Bachelorarbeit

Sensoranbindung mit IO-Link

Stefanie Schmidiger

Dozent:  
Thierry Prud‘homme

Hochschule Luzern – Technik & Architektur  
Horw, Hochschule Luzern – Technik & Architektur, Juni 2016

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc445992085)

[1.2 Entwicklungskosten 2](#_Toc445992086)

[2 IO-Link 3](#_Toc445992087)

[2.1 IO-Link Schnittstelle IP65/67 3](#_Toc445992088)

[2.2 Betriebsarten 4](#_Toc445992089)

[2.3 Datenübertragung 5](#_Toc445992090)

[2.4 IO-Link Protokoll 5](#_Toc445992091)

[2.5 Einbindung des IO-Link System in das übergeordnete Automatisierungssystem 9](#_Toc445992092)

[3 Softwareimplementierung 10](#_Toc445992093)

[Abbildungsverzeichnis 12](#_Toc445992094)

[Tabellenverzeichnis 13](#_Toc445992095)

[Literaturverzeichnis 13](#_Toc445992096)

[Anhang 14](#_Toc445992097)

Abstract

Einleitung / Ausgangslage

Das vorliegende Dokument befasst sich mit der Umsetzung des Lösungskonzepts im Rahmen des Moduls Produktentwicklung PREN 1 von Team 35 zur Aufgabenstellung an der Hochschule Luzern Technik und Architektur. Die Anforderung an das interdisziplinare Team besteht darin, ein Gerät zu entwickeln, welches autonom fünf Tennisbälle in einen Eimer schiessen kann. Die Position des Eimers soll beim Start ermittelt werden.

Problemdarstellung / Vorgehen

Das Lösungskonzept vom PREN 1 wurde in die Realität umgesetzt. Während der Realisierung und Testphase des Konzepts waren kleinere Änderungen am Konzept nötig, um das Ziel erfolgreich zu erreichen.

Beschreibung der Teilsysteme / Umsetzung

Für die Lösung der Aufgabenstellung wird eine Konstruktion, welche die Bälle in den Korb schiesst, entwickelt. Die Erkennung des Eimers erfolgt mit einer Kamera und der berechnete Winkel wird via USB Kabel an einen Mikrocontroller gesendet. Jener steuert dann einen Schrittmotor, der die Konstruktion horizontal um die gewünschte Anzahl Schritte dreht. Sobald die Ausrichtung des Gerätes beendet ist, werden die Bälle mit einem Hubteller zu den zwei drehenden Rollen befördert und so in den Eimer geworfen.

Ergebnisse / Ausblick

Das im PREN 1 erstellte Lösungskonzept wurde mit kleinen Anpassungen erfolgreich in die Realität umgesetzt. Das Ziel, alle Bälle unter einer Minute zu versenken wurde erreicht, denn die benötigte Zeit beträgt nur rund 20 Sekunden. Einzig beim Gewicht ist das Team nicht unter die 4 Kilogrammmarke gekommen, die Konstruktion wiegt 5.9kg.

Nun gilt es, den Roboter im Wettbewerb mit dem der anderen Teams zu messen.

# Einleitung

Die Firma Leister möchte für ihre Laserschweissmaschine die Sensordaten zentral auslesen und verarbeiten. Für einen ersten Demonstrationsaufbau sollen deshalb die Daten eines Leistungssensores und ein Pyrometers ausgelesen, in ein IO-Link Protokoll verpackt und an den IO-Link Master weitergesendet werden. Diese beiden Sensoren haben eine SPI resp. I2C Schnittstelle und es gilt, die Daten mit einem Mikrocontroller auszulesen und in ein IO-Link Protokoll einzufügen. Der Aufbau des Demobordes sollte deshalb über eine SPI und eine I2C Schnittstelle verfügen und die Daten im IO-Link Protokoll mit 24V Spannungspegel weiterleiten können.

# IO-Link

IO-Link ist ein herstellerunabhängiger Standard um mit Sensoren und Aktoren zu kommunizieren. Ein IO-Link System besteht aus IO-Link Masters und IO-Link Devices. Der IO-Link Master stellt die Verbindung zwischen Devices und System her. Der Master kommuniziert über Feldbusse oder produktspezifische Rückwandbusse. „Ein IO-Link Master kann mehrere IO-Link Ports (Kanäle) besitzen. An jedem Port ist ein IO-Link Device anschließbar (Punkt-zu-Punkt-Kommunikation). Somit ist IO-Link eine Punkt-zu-Punkt-Kommunikation und kein Feldbus.“ [1]

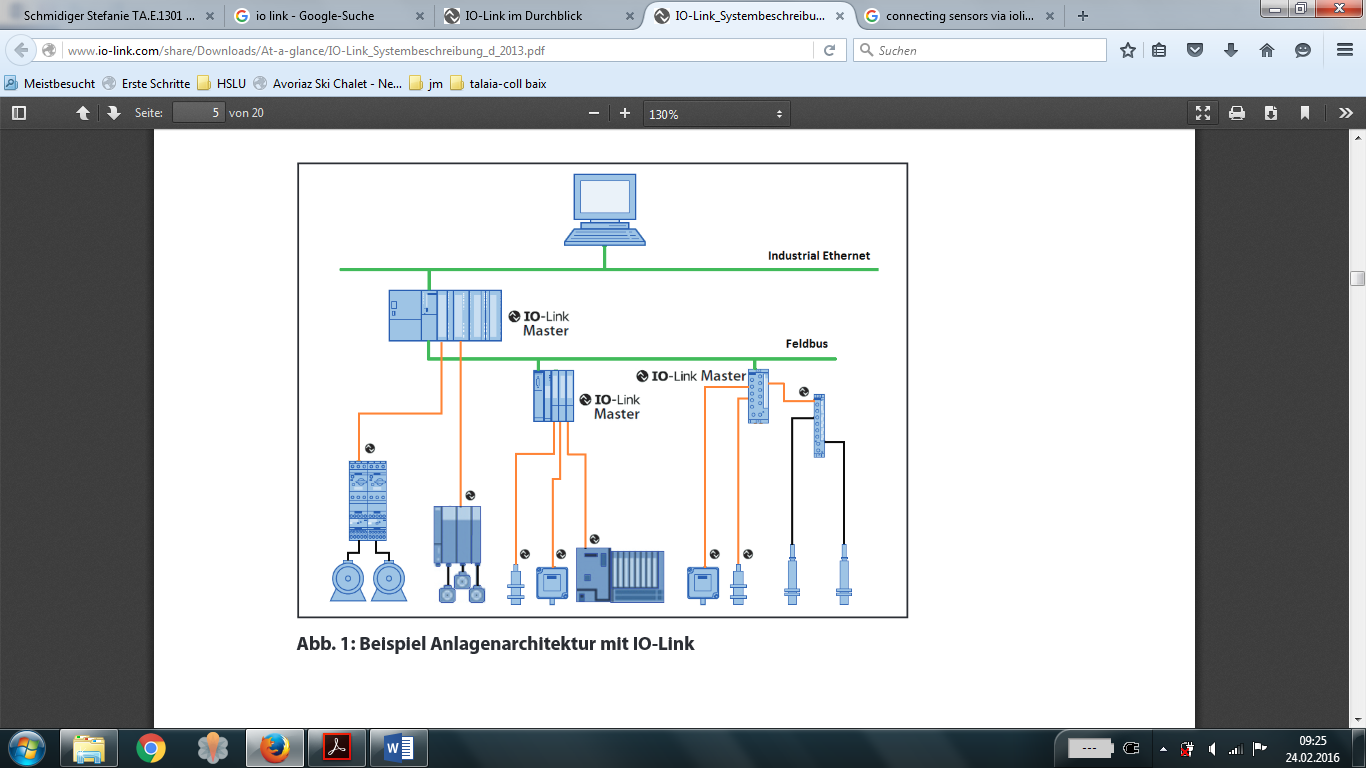


Abbildung 1 - Beispiel Anlagenarchitektur mit IO-Link [1]

## Physikalische IO-Link Schnittstelle

Bei IO-Link handelt es sich um eine serielle und bidirektionale Punkt-zu-Punkt Verbindung. Sensoren haben eine 3-polige Verbindung und Aktoren eine 5-polige Verbindung. Der Master verfügt über eine 5-polige Buchse (M5, M8 oder M12), wobei die Anschlussbelegung nach IEC 60974-5-2 Standard wie folgt definiert ist:

* Pin 1: 24V
* Pin 3: 0V
* Pin 4: Schalt- und Kommunikationsleitung

Die Anschlussbelegung kann ebenfalls aus Abbildung 2 entnommen werden.

Die Energieversorgung ist mit Pin 1 und Pin 3 gewährleistet und liefert maximal 200mA. Die maximale Leiterlänge für die ungeschirmte 3- bzw. 5-polige Verbindung zwischen Device und Master ist 20 m.

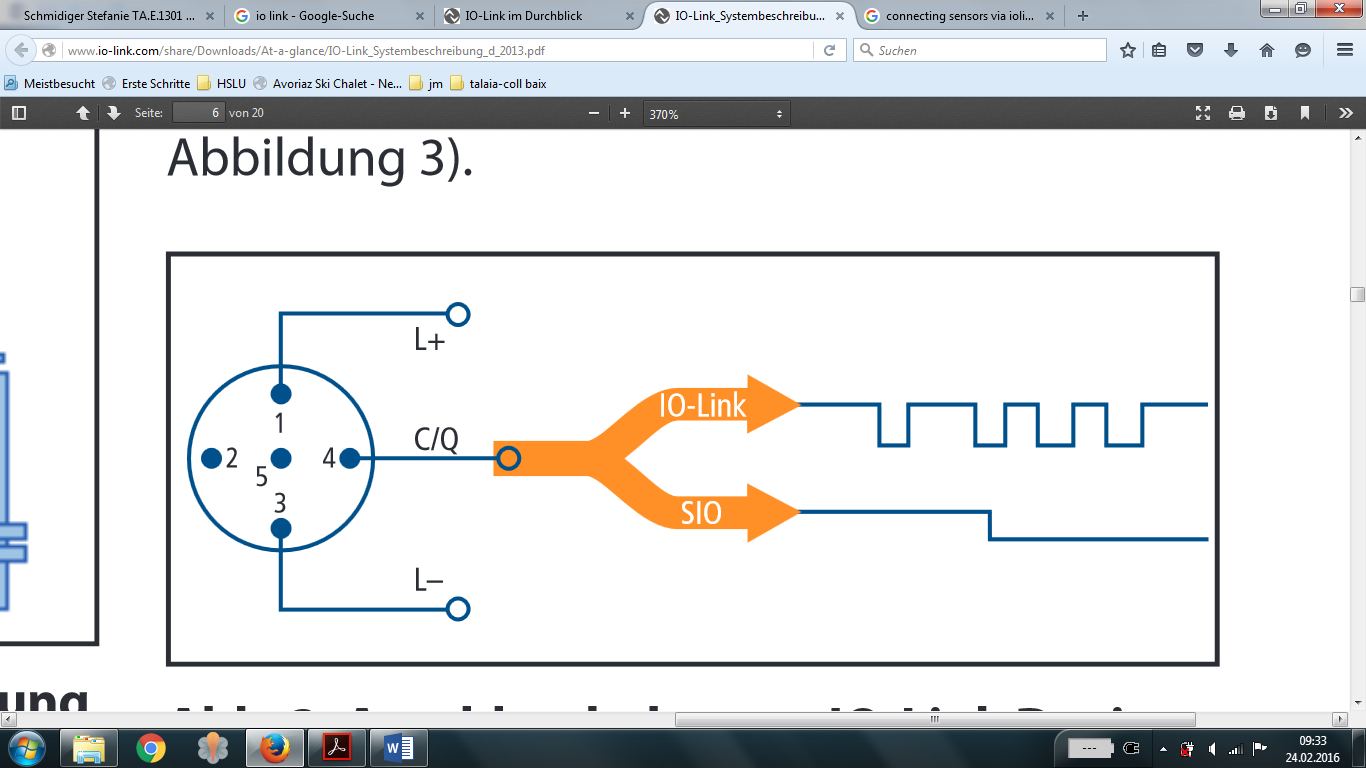


Abbildung 2 - Anschlussbelegung IO-Link Device [1]

Wenn eine Steuerung mit dem Sicherheitsstandard IP65/67, dann gibt es für die IO-Link Schnittstelle zwei verschiedene Porttypen:

### Port Class A

Die Funktion von Pin 2 und Pin 5 ist nicht vorgegeben und wird vom Hersteller definiert. Typischerweise wird Pin 2 zu einem zusätzlichen Digitalpin. Die Buchsengrösse variiert zwischen M5, M8 und M12.

### Port Class B

Dieser Porttyp ist für energieintensive Devices geeignet da Pin 2 und Pin 5 für eine zusätzliche galvanisch getrennte Versorgerspannung eingesetzt werden. Port Class B Steckverbindungen müssen eindeutig als solche gekennzeichnet werden und sind immer M12 Buchsen.

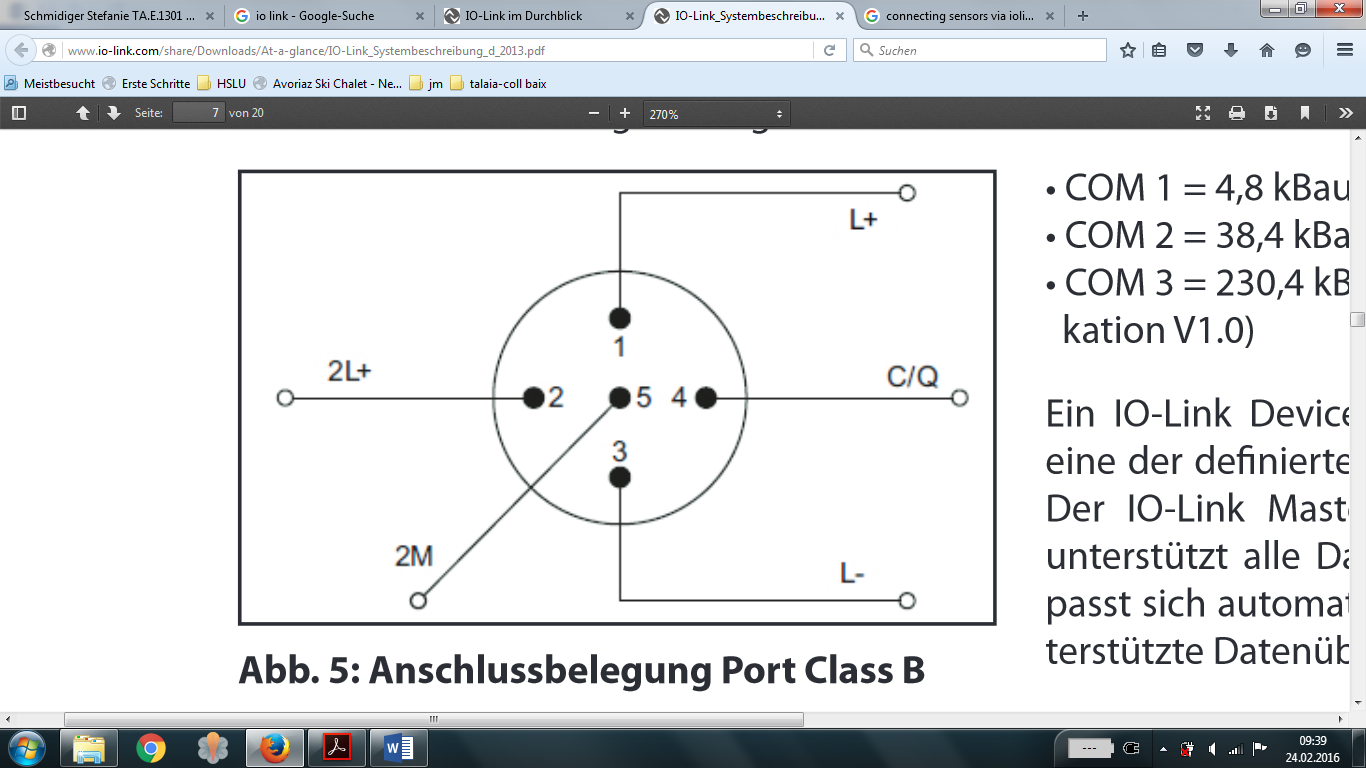


Abbildung 3 - Anschlussbelegung Port Class B

## Betriebsarten

Die IO-Link Ports des Masters lassen sich in drei verschiedenen Betriebsarten betreiben:

* IO-Link Der Port befindet sich in der IO-Kommunikation
* SIO Nach dem Einschalten befindet sich ein Port immer im SIO (Standard Input  
   Output) Modus und verhält sich wie ein digitaler Eingang. Der SIO Modus  
   ist ideal für binäre Devices die am Eingang nur High oder Low erwarten be-  
   ziehungsweise nur High oder Low senden.
* Deaktiviert Diese Betriebsart ist für unbenutzte Ports verwendbar

Im IO-Link Betriebsmodus sind drei Übertragungsgeschwindigkeiten (für die IO-Link V1.1) spezifiziert:

* COM1 4.8 kBaud
* COM2 38.4 kBaud
* COM3 230.4 kBaud

Mit IO-Link V1.0 sind nur die Übertragungsgeschwindigkeiten 4.8 kBaud und 38.4k Baud möglich. Beim neuen IO-Link V1.1 kann zusätzlich noch mit 230.4 kBaud übertragen werden.

Ein IO-Link Device unterstützt nur eine Übertragungsgeschwindigkeit und der Master passt sich deren automatisch an.

### Erstverbindung von Device und Master

Ist der Port des Masters auf IO-Link eingestellt, so wird er versuchen, mit dem angeschlossenen Device zu kommunizieren. Dazu sendet der Master zuerst in der höchsten definierten Übertragungsgeschwindigkeit einen Wake Up Impuls und wartet auf die Antwort des Devices. Bei Misserfolg versucht er es in der nächst niedrigen Übertragungsrate erneut.

Sobald der Master eine Antwort empfängt, beginnt die Kommunikation. Zuerst werden Geräteparameter ausgetauscht und allenfalls im System gespeicherte Parameter auf das Device übertragen. Anschliessend folgt die Übertragung der zyklischen Daten.

### IO-Link Device Description Datei

Die IO-Link Device Description (IODD) Datei ist eine elektronische Gerätebeschreibung und hält für die Systemintegration vielfältige Informationen bereit. Der Aufbau dieser IODD Datei ist dabei immer derselbe und wird von den IO-Link Konfigurationstools immer gleich dargestellt.

### Reaktionszeit und Zykluszeit

In der Gerätebeschreibungsdatei IODD des Devices ist ein Wert für die minimale Zykluszeit angegeben, welcher bestimmt, in welchen Abständen der Master das Device ansprechen darf um zyklische Daten zu erfragen bzw. übertragen. Zusätzlich benötigt der Master noch eine Bearbeitungszeit. Diese beiden Zeiten ergeben die Reaktionszeit des Systems. Da jedes Device eine andere minimale Zykluszeit haben kann, ist die Reaktionszeit verschiedener Devices am selben Master unterschiedlich. Mit der maximalen Übertragungsgeschwindigkeit der minimalen Protokollgrösse beträgt die minimale empfohlene Zykluszeit 400 µs.

### IO-Link V1.0 und IO-Link V1.1

An ein IO-Link Master der Version V1.0 können nur IO-Link Devices der Version V1.0 angeschlossen werden. Ein IO-Link Master der Version V1.1 kann jedoch gleichzeitig IO-Link Devices der Versionen V1.0 und V1.1 angeschlossen haben.

Die Datenübertragungsrate von 230.4 kBaud ist nur mit der Version V1.1 möglich. Deshalb wird ein IO-Link Master der Version V1.1 mit angehängtem V1.1 Device benötigt für diese schnelle Übertragung. Ein V1.0 Device an einem V1.1 Master kommuniziert mit maximal 38.4 kBaud.

## Datenübertragung

Bei der Kommunikation zwischen Master und Slave gibt es vier verschiedene Datenarten:

Prozessdaten (Process Data Innput / Process Data Output)  
Die Prozessdaten sind zyklische Daten wie beispielsweise Sensorwerte. Die Datenmenge kann dabei vom Device einmalig beim Einschalten festgelegt werden und liegt zwischen 0 und 32 Bytes.

Wertstatus  
Der Wertestatus kann mit den Prozessdaten zyklisch übertragen werden und zeigt an, ob die Prozessdaten gültig oder ungültig sind.

Gerätedaten  
Gerätedaten sind azyklische Daten die Parameter, Identifikationsdaten und Diagnoseinformationen enthalten. Diese Daten können in ein Device geschrieben oder aus dem Device gelesen werden.

Ereignisse (Events)  
Wenn ein Ereignis auftritt, signalisiert das Device dies dem Master und der Master liest es daraufhin aus. Ereignisse können Fehlermeldungen und Warnungen oder Wartungsdaten sein. Diese Ereignisse werden zur Steuerung weitergeleitet. Auch der Master kann Ereignisse haben und diese weiterleiten.

## IO-Link Protokoll

In diesem Kapitel werden der Kommunikationsservice und das IO-Link Protokoll spezifiziert.

Bei IO-Link wird der Austausch von zyklischen Prozessdaten immer vom Master initiiert, ersichtlich in Abbildung 4.

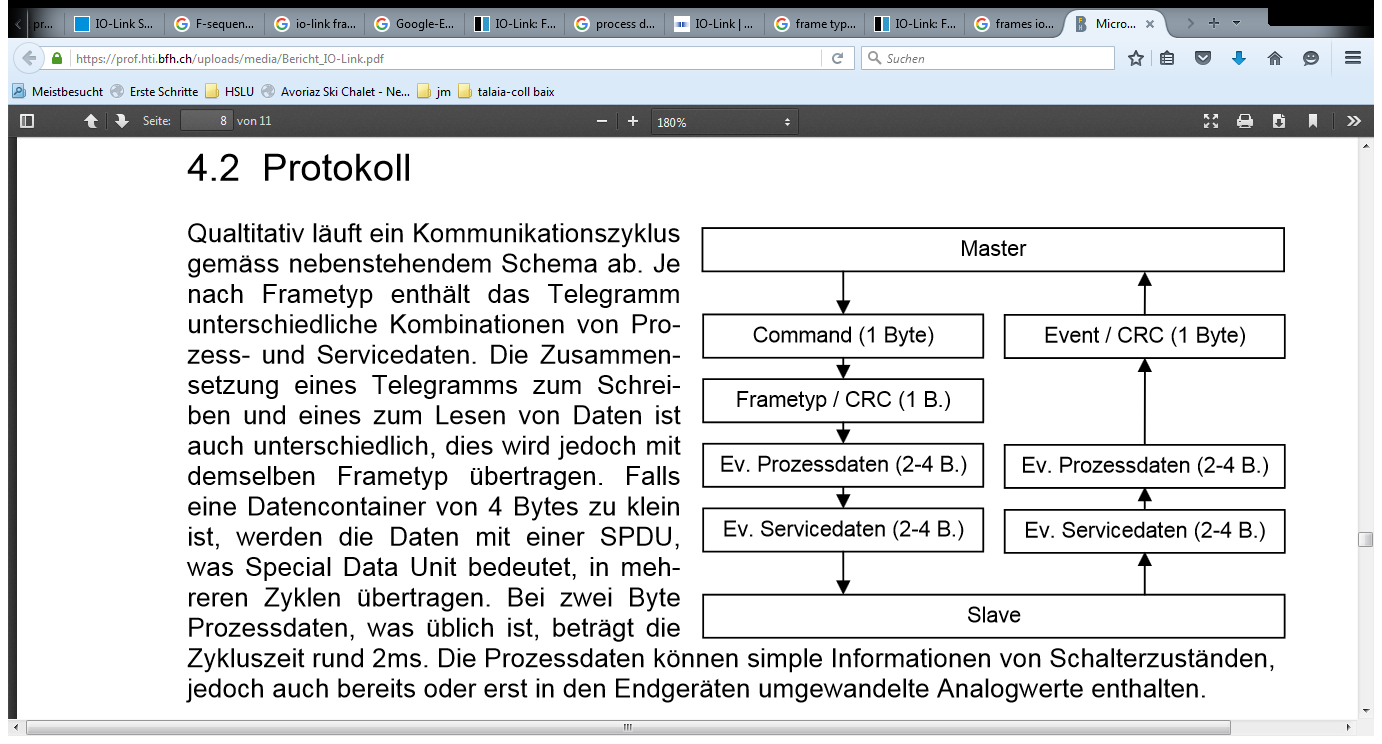


Abbildung 4 - Zyklischer Datenaustausch [3]

Command:

In diesem Byte werden die spezifischen Befehle abgesetzt.

Frametyp:

Die Antwort des Slaves kann in verschiedenen Frametypen sein: Typ 0, 1, 2. Dieses Byte  
spezifiziert den Frametyp und enthält auch die berechnete Prüfsumme.

Frame Typ 0 0 Bytes Daten  
Frame Typ 1 3-32 Bytes Daten in aufeinanderfolgenden Übertragungszyklen  
Frame Typ 2 3-32 Bytes Daten, variable Anzahl Nutzdaten pro Zyklus

🡨 Elektrotechnik und Elektronik für Maschinenbauer von Ekbert Hering,Rolf Martin,Jürgen

<http://www2.renesas.eu/_pdf/EPMC-PU-0103-1.0.PDF>

### ISO/OSI Modell

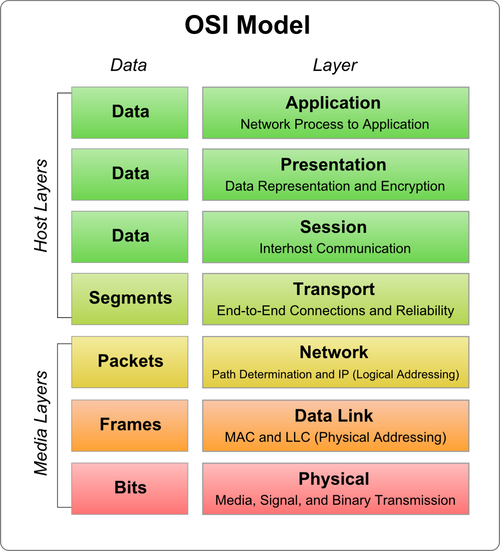
Das IO-Link Protokoll stützt sich ab auf das ISO/OSI Modell. Nach dem ISO/OSI Modell wird der Datenaustausch immer in sieben Layer unterteilt, die jeweils einen Teil der Kommunikation übernehmen.

Abbildung 5 - ISO/OSI Modell

Das IO-Link Protokoll benötigt nicht alle diese Layer, sondern baut nur auf drei davon auf, dem Physical Layer, dem Data Link Layer und dem Application Layer. Im folgenden werden die Funktionen dieser drei Layer erläutert.

Physical Layer  
Der 1. Layer im ISO/OSI Modell ist nur für die Bitübertragung verantwortlich. Die physikalische Layer kümmert sich um den Spannungspegel, den Übertragungskanal (Kabel, wireless etc) und um das physikalische Übertragungsprotokoll (z.B. UART, SPI, I2C etc).

Data Link Layer  
Der 2. Layer im ISO/OSI Modell ist für Punkt-Punkt Übertragung verantwortlich. Er schaut, dass die angekommenen Bits stimmen indem er Paritybits oder Prüfsumme anschaut. Der Data Link Layer ist aber nicht nur für die fehlerlose Übertragung verantwortlich, sondern ist auch das Interface zwischen der physikalischen Datenübertragung und der übergeordneten Datenverarbeitung. Der Data Link Layer nimmt Pakete vom übergeordneten Layer an, verkleinert diese in Frames, fügt eine Prüfsumme an und reicht sie an den Physical Layer weiter zur Übertragung.

Application Layer  
Der 7. Und letzte Layer im ISO/OSI Modell verbindet das Protokoll mit der Usersoftware. Der Application Layer ist die Schnittstelle zwischen den Daten und dem Nutzer (bzw. der Applikation).

Abbildung 7 zeigt die Implementierung der Layer im IO-Link Protokoll.

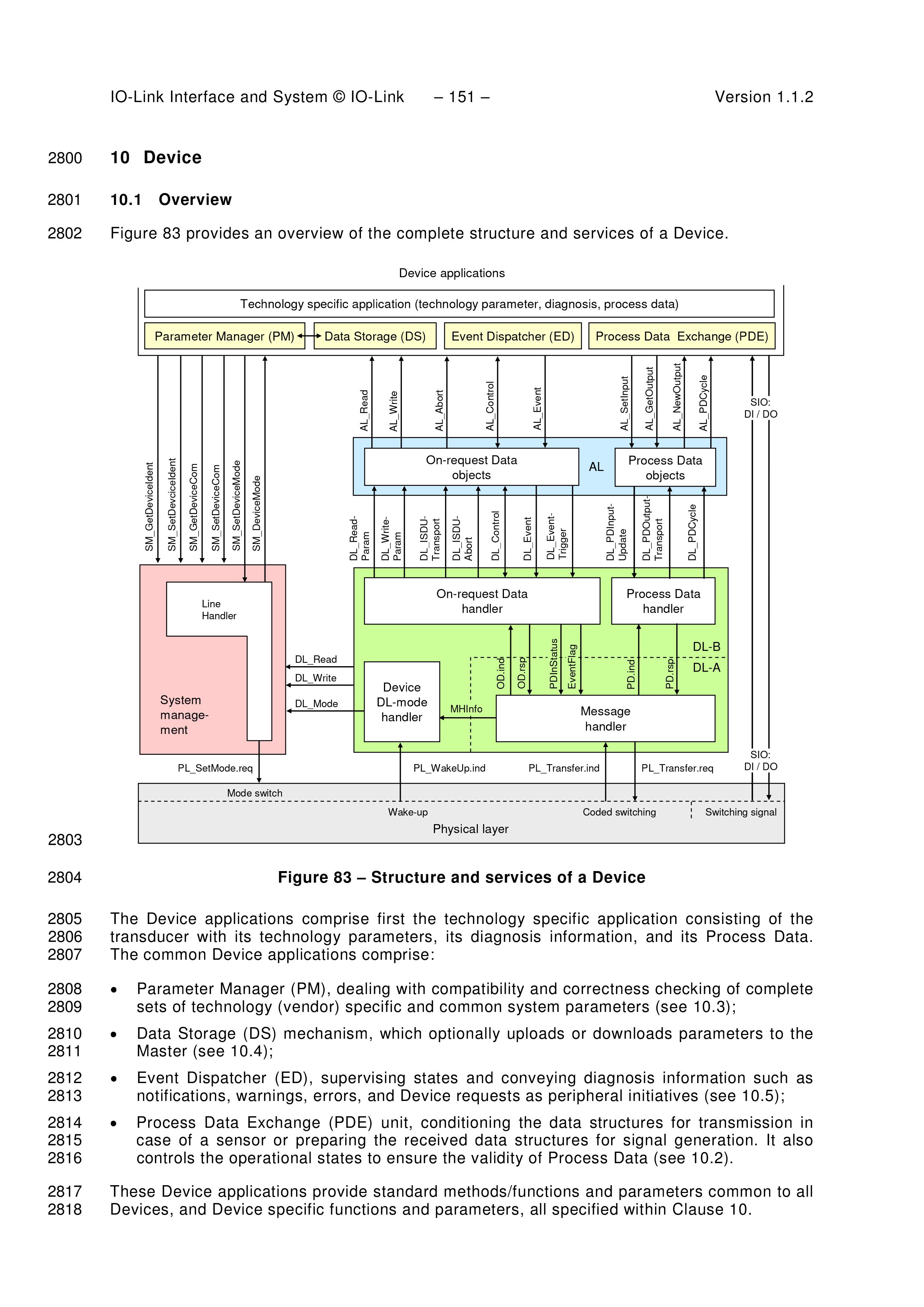


Abbildung 7 - Struktur und Service eines IO-Link Devices

Der Physical Layer übernimmt die Kommunikation mit dem IO-Link Master und informiert den darüberliegenden Layer mit Funktionsaufrufen. Der Data Link Layer (DL-A und DL-B) verarbeitet die Daten und kommuniziert mit dem Applikationslayer (AL) wiederum mit Funktionsaufrufen.

Die Aufgabe eines Funktionsaufrufes kann der jeweiligen Endung des Funktionsnamens entnommen werden. Dabei gibt es vier Möglichkeiten:

.req  
Beim Funktionsaufruf handelt es sich um einen Request. Der Initiator des Funktionsaufrufes bittet also um Daten.

.ind  
Die .ind Endung einer Funktion bedeutet, dass der Initiator des Funktionsaufrufes auf etwas Hinweisen möchte, beispielsweise einen Statuswechsel.

.resp  
Jede Funktion, die mit .resp endet, ist eine Antwort auf eine Anfrage oder ein Ereignis.

.cnf  
Wenn eine Funktion mit .cnf endet, so handelt es sich um eine Bestätigung.

Sofern die Funktionsnamen keine der vier Endungen aufweisen, so kann es sie in mehreren Ausführungen geben.

Eine detailierte Beschreibung der jeweiligen Funktionen kann dem Dokument „IO-Link Interface and System Specification“ [2] der IO-Link Community entnommen werden.

Physical Layer  
Bei IO-Link ist der physikalische Layer sehr robust, er arbeitet mit einem 24V Pegel. Die Bits werden mit einer halbduplexen UART Verbindung zwischen Master und Slave übertragen. Das Konzept einer halbduplexen Kommunikation kann aus Abbildung 5 entnommen werden.

Transmitter

Receiver

**IO-Link Master**

Transmitter

Receiver

**IO-Link Slave**

Abbildung 6 - Halbduplexe Kommunikation

Data Link Layer  
Beim zyklischen Datenaustausch ist vor allem der Cycle Frame Handler von Bedeutung. Er verpackt die Prozess- und Servicedaten in ein Frame. Bei einem Wake-Up Impuls übernimmt der Device DL-Mode Händler die Übermittlung der Initialisierungsdaten.

Application Layer  
Der Anwender, welcher das IO-Link Modul steuert, steht der Application Layer und die darauf implementierten Funktionen zur Verfügung um Daten abzufragen oder zu übermitteln.

# Sensoren

Das Ziel der Arbeit ist es, die Prozessdaten zweier Sensoren via IO-Link an die übergeordnete SPS Steuerung weiterzuleiten. Bei den beiden Sensoren handelt es sich um einen Pyrometer und einen Leistungsmesser. Die Sensorwerte werden jeweils seriell übertragen mit SPI respektive I2C.

## I2C Schnittstelle

Die Daten der Leistungsmessung werden mit I2C übertragen. Dabei wird die Leistung mit einer Photodiode gemessen, mit einem 10bit AD-Wandler abgetastet und im selben Baustein in ein I2C Protokoll verpackt. Dabei ist der Sensor als Slave implementiert. Die Implementation der Kommunikation kann aus Tabelle 1 entnommen werden.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Grösse** | **Wert** |
| I2C Slave Adresse | 7 bit | 0b 1001101 |
| Geschwindigkeit | Fast mode | 300 kbit/s |
| Sensordaten | 10 Bit | 1. Byte: 0 0 0 0 b9 b8 b7 b6 2. Byte: b5 b4 b3 b2 b1 b0 x x |
| Conversion Mode | Single conversion oder continuous conversion | Single conversion mode |

Tabelle 1 - I2C Protokoll Spezifikationen

Beim Single Conversion Mode wird der Sensorwert einmalig abgefragt und danach wird die Clockleitung vom Master wieder losgelassen. Der Sensor könnte auch im Continuous Mode betrieben werden indem der Slave ausgewählt wird mit der Adresse und danach kontinuierich ein Clock generiert wird. Der AD-Wandler wird dann kontinuierlich wandeln (typische Wandlungszeit: 8.96us, Analog Input Zeit: 1.12us, total 10.08us pro Wandlung) und den Sensorwert mit I2C weiterleiten, ohne dass dabei die Slave Adresse erneut versendet werden muss oder die Kommunikation abgesetzt wird.

Weitere Informationen über die I2C Implementation beim Slave können aus dem Datenblatt der Komponente MCP3021 von Microchip entnommen werden. Der MCP3021 ist ein AD-Wandler, der den digitalisierten Wert über I2C weiterversendet.

In der momentanen Betriebsart wird zuerst eine 7 Bit Adresse gefolgt von einem Readbit versendet und danach kommen zweimal 8 Bit Daten zurück. Jedes Bit wird bestätigt, es werden also insgesamt dreimal 9 Bit versendet, also total 27 Bit pro Sensorabfrage. Bei einer Geschwindigkeit von 300 kbit/s dauert diese Übertragung also 90 us.

## SPI Schnittstelle

Die Daten des Pyrometers werden über eine SPI Schnittstelle versendet. Dabei wird die Spannung über einer Photodiode gemessen und in einem Mikrocontroller verarbeitet. Dieser Mikrocontroller verfügt über eine SPI Schnittstelle, über welche die Daten abgefragt, kalibriert oder auch gesetzt werden können. SPI ist eine Vollduplexe serielle Schnittstelle. Da SPI nicht, anders als das I2C Interface, definierte Logikpegel hat, kann sie verschieden parametriert werden. Die Konfiguration der sensorseitigen SPI Konfiguration kann dem folgenden Abschnitt entnommen werden.

### SPI Konfiguration

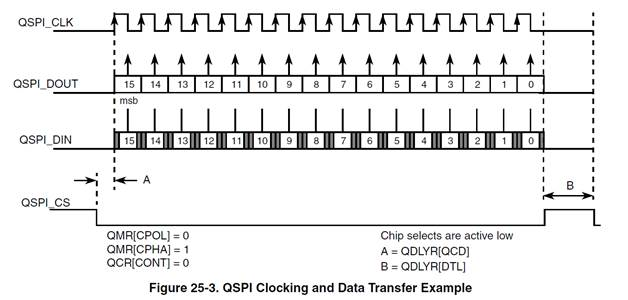
Die Aufsetzung der SPI Schnittstelle kann aus Abbildung 7 entnommen werden.

Abbildung 8 - SPI Konfiguration

Ein Datenpaket entspricht jeweils 16 Bit. Das MSB wird zuerst übermittelt. Die Leitung SlaveSelect ist low-aktiv und muss 3 Systemclockzyklen vor der Datenübertragung auf 0V gezogen werden. Zwischen zwei SlaveSelects müssen mindestens 600 Clockzyklen liegen um eine korrekte Datenverarbeitung im Mikrocontroller zu gewährleisten.



Es ist eine Baudrate von 1 MHz möglich.

Datenleitungen und Clockleitung sind idle-low. Der Pegel der Datenleitung wird bei der steigenden Clock-Flanke gesetzt und bei fallender Clock-Flanke ausgelesen.

### SPI Kommandos

Das Protokoll, das auf Seiten des Sensors im Mikrocontroller implementiert ist, sieht folgendermassen aus:

|  |  |  |
| --- | --- | --- |
| Protokoll |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Softwareimplementierung

Abbildungsverzeichnis

[Abbildung 1 - Beispiel Anlagenarchitektur mit IO-Link [1] 3](file:///C:\Users\Stefanie\Documents\Studium\6.Semester\BDA\Dokumentation\Dokumentationsvorlage.docx#_Toc445992109)

[Abbildung 2 - Anschlussbelegung IO-Link Device [1] 4](file:///C:\Users\Stefanie\Documents\Studium\6.Semester\BDA\Dokumentation\Dokumentationsvorlage.docx#_Toc445992110)

[Abbildung 3 - Anschlussbelegung Port Class B 4](file:///C:\Users\Stefanie\Documents\Studium\6.Semester\BDA\Dokumentation\Dokumentationsvorlage.docx#_Toc445992111)

[Abbildung 4 - ISO/OSI Modell 6](file:///C:\Users\Stefanie\Documents\Studium\6.Semester\BDA\Dokumentation\Dokumentationsvorlage.docx#_Toc445992112)

[Abbildung 5 - Struktur und Service eines IO-Link Devices 7](file:///C:\Users\Stefanie\Documents\Studium\6.Semester\BDA\Dokumentation\Dokumentationsvorlage.docx#_Toc445992113)

Tabellenverzeichnis

[Tabelle 1 Berechnungen 12](#_Toc420662401)

[Tabelle 2 Protokoll Schnittstelle 20](#_Toc420662402)

[Tabelle 3 Risikoanalyse 24](#_Toc420662403)

[Tabelle 4 Kosten 28](#_Toc420662404)

[Tabelle 5 Produktanforderungsliste 34](#_Toc420662405)

Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | PROFIBUS Nutzerorganisation e.V. (PNO), «IO-Link Systembeschreibung,» IO-Link, Juli 2013. [Online]. Available: http://www.io-link.com/share/Downloads/At-a-glance/IO-Link\_Systembeschreibung\_d\_2013.pdf. [Zugriff am Februar 2016]. |
| [2] | I.-L. Community, „IO-Link,“ Juli 2013. [Online]. Available: http://www.io-link.com/share/Downloads/Spec-Interface/IOL-Interface-Spec\_10002\_V112\_Jul13.pdf. |

Anhang

[A Produktanforderungsliste 34](#_Toc421909746)

[B Präferenzmatrix 35](#_Toc421909747)

[C Testfälle 36](#_Toc421909748)

[D Prints 39](#_Toc421909749)

[E CAD Zeichnungen 41](#_Toc421909750)

1. Produktanforderungsliste

In der Produktanforderungsliste werden die Anforderungen an das Produkt und die