

Abschlussbericht

Praxisforschungsprojekt

Integration einer Roboterfertigungszelle in den Fahrzeugherstellungsprozess

Hochschulbetreuer:	Prof. Dr.-Ing. Andreas Pretschner
Betriebl. Betreuer:	Elias Froschauer
Vorgelegt von:	Ronny Döring
Studiengang:	Elektrotechnik und Informationstechnik
Seminargruppe:	13EIM-AT
Matrikelnummer:	62308
Emailadresse:	RonnyDoering.Ing@gmail.com
Datum der Abgabe:	dd.mm.2016

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Abkürzungsverzeichnis	iv
1. Einleitung	1
1.1. Ausgangspunkt	1
1.2. Projektumfang	1
1.3. Aufgabenstellung	2
2. Planung	3
2.1. Vorbereitung	3
2.2. Steuerungskonzept	4
2.3. PROFINET Liste	7
2.4. Anlagenkennzeichnung	8
3. Test- und Präsentationsaufbau	13
3.1. Motivation	13
3.2. Komponentenauswahl und Aufbau	13
4. Anlagensoftware	16
4.1. Vorgehensweise	16
4.2. Hardware-Konfiguration	17
4.3. Safety-Software	18
4.4. Anbindung IT-Systemen	18
5. Nutzung der IPS-X Systeme	21
5.1. IPS-L	21
5.2. IPS-T	26
5.3. IPS-Q	27
Internetquellen	28
Sonstige Quellen	29
A. Steuerungskonzept	30

Abbildungsverzeichnis

1.1. Roboterzelle: Schematische Darstellung	2
2.1. Anlagenentstehungsprozess: Roadmap	4
2.2. Steuerungskonzept: Überblick	5
2.3. Steuerungskonzept: Sicherheitstechnik	6
2.4. Steuerungskonzept: Antriebstechnik	7
2.5. Steuerungskonzept: Softwarestruktur und Kommunikationswege	8
2.6. PROFINET-Liste	9
2.7. Anlagenkennzeichnungssystem: Masterstring	9
2.8. Anlagenkennzeichnungssystem: Überblick Gesamtanlage	10
2.9. Anlagenkennzeichnungssystem: Detaillierte Betriebsmittelkennzeichnung	11
3.1. Präsentationsaufbau: 3D-Prinzipdarstellung	15
4.1. Hardware-Konfiguration	17
4.2. PROFINET Topologie	17
4.3. Safety-Matrix	18
4.4. Vergleich Not-HALT Behandlung: Projektvorlage (li.), Testprojekt (re.)	19
4.5. SPS-seitige Konfiguration der TCP-Verbindungen	20
5.1. IPS-L Schnittstellenkontrakt	22
5.2. Sequenzdiagramm: Verwendung der IPS-L-Daten zur Typverwaltung	23
5.3. STRUCT: IPS-T Protokoll	27

Abkürzungsverzeichnis

AKZ	Anlagenkennzeichen
BMK	Betriebsmittelkennzeichen
CHAR	Character (Zeichen) 8 bit
DB	Datenbaustein
DNS	Domain Name System
FB	Funktionsbaustein
FC	Funktion
FG	Funktionsgruppe
FU	Frequenzumrichter
HMI	Human Machine Interface
INT	Integer (vorzeichenbehaftete Ganzzahl) 16 bit
IO	Input Output
IP	Internetprotokoll
IPS	Internationales Produktionssystem
IPS-L	Logistiknetz
IPS-Q	Qualitätsnetz
IPS-T	Techniknetz
OB	Organisationsbaustein
PN	PROFINET
RFC	Request for Comments
S7	STEP 7

SPS Speicherprogrammierbare Steuerung

TSAP Transport Service Access Point

UDT Datentyp (User Defined Type)

WT Werkstückträger

1. Einleitung

1.1. Ausgangspunkt

Die OSB AG ist ein Unternehmen mit deutschlandweit zwölf Niederlassungen und über 500 Mitarbeitern. Der Standort Leipzig ist in den Bereichen der Software und Automation tätig. Zu den Kunden zählen hauptsächlich lokal ansässige Produktionsunternehmen aus verschiedensten Industriezweigen. Dabei spielen die im Leipziger Einzugsgebiet vertretenen Automobilhersteller eine große Rolle. Das BMW Group Werk Leipzig gehört zu den langjährigen Kunden der OSB AG. Speziell für die Abteilung CFK wurde eine Vielzahl an Projekten durchgeführt.

Den Löwenanteil der BMW Niederlassung in Leipzig bildet jedoch der Technologiebereich Montage. Dieser Bereich ist hochautomatisiert mit verschiedensten Anlagen unterschiedlicher Hersteller und ausgelegt auf einen nahezu unterbrechungsfreien Betrieb. Um eine derartige Komplexität zu beherrschen, stützen sich die Anlagenhersteller auf einen sehr umfangreichen und detaillierten Standard, dem *TMO V1* Standard, welcher eigens von BMW für die Montage entwickelt wurde.

Der genannte Standard stellt einen kompletten Leitfaden für den Anlagenhersteller von Projektbeginn bis zur Übergabe an BMW dar. Damit wird sichergestellt, dass während der gesamten Projektphase die Konformität zum Standard eingehalten wird und bei der späteren Integration der Anlage in die Produktion keine Kompatibilitätsprobleme auftreten.

Um Anlagen herzustellen, zu betreuen oder zu erweitern verlangt BMW von potentiellen Dienstleistern einen Nachweis der Konformität zu diesem Standard. Im Fall der OSB AG am Standort Leipzig besteht dieser Nachweis in der Bearbeitung eines fiktiven Projektes, im Folgenden *Testprojekt* genannt, welches einen Großteil der geforderten Design-Richtlinien und Prozesse des Anlagenentstehungsprozesses abdeckt. Neben der Konformitätsbewertung der OSB AG hinsichtlich des TMO V1 Standards bildet das Testprojekt den thematischen Rahmen für diesen Bericht.

1.2. Projektumfang

Gegenstand des Testprojektes ist die Projektierung einer Montagestation. Dabei handelt es sich um eine Roboterzelle mit Fördertechnik entsprechend der schematischen Darstellung in Abbildung 1.1. Weiterhin sind zwei Bedien-Panels, sinnvoll angeordnete

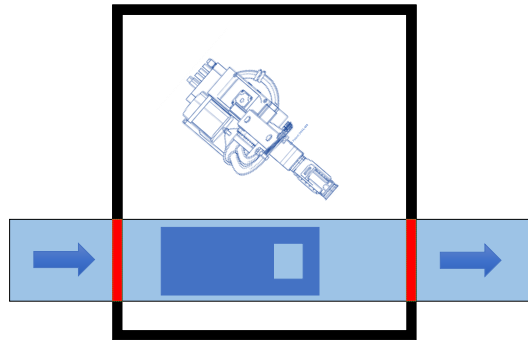


Abb. 1.1.: Roboterzelle: Schematische Darstellung

NOT-HALT Taster, eine Sicherheitstür sowie Lichtvorhänge zum Absichern der Ein- und Ausfahrtbereiche spezifiziert.

Die Station soll dabei folgende Funktionen erfüllen:

1. Übernahme des Fahrzeugs von Vorgängerstation
2. Bearbeitung in Abhängigkeit des Fahrzeugtyps
3. Übergabe des Fahrzeugs an Nachfolgestation
4. Kommunikation mit übergeordneten IT-Systemen

Da es sich um ein fiktives Projekt handelt wird die in Punkt 2 beschriebene Funktion nur beispielhaft implementiert. Es ist vorgesehen einen Art Greifer für den Roboter zu projektieren, der über zwei Zylinder verfügt. Auf Basis einer Typverwaltung wird dann entschieden, bei welchem Fahrzeugtyp welcher Zylinder aktiv wird.

1.3. Aufgabenstellung

Integration einer Roboterzelle in den Fahrzeugherstellungsprozess

Teilaufgaben:

1. Einarbeitung in die herstellerepezifischen Richtlinien
2. Konzeption der Steuerungstechnik
3. Konzeption interner und externer Schnittstellen
4. Auswahl und Aufbau eines Testsystems
5. Softwaretechnische Umsetzung

2. Planung

2.1. Vorbereitung

Der Anlagenentstehungsprozess ist an eine im Standard enthaltene Roadmap geknüpft. Abbildung 2.1 zeigt eine gegliederte Darstellung dieser Roadmap in etwas vereinfachter Weise. Der gesamte Prozess lässt sich in sechs Abschnitte gliedern. Jeder Abschnitt enthält wiederum verschiedene Schritte. Ein neuer Schritt wird erst abgearbeitet, wenn der vorangegangene abgeschlossen und durch BMW abgenommen ist.

Vor der eigentlichen Bearbeitung des Projektumfanges erhält der Anlagenhersteller eine aktuelle Version der Dokumente, die den Standard bilden. Das sind Dokumente folgender Kategorien:

- Kaufteile
- Planung
- EPLAN
- Hardware Konstruktion
- Antriebstechnik
- IT-System Schnittstellen
- Roboter
- Visualisierung mit TIA-WinCC
- Classic SPS Siemens SIAMTIC
- Dokumentation
- Fördertechnik
- Allgemeines

In seiner Gesamtheit umfasst der TMO V1 Standard über 250 Dokumente, die zu einem großen Teil im Excel-, Word- oder PDF-Format vorliegen. Neben einer ganzen Reihe an Richtlinien, steht eine Vielzahl an fertigen Projektvorlagen zur Verfügung. Diese erleichtern die spätere Bearbeitung und sorgen dafür, dass alle Projekte auf einer gleichen Basis aufbauen.

2. Planung

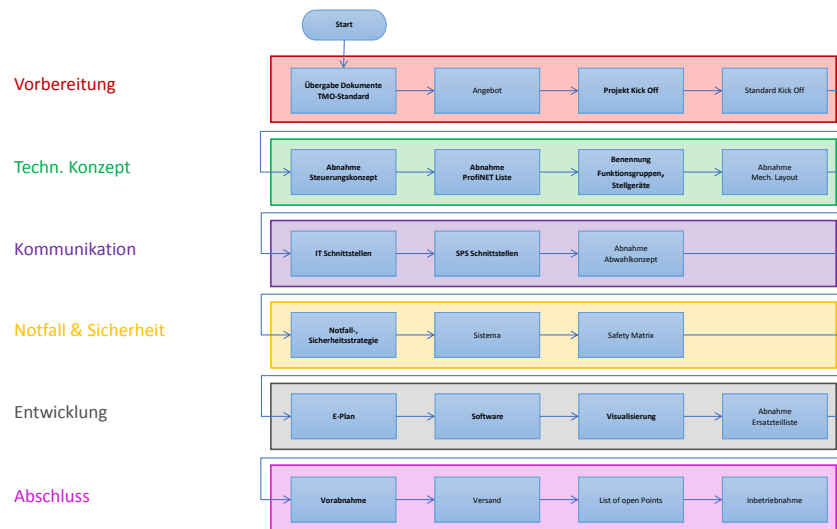


Abb. 2.1.: Anlagenentstehungsprozess: Roadmap

Nach dem der Angebotsprozess abgeschlossen ist und die Formalitäten zwischen Anlagenbetreiber und Hersteller geklärt sind, findet schließlich der Kick Off auf Leitungsebene und anschließend auf technischer Ebene statt.

2.2. Steuerungskonzept

2.2.1. Überblick

Im ersten Projektierungsschritt wird ein Konzept erarbeitet, welches alle wesentlichen Komponenten der zukünftigen Anlage sowie deren Verknüpfung miteinander enthält (siehe Abb. 2.2). Ziel des Steuerungskonzeptes ist es, eine Art Leitfaden für den gesamten Entwicklungsprozess zu schaffen, auf den sich Konstrukteure und Programmierer während des gesamten Projektes stützen können. Die folgenden Abschnitte enthalten jeweils Auszüge und dienen der Veranschaulichung des Inhaltes und Umfanges des Steuerungskonzeptes. Das komplette Dokument liegt dem Bericht als Anhang A bei.

2.2.2. Sicherheitstechnik

Die Anlagensicherheit ist ein wichtiger Teil der Projektierung. Das Schema in Abbildung 2.3 zeigt die Verwendung einer Safety-SPS. Die Verbindung zu den dezentralen Komponenten wie ET200 SP, ET200 PRO oder den Murr Safety Modulen erfolgt über

2. Planung

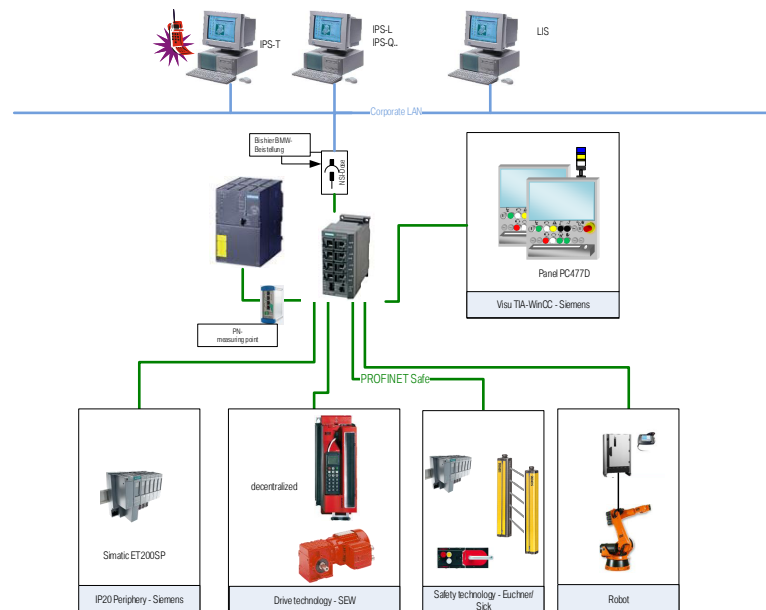


Abb. 2.2.: Steuerungskonzept: Überblick

den Industrial-Ethernet-Standard PROFINET¹. Zur Datenübertragung dient das Protokoll PROFIsafe². An die dezentrale Peripherie werden die sicherheitsrelevanten Baugruppen per digitaler Signale verbunden. Dafür werden spezielle sichere Eingangs- und Ausgangsbaugruppen verwendet. Konkret kommen in diesem Fall NOT-HALT-Taster, Türverriegelungen, Lichtvorhänge und Zustimmungstaster zum Einsatz. Im Schema werden die einzelnen Komponenten nur mit jeweils einem Vertreter dargestellt. Der Detaillierungsgrad in der Darstellung der Sicherheitstechnik ist zwar nicht allzu hoch, zeigt aber den grundsätzlichen Aufbau. Aus der Abbildung geht hervor, dass die Verknüpfung aller sicherheitsrelevanter Komponenten innerhalb der Safety-Software geschieht. Hart verdrahtete Sicherheitskreise werden nicht verwendet.

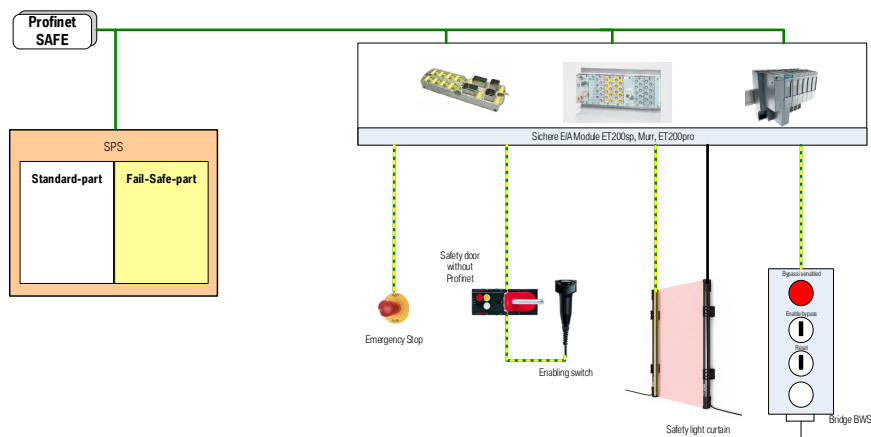


Abb. 2.3.: Steuerungskonzept: Sicherheitstechnik

2.2.3. Antriebstechnik

Die zu projektierende Roboterzelle verfügt über ein sehr übersichtliches Antriebskonzept. Lediglich die Fördertechnik wird mit Frequenzumrichter (FU)-Antrieben ausgestattet. Es handelt sich um Movifit FC Antriebe der Firma SEW. Jeder FU ist in der Lage bis zu drei Motoren zu steuern. Darüber hinaus verfügen die Antriebe über sichere digitale Ein- und Ausgänge, welche anstelle dedizierter dezentraler Peripherie verwendet werden können.

Die FUs werden an drei verschiedene Netze angeschlossen. Das 24 V-Netz versorgt die digitalen Ein- und Ausgangsbaugruppen sowie den Steuerteil. Über das 400 V-Netz wird die Leistung für die Motoren übertragen. Zur Kommunikation von Steuerbefehlen und Statusinformationen zwischen SPS und FUs erfolgt per PROFINET.

2.2.4. Softwarestruktur und Kommunikationswege

Etwas komplexer als die Antriebstechnik ist der Aufbau der Softwarestruktur und Kommunikationswege. Abbildung 2.5 veranschaulicht das Zusammenspiel aller Komponenten.

¹Profinet basiert auf Ethernet-TCP/IP und ergänzt die Profibus-Technologie für Anwendungen, bei denen schnelle Datenkommunikation über Ethernet-Netzwerke in Kombination mit industriellen IT-Funktionen gefordert wird. [2]

²Das PROFIsafe-System ist eine Erweiterung des Profibus- und PROFINET-Systems. Mit dem System können frei programmierbare Sicherheitsfunktionen ausgeführt und die hierfür notwendigen sicheren Ein- und Ausgangsdaten von und zu den sicheren I/O-Geräten übertragen werden.[1]

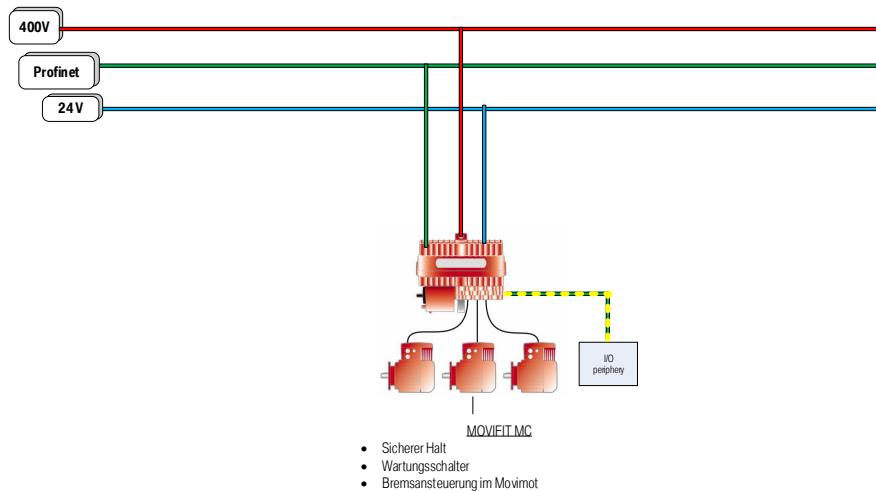


Abb. 2.4.: Steuerungskonzept: Antriebstechnik

Innerhalb der Station erfolgt sämtlicher Informationsaustausch Software-basierter Baugruppen mittels PROFINET (grün dargestellt). Das Kernelement bildet die Siemens 319F SPS. In dieser laufen alle Informationen der Panels, FUs, Robotersteuerung und Schraubersteuerung zusammen. Die SPS dient als Kommunikationsknoten zwischen der Station und den übergeordneten IT-Netzen. Die einzige Ausnahme bildet die Schraubersteuerung. Sie kommuniziert ihre Daten ohne Umwege über die Steuerung direkt an das IPM-System. Als Gegenstand des Projektes werden nur die Anbindungen an die Systeme IPS-Q, IPS-L und IPS-T projektiert.

2.3. PROFINET Liste

Ein weiterer Meilenstein im Anlagenentstehungsprozess (s. Abb. 2.1) ist die Abnahme der PROFINET-Liste (PN-Liste). Dabei handelt es sich um ein Excel-Dokument, das nach der BMW-Vorlage angefertigt wird. Ein Ausschnitt der PN-Liste für das Testprojekt ist in Abbildung 2.6 dargestellt.

Der obere Teil der Liste wird von der IT-Abteilung des Auftrag-vergebenden BMW-Standortes ausgefüllt. Neben dem Anlagenkennzeichen (AKZ) und dem Standort der Anlage, wird in diesem Teil die IP-Netzkonfiguration festgelegt.

Den unteren Teil füllt der Projektbearbeiter aus. Jeder PN-Teilnehmer erhält eine feste IP-Adresse im Rahmen der vorgegebenen Subnet-Maske. Weiterhin wird allen Geräten ein DNS-Name zugeordnet. Dieser setzt sich aus dem Kürzel für den Standort, dem AKZ und einer dreistelligen Device-ID zusammen. Die Device-ID ist ebenfalls in der



Entsprechend ihrer Typen belegen einige Gerät im PN-Netz einen festen Bereich im Prozessabbild der Eingangs- und Ausgangsdaten der SPS. In der PN-Liste werden jedem dieser Bereiche definierte Adressen zugeordnet. Dabei wird zwischen Standard-Adressen, Safety-Adressen und Diagnose-Adressen (>2048) unterschieden.

Masterstring

8

2. Planung

IE-Netz Example Profinetliste für Standard TMO V1						
AKZ		M9TA1G01				Hammerschmidt TP-471, 22.08.19
Location				OW20		
Net-ID		010.172.237.128				
Gateway		010.172.237.129				
Subnet		255.255.255.128				25-Bit -> 128. 252 IP-Adressen

Profinet-Network									
NSI	IP-Adresse	DNS-Name	Device	S7-Devicename	Description	Process image		F-Peripherals	
		123456789012345				E-Adr	A-Adr	F-E-Adr	F-A-Adr
xxxx	10.172.237.135	OW20M9TA1G01RSW	10	CC001-KF01-Root-Switch	+CC001-KF01 SCALANCE X 216 ROOT SWITCH				
		M9TA1G01			+CC001 ROOT SWITCH Alias				
	10.172.237.136	OW20M9TA1G01011	11	CR001-KF01-Switch	+CR001-KF1 Switch 1 SCALANCE X208				
	10.172.237.137	OW20M9TA1G01012	12	CD001-KF01-Switch	+CD001-KF1 Switch 1 SCALANCE X208				
	10.172.237.150	OW20M9TA1G01PLC		CC001-KF10-PLC	+CC001-KF10 S7-300 CPU319F-3 PN/DP				
	10.172.237.155	OW20M9TA1G01025	25	CC001-KF05-PN-Coupler-1-xxx	+CC001-KF05 PN/PN Coupler side 1 with xxxxxx	120-135	120-135	1200-1217	1200-1217
	10.172.237.156	OW20M9TA1G01026	26	CC002-KF05-PN-Coupler-2-yyy	+CC001-KF05 PN/PN Coupler side 2 with yyyyyy				
	10.172.237.160	OW20M9TA1G01EMG			+CE001 Janitza Power measurement				
	10.172.237.161	OW20M9TA1G01V01			+OP001-KF20 Siemens IPC477D-22" WinCC				
	10.172.237.162	OW20M9TA1G01V02			+OP002-KF20 Siemens IPC477D-22" WinCC				
	10.172.237.171	OW20M9TA1G01041	41	OP001-KF70-ET200SP	+OP001-KF70 ET200SP in HMI	80-81	80-81	1070-1075	1070-1073
	10.172.237.172	OW20M9TA1G01042	42	OP002-KF70-ET200SP	+OP002-KF70 ET200SP in HMI	82-83	82-83	1080-1085	1080-1083

Abb. 2.6.: PROFINET-Liste

Anlagenname ==071090500=M9TA1G01. Die Zusammensetzung geht aus folgender Aufstellung hervor:

- == vorangestelltes Trennzeichen
- 07 BMW-interne Werksnummer (Hauptgruppe)
- 10 BMW-interne Werksnummer (Untergruppe)
- 9 Technologiekennung Fahrzeugmontage
- 050 Gebäudenummer
- 0 Gebäudeteil
- = Trennzeichen
- M9 Technologie Montage
- TA1 Testanlage 1
- G01 Gruppensteuerung 1

Bedeutung		Anlagenstandort (S2) Installation/Facility location (S2)										Anlagenname (S2) Installation/Facility name (S2)										Aufstellungsart (T2) Ebene 1										Einbauart (S2) Ebene 2										Einbauart optional (S2)										BMW (S2)										BMW Erweiterung optional (S2)									
Datensatzfelder (wenn erforderlich)		N N N N N N N N N N										X X X X X X X X X X										X X N N N										A A X X X										A A X X X										A A N N a n n										o o o o o o o o									
Zusätzliche Felder (wenn erforderlich)																																																																							
Masterstring		= 0 2 4 0 9 0 5 0 0 =										M 9 T E S G 0 1 ++										S T 0 0 1 +										T L 0 0 1 .										- M M 0 1 . 0 1 -										B G 1 1																			
Bedeutung		Feststehende Zeichensatz Daten										Technologiecode										Technologiecode										Technologiecode										Technologiecode										Technologiecode										Technologiecode									
BMW Element		Symbolbit 57																																																																					

Abb. 2.7.: Anlagenkennzeichnungssystem: Masterstring

2. Planung

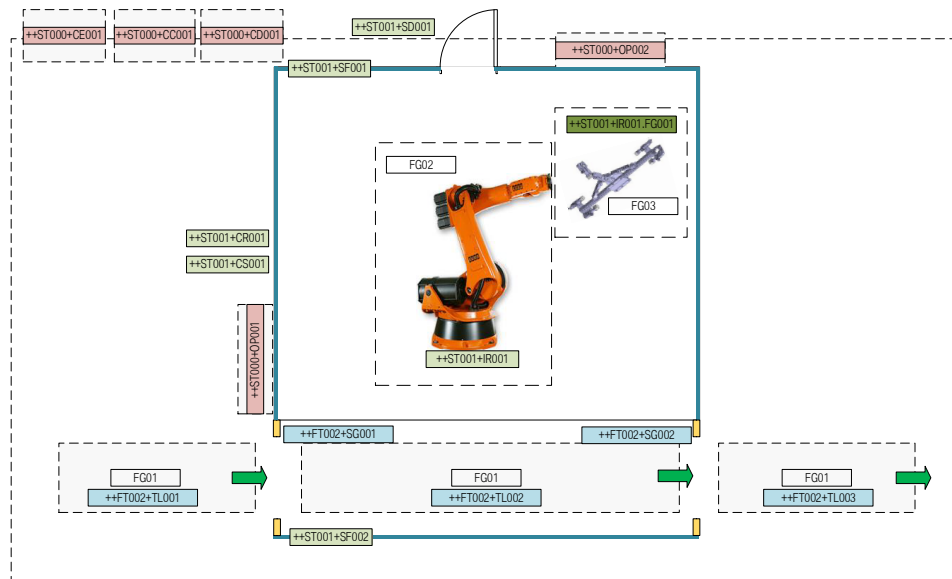


Abb. 2.8.: Anlagenkennzeichnungssystem: Überblick Gesamtanlage

Überblick Gesamtanlage

Bei der Vergabe der BMK wird ein Top-Down-Verfahren angewendet. Dabei wird zunächst eine Grobgliederung nach Aufstellungsort und Einbauort (Vgl. Abb. 2.7) vorgenommen. Für die vorliegende Roboter-Zelle ergibt sich die nachstehende Einteilung nach dem Schema ++Aufstellungsort+Einbauort:

++ST000 Schaltschränke, Panels

- +CE001 Einspeiseschrank
- +CC001 Steuerschrank
- +CD001 Antriebsschrank
- +OP001 Panel Fertigungstechnik
- +OP001 Panel Fördertechnik

++ST001 Roboterstation

- +CF001...002 Schutzzäune
- +CD001 Schutztür
- +CR001 Robotersteuerung

2. Planung

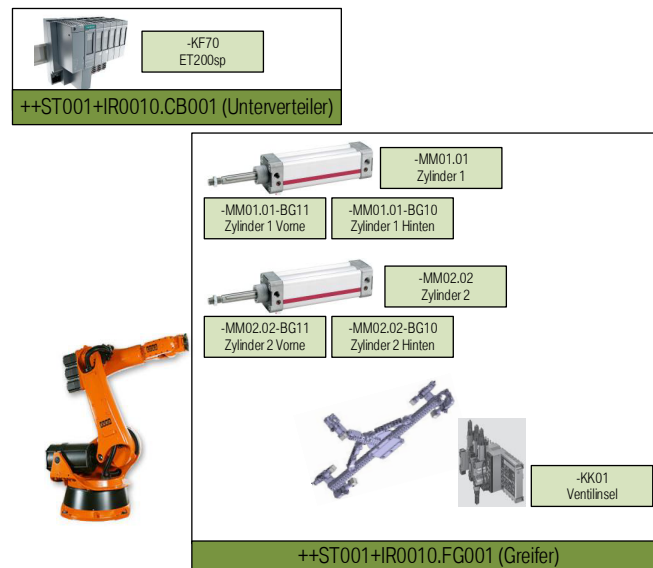


Abb. 2.9.: Anlagenkennzeichnungssystem: Detaillierte Betriebsmittelkennzeichnung

- +IR001** Industrieroboter
- ++FT002** Fördertechnik
 - +TL001** Zuführung
 - +TL002** Bearbeitung
 - +TL003** Weiterführung

Die Kennzeichen für Aufstellungsort und Einbauort weisen die gleiche Syntax auf. Vorangestellt ist jeweils ein Trennzeichen. Dieses wird gefolgt von einem Funktionskürzel, anhand dessen die Art des Aufstellungs-/Einbauortes erkennbar ist (z.B. FT: Fördertechnik, TL: Transportförderer Längs). Zudem hat jedes Kürzel eine angehängte, dreistellige laufende Nummer. Bei dieser Nummer ist zu beachten, dass die Aufstellungsorte unabhängig von ihrem Funktionskürzel hochgezählt werden. Beim Einbauort wird für jedes Funktionskürzel bei der Nummer 001 begonnen.

Im Anschluss an die Grobgliederung folgt die Kennzeichnung einzelner Betriebsmittel wie Sicherungen, SPSn, Netzteile, Zylinder und Sensoren. Eine derartige Kennzeichnung ist in Abbildung 2.9 am Beispiel des Roboters und dessen Greifer dargestellt. In diesem Fall kommen zur eindeutigen Beschreibung der Elemente auch optionale Felder zum Einsatz. Die Bezeichnung `++ST001+IR001.FG001-MM01.01-BG11` des Endlagensensors für die vordere Endlage von Zylinder 1 erklärt sich wie folgt:

2. Planung

++ST001	Roboterstation (Aufstellungsort)
+IR001	Industrieroboter (Einbauort)
.FG001	Greifer (Einbauort optional)
-M	Bereitstellung mechanischer Energie
M	Antrieb durch fluidtechnische, pneumatische Kraft
0	Antriebsrichtung (Allgemein)
1	Antrieb-Nummer
.01	Zylinder 1 (lfd. Nummer für Aktor)
-BG	Endlagensensor (BMK Sensorik)
1	lfd. Nummer Sensorik mit gleicher Abfrage
1	Eigenschaft der Positionsabfrage (vorn)

Das Anlagenkennzeichnungssystem findet Anwendung bei der Benennung aller Elemente innerhalb der Anlage. Dadurch ist es problemlos möglich die Elektrokonstruktion, Mechanik und Software zu synchronisieren. Neben der Beschriftung der Bauteile im Schaltplan, im Schaltschrank und der Baugruppen direkt in der Anlage werden die Kennzeichen auch für die Benennung von Variablen, Ein-/Ausgängen und Code-Bausteinen innerhalb der Software verwendet. Im Fall von Wartungs-, Instandsetzungs- oder Erweiterungsarbeiten erleichtert diese Vorgehensweise die Arbeit enorm. Auch Fremdfirmen finden sich schnell in einer Anlage zurecht, sofern sie den Standard kennen.

3. Test- und Präsentationsaufbau

3.1. Motivation

Das Testprojekt gilt als abgeschlossen, wenn die von BMW gestellten Anforderungen umgesetzt worden sind. Das betrifft zum Einen die formellen Anforderungen an den Anlagenentstehungsprozess und die daraus hervorgehenden Dokumente und zum Anderen die funktionalen Anforderungen an die Software. Zur Software zählen in diesem Fall das SPS-Programm, das HMI und die Anbindung an das IT-Netzwerk.

Damit die funktionalen Anforderungen abgenommen werden können, bedarf es einer Vorführung. Die Anlage entsteht nur virtuell, wodurch es keine Möglichkeit gibt, die Software unter realen Bedingungen zu testen. Für SPS und HMI bestünde die Möglichkeit, eine Simulation mit PLCSim und einer HMI-Desktop-Runtime zu erstellen und so die Funktion nachzuweisen. Selbst für die IPS-X-Systeme liefert BMW eine Simulationsumgebung mit der Bezeichnung ICOMM mit. Mit den aktuellen Versionen von ICOMM und PLCSim ist es allerdings nicht möglich, eine Verbindung untereinander herzustellen. Dadurch entsteht die Notwendigkeit, die entwickelte Software auf einer realen SPS laufen zu lassen.

3.2. Komponentenauswahl und Aufbau

Als Grundlage für die Hardware-Projektierung dient eine Bauteil-Liste, die Bestandteil der Standard-Dokumentensammlung ist. Diese Liste liegt in Form einer sehr umfangreichen Excel-Arbeitsmappe vor. Darin enthalten ist eine Vielzahl an Bauteilen aus den Domänen Elektrotechnik, Steuerungstechnik, Mechatronik und Mechanik, auf die bei der Projektierung vorzugsweise zurückgegriffen werden soll. Abweichungen von der Liste dürfen nur in Absprache mit dem verantwortlichen BMW-Projektleiter stattfinden.

Bei dem Testaufbau werden die technischen Anforderungen, welche sich aus dem Umfang der Aufgabe ergeben, begleitet von nicht technischen Anforderungen, die seitens der OSB gelten. Da es sich um ein Non-Profit-Projekt handelt und eine Wiederverwendbarkeit der Hardware für zukünftige Projekte nicht garantiert ist, sind die Hardware-Kosten so niedrig wie möglich zu halten. Das hat zur Folge, dass für den Testaufbau lediglich gebrauchte Baugruppen verwendet werden. Tabelle 3.1 zeigt eine Auswahl an steuerungstechnischen Baugruppen, die der Realisierung des Testaufbaus dienen.

3. Test- und Präsentationsaufbau

Tabelle 3.1.: Bauteilliste Testaufbau

Gerät	Bezeichnung	Artikelnummer	Anzahl
S7-300	SIMATIC S7-300, Profilschiene L=480mm	6ES7390-1AE80-0AA0	1
S7-300	PS 307 Stromversorgung 5A	6ES7307-1EA01-0AA0	1
S7-300	SIMATIC S7-300 CPU317F-2 PN/DP	6ES7317-2FK14-0AB0	1
S7-300	IP-20 Switch 8P SCALANCE XF208	6GK5208-0BA00-2AF2	1
ET 200SP	IM 155-6 PN ST inkl. Servermodul inkl. Busadapter	6ES7155-6AA00-0BN0	1
ET 200SP	Base-Unit BU15-P16+A0+2D, BU-Typ A0	6ES7193-6BP00-0DA0	3
ET 200SP	digitales Eingangsmodul, DI 8x DC 24V	6ES7131-6BF00-0BA0	1
ET 200SP	digitales Ausgangsmodul, DQ 8x 24VDC/0,5A	6ES7132-6BF00-0BA0	1
ET 200SP	sicheres digitales Eingangsmodul, F-DI 8x 24VDC HF	6ES7136-6BA00-0CA0	1

Zusätzlich zu den aufgelisteten Baugruppen werden Bauelemente aus der Elektrotechnik benötigt, um eine Verbindung der Baugruppen untereinander herzustellen. Dazu gehören:

Tabelle 3.2.: Elektrotechnisches Zusatzmaterial

lfd. Nr.	Bezeichnung	Variante	Anzahl
1	NOT-HALT Taster mit Meldeleuchte, zweikanalig		1
2	Reihenklemmen	1,5 mm ²	15
3	Reihenklemmen Abschlussplatten		2
4	Klemmleisten Endklemmen		4
5	Aderleitung schwarz	1,0 mm ²	10 m
6	Aderleitung blau	1,0 mm ²	10 m
7	Netzwerkkabel mit RJ45 Stecker	1 m	3
8	Hutschiene	35 mm x 7,5 mm	0,5 m
9	Verdrahtungskanal	45 mm x 45 mm	2,5 m
10	Anschlussleitung mit Schutzkontakt-Stecker	3 x 1,5 mm ²	1,5 m

Damit ein ganzheitlicher Präsentationsaufbau entsteht, werden alle Komponenten auf einem Testgestell ähnlich dem Modell in Abbildung 3.1 montiert. Dafür wird ein simpler Aufbau, bestehend aus einem Aluminiumprofil-Rahmen, einer verzinkten Lochraasterplatte und den darauf montierten Verdrahtungskanälen und Hutschienen vorgesehen.

Der Testaufbau verfügt über eine Einspeise-Klemmleiste, die mittels Anschlussleitung an das 230 V-Netz angeschlossen ist. Die Spannungsversorgung der Automatisierungskomponenten erfolgt über ein Netzteil der Firma Siemens (siehe 3.1). Die 24 V Gleichspannung wird schließlich mithilfe einer zweiten Klemmleiste auf die Verbraucher verteilt. Die beiden Kanäle des NOT-HALT-Tasters werden direkt auf die sichere Eingangsbaugruppe gelegt.

Für die Vernetzung der CPU und der ET 200 kommt ein SCALANCE 8-Port Switch (nicht in der Darstellung vorhanden) zum Einsatz. Dieser ist durch den TMO V1 Stan-

3. Test- und Präsentationsaufbau

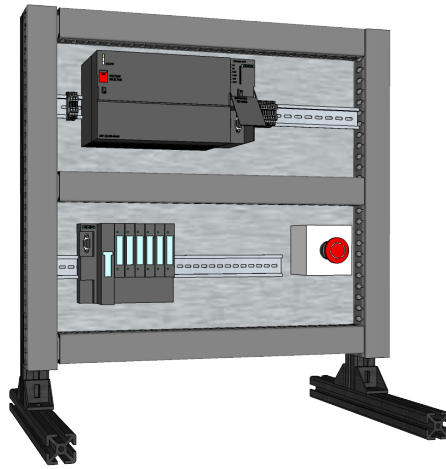


Abb. 3.1.: Präsentationsaufbau: 3D-Prinzipdarstellung

dard vorgeschrieben und bietet den Vorteil, den Aufbau sehr einfach mit zusätzlichen PROFINET-Teilnehmern zu erweitern. Zudem kann ein freier Port zum Anschluss des Programmier-Laptops genutzt werden.

4. Anlagensoftware

4.1. Vorgehensweise

Wie für die meisten Entwicklungsschritte im Anlagenentstehungsprozess enthält der TMO V1 Standard auch für die Anlagensoftware eine Projektvorlage. Diese besteht im Wesentlichen aus zwei Teilen, einem Step7-Projekt, welches das Programm für die SPS enthält und einem TIA V13 SP1 Projekt für die Visualisierung. Die Projekte sind so aufgebaut, dass sie nahezu alle im Standard vorhandenen Bausteine und Konzepte implementieren. Die Nutzung dieser Vorlagen ist dabei nicht optional sondern obligatorisch. Wie in anderen Domänen des Standards ist auch bei der Anlagensoftware das Ziel, eine einheitliche Basis für ein breites Feld von Maschinen zu schaffen, auch wenn diese von unterschiedlichen Herstellern entwickelt werden.

Die Bearbeitung der Anlagensoftware orientiert sich am Leitfaden [2] zur Entwicklung der S7-Software. Als Voraussetzung dienen der Ergebnisse der Planungsphase, sowie eine vorhandene Feinspezifikation. Folgende Vorgehensweise wird von BMW empfohlen:

1. Vorbereitung
 - a) Erstellen der Hardware-Konfiguration
 - b) Erarbeiten der Symboltabellen
 - c) Entfernen nicht benötigter Software Bausteine
2. Anbindung der HMI-Panels und Konfiguration der Funktionsgruppen (FG)
3. Konfiguration des Meldesystems
4. Anlagendiagnose in der Steuerung
5. Entwicklung der Safety-Software
6. Entwicklung der Logik für Aktorik und Sensorik
7. Ablaufprogrammierung in Schrittketten
8. Anbindung IT-Systemen

In den folgenden Abschnitten werden exemplarisch die Punkte 1a, 5 und 8 beschrieben.

4. Anlagensoftware

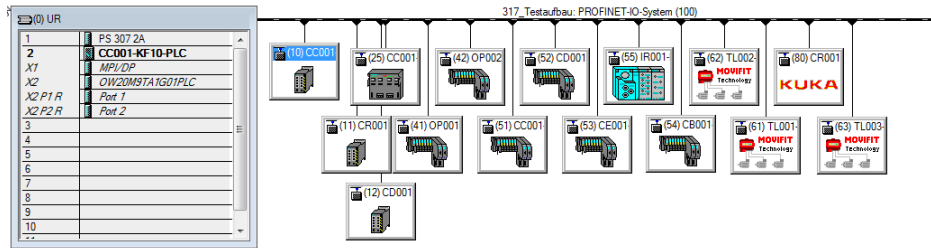


Abb. 4.1.: Hardware-Konfiguration

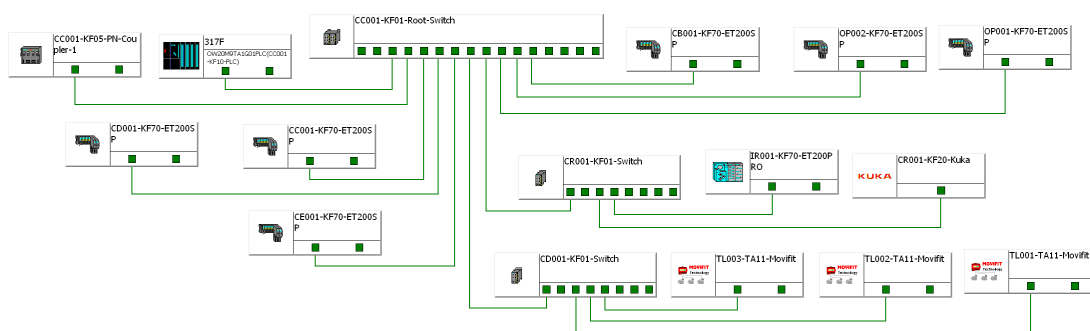


Abb. 4.2.: PROFINET Topologie

4.2. Hardware-Konfiguration

Basierend auf der in Abschnitt 2.3 beschriebenen PROFINET Liste wird die Hardware-Konfiguration vorgenommen. Zunächst wird die verwendete Steuerung eingefügt, benannt und mit der projektierten IP-Adresse versehen. Anschließend werden nach und nach alle Teilnehmer aus der PROFINET-Liste hinzugefügt, adressiert und die entsprechenden Ein-/Ausgangsbereiche zugeordnet. Danach wird die Topologie des Netzwerks konfiguriert. Das heißt, jedem Port, sofern er verschaltet werden soll, wird der Port zugeordnet, mit dem er verbunden ist. Dieses Vorgehen bietet erweiterte Diagnose-Möglichkeiten in der CPU, wenn Teilnehmer am Netz ausfallen oder ausgetauscht werden. Für das Testprojekt entsteht die in Abbildung 4.1 dargestellte Hardware-Konfiguration. Die zugehörige PROFINET-Topologie ist in Abbildung 4.2 zu sehen.

Da die verwendete Steuerung mit ihrer integrierten Safety-CPU sicherheitstechnische Peripherie auswerten und steuern können soll, ist es notwendig, die entsprechenden F-Parameter in der Hardware-Konfiguration einzustellen. Eine Beschreibung aller relevanten Daten sowie die Vorgehensweise beim Anpassen sind Teil der TMO V1 Dokumentation [1].

4. Anlagensoftware

	FG= Funktionsgruppe LK = Lastkreis	Abschaltbereiche			
		FG01	FG02	FG03	
		LK01			
Nothalt-Organ		TL001-TL003 A 1000.0	IR001 Roboter Nothaltkreis	IR001 Roboter Schutztürkreis	IR001.FG001 A 1000.3
	Not-Halt OP001	X	X		X
	Not-Halt OP002	X	X		X
	MSL Einlauf Zelle SG001	X	X		X
	MSL Auslauf Zelle SG002	X	X		X
	Not-Halt Roboter PHG IR001	X	X		X
	Schutztüre 1 SD001	X		X	X

Abb. 4.3.: Safety-Matrix

4.3. Safety-Software

Als Grundlage für die Safety-Software dient die Safety-Matrix (Vgl. Abb. 4.3). In dieser steht beschrieben, welches Nothalt-Organ welchen Lastkreis und welche Funktionsgruppe betrifft. Für den vergleichsweise geringen Umfang der Anlage im Testprojekt ist die Matrix recht simpel. Hier wirkt jedes Nothalt-Organ auf jede Funktionsgruppe. Bei komplexeren, flächenmäßig weitläufigen Anlagen lautet die Vorgabe, dass Nothalt-Organen jeweils nur Funktionsgruppen betreffen, die sich in unmittelbarer Nähe befinden. Speziell bei Nothalt-Tastern gilt, dass nur Funktionsgruppen betroffen sind, die sich im Sichtbereich der betätigenden Person befinden.

Zur Realisierung der Sicherheitsfunktionen im Testprojekt werden Bausteine aus der Projektvorlage verwendet. Das darin enthaltene Safety-Programm umfasst wesentlich mehr Peripherie als für das Testprojekt vorgesehen. Abbildung 4.4 verdeutlicht den Unterschied anhand der Not-HALT-Behandlung. Anstelle der vier Not-HALT-Organen aus der Projektvorlage werden hier lediglich zwei verarbeitet. Jedem Not-HALT Taster ist ein Netzwerk zugeordnet, in dem die zugehörigen sicheren digitalen Eingänge verschaltet sind. In weiteren Bausteinen wird die entsprechende Peripherie für Schutz-Türen, Lichtgitter, Roboter, Lastkreise etc. verarbeitet.

4.4. Anbindung IT-Systemen

Für die Anbindung der IT-Systeme (IPS-X System) ist die Konfiguration mehrerer Bausteine notwendig. Hardware-seitig werden die IT-Systeme über die Onboard-Schnittstelle der CPU angebunden. Software-seitig wird in einem Funktionsbaustein (FB61-FB_IE_VERB) jede einzelne Verbindung separate eingerichtet (s. Abb. 4.5).

Im ersten Netzwerk wird die CPU gewählt, indem der entsprechende Zahlenwert in eine temporäre Variable geladen wird. Diese wird in den nachfolgenden Netzwerken für jede einzelne Verbindung wiederverwendet. Im Rahmen des Testprojektes sind drei

4. Anlagensoftware

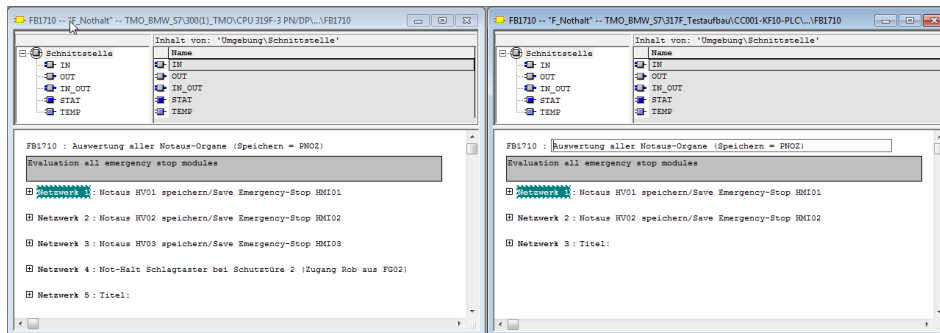


Abb. 4.4.: Vergleich Not-HALT Behandlung: Projektvorlage (li.), Testprojekt (re.)

Verbindungen herzustellen, bei denen die SPS jeweils der passive Verbindungspartner ist. Es handelt sich allen drei Fällen um RFC1006-Verbindungen¹. Damit eine solche Verbindung erfolgreich aufgebaut werden kann, sind die IP-Adressen der Partner und deren TSAP notwendig. Als TSAP der SPS dient der Anlagenname aus dem Anlagenkennzeichen (siehe Abschnitt 2.4). Der TSAP des jeweiligen IPS-X System ist *hart* in den FB61 codiert und wird mit folgenden Lade- und Trasfer-Anweisungen in den Netzwerken 2...4 in die statischen Daten des Bausteins geladen. Beispiel IPS-T-System:

```
1      L      8
2      T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id_len
3      L      'M'
4      T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[1]
5      L      'O'
6      T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[2]
7      L      'N'
8      T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[3]
9      L      'T'
10     T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[4]
11     L      'I'
12     T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[5]
13     L      'P'
14     T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[6]
15     L      'S'
16     T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[7]
17     L      'T'
18     T      "DI_FB61".IPS_T_01.Connect_Data.rem_tsap_id[8]
```

¹RFC 1006 mit dem Titel "ISO Transport Service on top of the TCP" (ISO Transportdienst über TCP) ist eine Protokoll-Erweiterung für das TCP-Protokoll. Hierbei werden zusätzlich zu den TCP Daten weitere Informationen zwischen den Teilnehmern übertragen, um bestimmte Dienste für den Anwender erbringen zu können (ISO-Dienste als Erweiterung zu TCP) [3]

4. Anlagensoftware

```
FB61 : Aufruf aller Verbindungen für T-Kommunikation Onboard
/ Calling all connections for T-communicatio onboard

Netzwerk 1 : Auswahl der fixen Parameter für CPU-Typen 31x oder 41x
Auswahl der fixen Parameter für CPU-Typen 31x oder 41x / Selection of the fix
members for CPU-Types 31x or 41x
1 = CPU151-8PN
2 = CPUs 315/317(F)-2 PN/DP
3 = CPUs 319(F)-3 PN/DP
5 = CPUs 41x(F)-3 PN/DP

L      2
T      #cpu_typ          #cpu_typ          -- 2=CPU317 / 5=CPU416

Netzwerk 2 : Verbindung 1: IPS_T / Connection 1: IPS_T
Netzwerk 3 : Verbindung 2: IPS_L / Connection 2: IPS_L
Netzwerk 4 : Verbindung 3: IPS_Q über KM direkt oder über LPA / Con. 3 IPSQ
Netzwerk 5 : Titel:
BE
```

Abb. 4.5.: SPS-seitige Konfiguration der TCP-Verbindungen

Schließlich wird mit dem Aufruf eines Multiinstanz-Funktionsbausteins für jedes IPS-X System separat die Verbindung aufgebaut. Beispiel IPS-T-System:

```
1 // Verbindung auf die integrierte IE-Schnittstelle laden und aufbauen
2 //load and connect connection to the integrated interface
3 CALL #IPS_T_01 // Funktion Verbindungsparameter laden
4 connect:="DI_FB61".IPS_T_01.connect // Verbindung aufbauen
5 CPU_Typ:=#cpu_typ
```

Nach erfolgreicher Konfiguration ist die SPS in der Lage, Daten mit den IPS-X-Systemen auszutauschen. Welche Daten ausgetauscht werden sollen und wie die Kommunikation im Einzelnen vonstatten geht, wird in Kapitel 5 behandelt.

5. Nutzung der IPS-X Systeme

5.1. IPS-L

5.1.1. Schnittstellenkontrakt

Das IPS-L-Netz ist Bestandteil des übergeordneten IT-Netzes der Anlage. Es dient zum Austausch von Daten die die Logistik-Steuerung der Auftragsdaten betreffen. Dazu verwaltet das IPS-L-System Karossen und Anbauteile und stellt diese Informationen entsprechend der Anlage bereit. Dabei entspricht die Reihenfolge der Daten, die vom IPS-L-System an die Steuerung übermittelt werden, der Sequenz an Fahrzeugen, die in der Anlage bearbeitet werden. Somit können die automatischen Bearbeitungsprozesse in Abhängigkeit des vorliegenden Fahrzeuges angepasst werden.

Im Schnittstellenkontrakt mit BMW wurde zu Projektbeginn festgelegt, dass es sich bei der Schlüsselnummer der Datensätze vom IPS-L-System um die Ordernummer handelt. Zudem wurden gemäß Abbildung 5.1 folgende vier Telegramme für den Datenaustausch spezifiziert:

Einzeldatensatz-Anforderung Die SPS sendet ein Einzel-Anforderungstelegramm an das IPS-L-System und dieses antwortet mit einem einzelnen Datensatz

Blockdatensatz-Anforderung Die SPS sendet ein Block-Anforderungstelegramm an das IPS-L-System und dieses antwortet mit einem Block-Datensatz dessen Größe auf fünf Einzeldatensätze festgelegt wurde

IPS-L Verbindungsüberwachung Die SPS sendet einen Ping an das IPS-L-System und dieses antwortet mit einer Ping-Antwort

SPS Verbindungsüberwachung Das IPS-L-System sendet einen Ping an die SPS und diese antwortet mit einer Ping-Antwort

5.1.2. Realisierungskonzept

Die Daten aus dem IPS-L-System sollen in der Typverwaltung zu Anwendung kommen. Wie in Abschnitt 1.2 beschrieben, soll anhand des in der Station befindlichen Fahrzeugtyps ein entsprechendes Bearbeitungsprofil ausgewählt werden. Zur Identifikation des Fahrzeugtypes wird dessen Entwicklungsserie (z.B. E-90, E-87) verwendet.

5. Nutzung der IPS-X Systeme

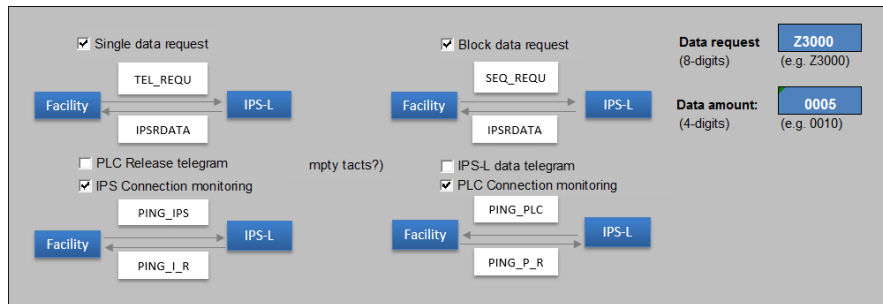


Abb. 5.1.: IPS-L Schnittstellenkontrakt

Für die Typverwaltung ist ein eigener Funktionsbaustein (FB) `FB_TYPE_ADMIN` vorgesehen, der zyklisch im Hauptprogramm (OB1) der SPS aufgerufen werden soll. Dessen Logik ist konzeptionell in dem Sequenzdiagramm in Abbildung 5.2 dargestellt und soll auf Basis dieser Vorlage umgesetzt werden. Folgende Funktionalität ist dabei zu implementieren. Zunächst wird auf die Meldung der Fördertechnik gewartet, dass der Werkstückträger (WT) in Bearbeitungsposition ist. Außerdem wird geprüft, ob die Bearbeitung nicht aktiv ist. Sind beide Bedingungen erfüllt, wird die Nummer des WT's von Förderelement +TL002 aus dem Datenbaustein (DB) `DB_WT_NR` gelesen. Anhand dieser Nummer wird anschließend aus `DB_WT_DA` die auf dem WT befindliche Ordnernummer ermittelt. Mithilfe dieser wird danach die zur Ordnernummer gehörige Entwicklungsreihe aus dem IPS-L DB `DB_IPSL_SEQU` gelesen. Der DB `DB_TYPE_ADMIN` enthält schließlich die Zuordnung der Bearbeitungsprofile zu den Entwicklungsreihen. Das zugehörige Bearbeitungsprofil wird also ausgewählt und an die Roboterschnittstelle `DB_ROB_ITF` übergeben. Nach erfolgter Übergabe erhält der Roboter die Freigabe zur Bearbeitung, er meldet zurück, sobald die Bearbeitung aktiv ist.

5.1.3. Implementierung

Für die Implementierung der beschriebenen Logik werden neben dem Funktionsbaustein `FB_TYPE_ADMIN`, der die Logik enthält, zahlreiche Hilfsfunktionen (FCs), der Datenbaustein `DB_TYPE_ADMIN` sowie ein neuer Datentyp (UDT) zum Darstellen des Bearbeitungsprofils benötigt. Folgende Auflistung beinhaltet die nötigen Objekte und beschreibt deren Zweck in Anlehnung an Abbildung 5.2. Zudem enthält die Auflistung die Implementierung der Objekte in der Sprache `SCL`. Alle anderen im Sequenzdiagramm aufgeführten Datenbausteine, welche nicht beschrieben sind, sind bereits in der STEP7-Projektvorlage vorhanden und können ohne Modifikation verwendet werden.

`UDT_PROCESS_PROFILE` ist ein benutzerdefinierter Datentyp, der als Repräsentation für ein Bearbeitungsprofil dient. Die Entwicklungsreihe wird einem Feld von vier Ele-

5. Nutzung der IPS-X Systeme

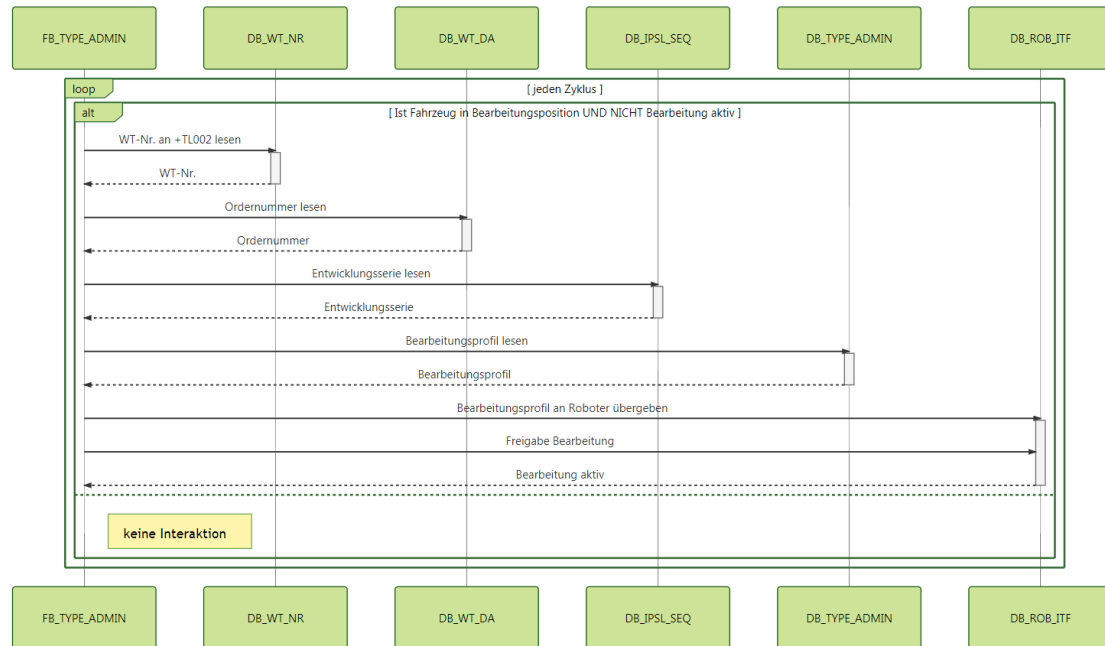


Abb. 5.2.: Sequenzdiagramm: Verwendung der IPS-L-Daten zur Typverwaltung

menten des Typs CHAR dargestellt. Das entspricht exakt der Repräsentation, wie sie im Datenbaustein DB_IPSL_SEQU verwendet wird. Daneben enthält der Datentyp für jeden anzusteuern den Zylinder jeweils eine boolesche. Weißt eine useCylX-Variable den Wert TRUE auf, wird der zugehörige Zylinder bei der Bearbeitung des Fahrzeugs angesteuert.

```

1 TYPE UDT_PROCESS_PROFILE
2     STRUCT
3         series : ARRAY [0..3] OF CHAR; // ID of development series
4         useCyl1 : BOOL; // use cylinder 1?
5         useCyl2 : BOOL; // use cylinder 2?
6     END_STRUCT
7 END_TYPE
  
```

DB_TYPE_ADMIN: ist ein Datenbaustein, der die verschiedenen Bearbeitungsprofile als ein Feld mit zehn Elementen vom Typ UDT_PROCESS_PROFILE enthält. Die enthaltenen Daten werden über das HMI eingepflegt und geändert.

```

1 DATA_BLOCK DB_TYPE_ADMIN
2     STRUCT
  
```

5. Nutzung der IPS-X Systeme

```
3      ProcessProfiles : ARRAY [1..10] OF UDT_PROCESS_PROFILE;  
4  END_STRUCT  
5 BEGIN  
6  
7 END_DATA_BLOCK
```

FC_GET_WT_NO(conveyorElement:INT):INT liest die WT-Nummer aus DB_WT_NR an einem durch conveyorElement spezifizierten Förderelement und gibt sie als Ganzzahl zurück. Für conveyorElement < 0 wird -1 zurückgegeben, um damit einen Fehler zu signalisieren.

```
1 FUNCTION FC_GET_WT_NO : INT  
2   VAR_INPUT  
3     conveyorElement: INT;  
4   END_VAR  
5  
6   IF conveyorElement > 0 THEN  
7     FC_GET_WT_NO := DB_WT_NR.EL[conveyorElement].WT_Nr;  
8   ELSE  
9     FC_GET_WT_NO := -1;  
10  END_IF;  
11 END_FUNCTION
```

FC_GET_OR(wtNumber:INT, orderNumber:CHAR[8]):VOID liest die Ordnernummer aus dem Datenbaustein DB_WT_DA von einem durch wtNumber spezifizierten Werkstückträger und gibt diese über den In/Out-Parameter orderNumber als ein Feld von acht Elementen des Typs CHAR zurück. Die Funktion selbst ist vom Typ VOID. Die Einschränkungen für Rückgabewerte seitens des SIMATIC Managers erlauben keine Arrays. Das Format für die Ordnernummer entspricht dem, welches im DB DB_IPSL_SEQU verwendet wird und erleichtert damit die weitere Verwendung.

```
1 FUNCTION FC_GET_OR : VOID  
2   VAR_INPUT  
3     wtNumber : INT;  
4   END_VAR  
5  
6   VAR_IN_OUT  
7     orderNumber : ARRAY [0..7] OF CHAR;  
8     orMask AT orderNumber : ARRAY [0..7] OF BYTE;  
9   END_VAR  
10  
11  VAR_TEMP  
12    ii : INT;          // loop counter  
13    byteIndex: INT;  
14  END_VAR  
15  
16  CONST  
17    WT_DATA_LENGTH := 2 + 78;  // bytes  
18    WT_DATA_OFFSET := 80;      // bytes  
19    OR_OFFSET := 2;           // offset of OR in UDT_WT_Daten
```

5. Nutzung der IPS-X Systeme

```
20     END_CONST
21
22     byteIndex := WT_DATA_OFFSET
23               + (wtNumber * WT_DATA_LENGTH)
24               + OR_OFFSET;
25
26     FOR ii := 0 TO 7 BY 1 DO
27         orMask[ii] := DB_WT_DA.DB[byteIndex + ii];
28     END_FOR;
29 END_FUNCTION
```

FC_GET_SERIES(orderNumber:CHAR[8], series:CHAR[4]):VOID liest unter Angabe der Ordnernummer als Parameter orderNumber die Entwicklungsserie aus dem Datenbaustein DB_IPSL_SEQU und gibt diese als In/Out-Parameter series als Feld von vier Elementen des Typs CHAR zurück.

Beim Lesen der IPS-L Daten bedient sich die Funktion des im Datenbaustein DB_IPSL_VERW vorhandenen *Zeiger 8*. Dieser ist vom TMO V1 Standard für den einfachen Zugriff auf das IPS-L Sequenzregister definiert und zeigt auf den Datensatz des aktuellen Fahrzeuges. Der Zeigerwert wird von einer separaten Funktion verwaltet. Bevor die Entwicklungsserie aus den IPSL-Daten gelesen wird prüft die Funktion, ob die angeforderte Ordnernummer derjenigen entspricht, welche durch den *Zeiger 8* referenziert wird. Ist dies nicht der Fall gibt die Funktion einen Fehler zurück.

```
1 FUNCTION FC_GET_SERIES : INT
2     VAR_IN_OUT
3         orderNumber : ARRAY [0..7] OF CHAR;
4         series : ARRAY [0..3] OF CHAR;
5     END_VAR
6
7     VAR_TEMP
8         ii : INT;           // loop counter
9         ipslPointer : INT;  // pointer to current IPSL data set
10        blkMovResult : INT;
11    END_VAR
12
13    ipslPointer := DB_IPSL_VERW.ZEIG_8;
14
15    IF FC_DATA_EQUAL( in1:=DB_IPSL_SEQU.SEQ[ipslPointer].Key,
16                    in2:=orderNumber)
17    THEN
18        // order number of current IPSL dataset is equal to requested orderNumber
19        blkMovResult :=
20            BLKMOV( SRCBLK := DB_IPSL_SEQU.SEQ[ipslPointer].SERIES,
21                  DSTBLK := series);
22        FC_GET_SERIES := 0;    // return success
23    ELSE
24        FC_GET_SERIES := -1;   // return error
25    END_IF;
26
27 END_FUNCTION
```

FC_GET_PROCESS_PROFILE(series:CHAR[4]:UDT_PROCESS_PROFILE liest unter Angabe der Entwicklungsserie als Parameter **series** das zugehörige Bearbeitungsprofil aus dem Datenbaustein **DB_TYPE_ADMIN** und gibt es als In/Out-Parameter **processProfile** des Typs **UDT_PROCESS_PROFILE** zurück.

```
1 FUNCTION FC_GET_PROCESS_PROFILE : INT
2   VAR_IN_OUT
3     series : ARRAY [0..3] OF CHAR;
4     processProfile : UDT_PROCESS_PROFILE;
5   END_VAR
6
7   VAR_TEMP
8     ii : INT;          // loop counter
9   END_VAR
10
11  FOR ii:=1 TO 10 BY 1 DO
12    IF FC_DATA_EQUAL( in1:=series,
13                     in2:=DB_TYPE_ADMIN.ProcessProfiles[ii].series)
14    THEN
15      processProfile := DB_TYPE_ADMIN.ProcessProfiles[ii];
16      FC_GET_PROCESS_PROFILE := 0;          // return success
17      RETURN;
18    END_IF;
19  END_FOR;
20
21  FC_GET_PROCESS_PROFILE := -1;              // return error
22 END_FUNCTION
```

5.2. IPS-T

5.2.1. Schnittstellenkontrakt

Es handelt sich dabei um das Technik-Netz. Mit diesem werden Daten unterschiedlicher Art ausgetauscht. Zu den wichtigsten Daten gehören die Störmeldungen, welche die SPS an das IPS-T-Netz übermittelt. Zusätzlich werden Daten zu Zählerständen, Füllständen und Taktzeiten ausgetauscht. Dabei agiert die SPS nicht nur als Sender sondern auch Empfänger. So können mittels IPS-T beispielsweise Sollwerte für den Prozess vorgegeben, bit-getriggerte Events in der Anlage ausgelöst oder Schichtwechsel-Signale an die Steuerung gesendet werden.

Zum Generieren und Interpretieren der Daten bedienen sich die Kommunikationspartner eines im TMO-V1 definierten Protokolls. Seitens der SPS ist dieses in Form eines STRUCT in den Sende- und Empfangs-DBs hinterlegt (siehe Abb. 5.3). Das Befüllen der STRUCTs in den DBs geschieht mittels BMW-eigener Funktionen und Funktionsbausteinen, die wiederum separate DBs als Datenbasis nutzen. Die Aufgabe des SPS-Entwicklers besteht darin die Datenbasis mit den relevanten Werten zu befüllen. Im vorliegenden Projekt ist der Austausch von insgesamt sechs Zählerständen vorgesehen, die die folgenden Daten enthalten sollen:

5. Nutzung der IPS-X Systeme

+10.0	Empfang	STRUCT	Empfangspuffer von IPS-T / Receipt buffer of IPS-T
+0.0	Source	ARRAY[1..8]	Quelle IPS-T / Source IPS-T
+1.0		CHAR	
+8.0	Destin	ARRAY[1..8]	AKZ der Anlage / AKZ of the system
+1.0		CHAR	
+16.0	Telname	ARRAY[1..8]	Telegramm-Kennung / Telegram identifier
+1.0		CHAR	
+24.0	Ackmode	ARRAY[1..4]	Quittungsmodus / Acknowledge mode
+1.0		CHAR	
+28.0	Timestamp	ARRAY[1..14]	Zeitstempel / Timestamp
+1.0		CHAR	
+42.0	Len	ARRAY[1..4]	Telegrammlänge in Byte / Length of telegram in byte
+1.0		CHAR	
+46.0	Daten	ARRAY[46..2048]	Datenbereich / Data area
+1.0		CHAR	
=2050.0		END_STRUCT	

Abb. 5.3.: STRUCT: IPS-T Protokoll

IPS-T → SPS Sollwert Taktzeit (Einheit: 1/10 s)
IPS-T → SPS Schichtziel (Einheit: Autos)
SPS → IPS-T Gesamtzahl Autos (Einheit: Autos)
SPS → IPS-T Aktuelle Taktzeit (Einheit: 1/10 s)
SPS → IPS-T Gesamtzahl F20 (Einheit: Autos)
SPS → IPS-T Gesamtzahl F45 (Einheit: Autos)

5.3. IPS-Q

Das Qualitätsnetz IPS-Q hat im Rahmen des Testprojektes keine Relevanz und wird aus diesem Grund nicht konzeptioniert.

Internetquellen

- [1] CONTACT, PHOENIX: *PROFIsafe*. https://www.phoenixcontact.com/online/portal/de?1dmy&urile=wcm:path:/dede/web/main/products/technology_pages/subcategory_pages/Safety/916ff0d6-5046-4fa4-9a36-00e3abbeca60/916ff0d6-5046-4fa4-9a36-00e3abbeca60. – Zugriff: 04.08.2017
- [2] FELDBUSSE.DE: *Profinet*. <http://www.feldbusse.de/Profinet/profinet.shtml>. – Zugriff: 04.08.2017
- [3] SIEMENS: *Was ist RFC1006 und wozu brauche ich diesen Dienst?* <https://support.industry.siemens.com/cs/document/15048962/was-ist-rfc1006-und-wozu-brauche-ich-diesen-dienst-?dti=0&lc=de-WW>. – Zugriff: 01.11.2017

Sonstige Quellen

- [1] TMO, BMW: *Description S7-Safety Software*. Internes Dokument, . – Version 1.2
- [2] WEGERER, Alois: *Beschreibung der TMO_BMW_S7 Software V1 der Technologie Montage*. Internes Dokument, . – Version 0.3

A. Steuerungskonzept