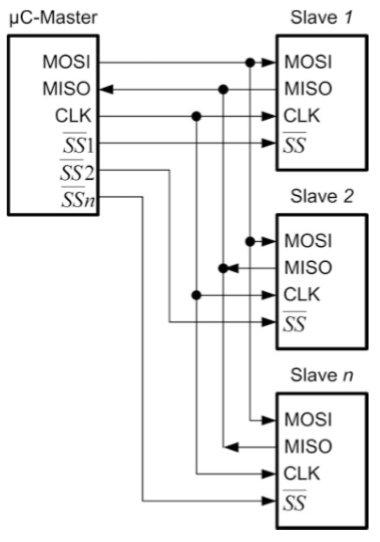


Abbildung 11: Beschaltung mehrerer Sensoren über SPI [17]

Für die Konfiguration der Schnittstelle legt man fest, in welcher Reihenfolge die einzelnen Bits eines Bytes übertragen werden. Überträgt man zuerst Bit 0, spricht man vom LSB (least significant bit), überträgt man zuerst Bit 7, spricht man vom MSB (most significant bit). Des Weiteren gibt es vier verschiedene Modi, die sich in der Polarität des Taktes (CPOL) und der Taktphase (CPHA) unterscheiden. [17] Die Polarität des Taktes gibt an, ob der Ruhepegel bei 0 oder bei 1 liegt, die Taktphase gibt an, bei welche Flanke (erste oder zweite) die Daten übertragen werden sollen.

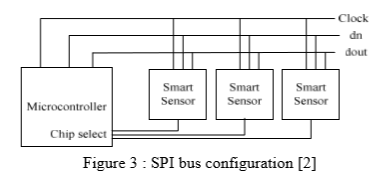


Abbildung 12: SPI Konfiguration [16]

Die Möglichkeit der Nutzung von vier verschiedenen Modi und die einfache Übertragung der Daten durch Schieben ermöglicht eine vielseitige Nutzung, sodass auch Einzelanfertigungen als SPI-Slave genutzt werden können. Die Hardware muss also nicht kompliziert gestaltet sein wie beim I2C-Bus (Signale sind beim I2C-Bus verschlüsselt). Allerdings ist die Größe des Netzwerkes dadurch beschränkt, dass jedes Gerät über die Slave Select mit dem Master verbunden sein muss. Eine Integration eines neuen Slaves gestaltet sich damit auch schwieriger als beim I2C-Bus. [16]

### Universal Asynchronous Receiver and Transmitter (UART)

UART ist eine serielle, asynchrone Schnittstelle, die häufig bei PCs und Mikrocontrollern genutzt wird. Daten werden über zwei unidirektionale Datenleitungen mit Datenraten zwischen 50 Bit/s und 500 kBit/s übertragen. [3] Der Ausgang zum Senden ist dabei TX und der Eingang zum gleichzeitigen Empfangen ist RX. [17] Durch die asynchrone Übertragung und damit dem Fehlen eines separaten Taktsignales, muss sich der Empfänger durch die Schaltflanken des Start- und Stoppbits und durch die eingestellte Baudrate auf die empfangenen Daten synchronisieren. [3] Die Baudrate und die Länge des Datenpakets (fünf bis neun Bit Größe) müssen daher vor der Übertragung festgelegt werden. Die Übertragung startet mit einem Startbit, der dadurch erkannt wird, dass er komplementär zum Ruhepegel ist. Es folgt die Übertragung eines Datenworts (zwischen fünf und neun Bit; i.d.R. acht Bit), auf die ein Paritätsbit und ein oder zwei Bit im Ruhepegel folgen, welche das Zeichen für die Beendung der Übertragung sind. Das Paritätsbit dient der Erkennung, ob eine fehlerhafte Kommunikation stattgefunden hat. [17]

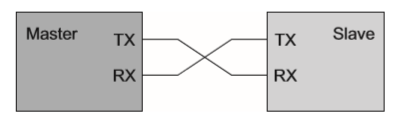


Abbildung 13: Konfiguration UART [3]

Die Schnittstelle hat einen Transmitter- und einen Receiver-Teil. Der Transmitter konvertiert die Bytes in einen seriellen Fluss aus den zu übertragenden Datenbits. Der Receiver nimmt den eingehenden Strom aus Datenbits auf und gruppiert diese in von 8-Bits, sodass diese als Bytes strukturiert werden können. [16]

Über UART lassen sich einfache Verbindungen zu PCs aufbauen, da eine digitale serielle Schnittstelle realisiert wird. [17]

# Industrial Internet of Things

## Möglichkeiten Smarter Module

# Konzipierung

## Anwendungsszenario

## Anforderungen

## Auswahl Sensorik und Aktorik

## Systemarchitektur

## Auswahl Bussystem

# Integration in höhere Systeme

# Fazit und Ausblick

# Literaturverzeichnis

# Abbildungsverzeichnis