



2023

Prozessbusleitsysteme

30. Juni 2023

Florian Tietjen 2519584

PF: Emily Antosch 2519935

Karl Döring 2519590

Abbildungsverzeichnis	3
1 Bussysteme	4
1.1 Einführung	4
1.2 Lernziele	4
1.3 Kurzbeschreibung der Hardware und des Bussystems	4
1.3.1 Anlage	4
1.3.2 Steuerungshardware	5
1.3.3 Bussysteme PROFIBUS DP	5
1.4 Vorbereitende Arbeiten	6
1.5 Vorbereitung	6
1.5.1 OB100	6
1.5.2 OB1	6
1.6 Aufbau	7
1.7 Aufgabe 5: Inbetriebnahme	7
1.8 Aufgabe 6: ProfiTrace	7
1.9 Aufgabe 7: Busfehler	8
1.9.1 Aufgabe 7.1: Ausfall einer dezentralen Peripherie	8
1.9.2 7.2: Abschlusswiderstand	8
1.10 Aufgabe 8: Einbinden eines neuen DP-Moduls	9
1.10.1 Aufgabe 8.1: Einbinden ohne Hardwarekonfiguration	9
1.10.2 Aufgabe 8.2: Einbinden mit Hardwarekonfiguration	9
1.10.3 Aufgabe 8.3: ProfiTrace-Ausgabe	9
1.10.4 Aufgabe 8.4: Zugriff auf die Baugruppe im Programmablauf	9
1.11 Fazit	10

ABBILDUNGSVERZEICHNIS

1.1	Telegrammaufbau	6
1.2	Der Aufbau eines PROFIBUS DP Telegramms	6
1.3	ProfiTrace-Ausgabe nach der Inbetriebnahme	8
1.4	ProfiTrace-Ausgabe mit mehreren Teilnehmern	9

KAPITEL 1 BUSSYSTEME

1.1 Einführung

In diesem Laborversuch steht die Steuerung eines Transportbands mit einer Laufrichtungsumkehrung für zylindrische Werkstücke im Fokus. Um dieses Ziel zu erreichen, wird ein speicherprogrammierbares Steuerungssystem (SPS) eingesetzt, das über das PROFIBUS DP Bussystem mit der Sensorik und Aktorik der Anlage verbunden ist. Dabei liegt der Schwerpunkt auf der Untersuchung der entstehenden Kommunikation auf dem Bus sowie dem Verhalten des Systems bei verschiedenen Arten von Fehlern, die auftreten können. Im Zuge des Versuchs wird das Programm zur Steuerung der Anlage implementiert und die daraus resultierenden Beobachtungen und Interpretationen dokumentiert.

1.2 Lernziele

Die Lernziele dieses Labors umfassen das Erlangen eines Grundverständnisses von PROFIBUS DP und den dazugehörigen Baugruppen sowie die Konfiguration und den Betrieb eines realen Bussystems an einer konkreten Anlage. Der Aufbau und die Inhalte von PROFIBUS-Telegrammen sollen verstanden und einfache Fehlersituationen diagnostiziert werden können. Ein weiteres Ziel ist das Einbinden eines zusätzlichen Busteilnehmers in das Bussystem.

1.3 Kurzbeschreibung der Hardware und des Bussystems

1.3.1 Anlage

Ein Transportband ermöglicht den Transport von zylinderförmigen Objekten. Die Position der Objekte wird mithilfe von Lichtschranken am Anfang (LS1) und vor dem Vereinzeler (LS4) erkannt. Der Vereinzeler ermöglicht die Separierung der Werkstücke am Ende des Bands, während eine weitere Lichtschranke ihre Erkennung gewährleistet. Zudem wird die Höhe der Werkstücke mittels eines Entfernungsmessers bestimmt. Für diesen Versuch werden lediglich das Band selbst und die Lichtschranken LS1 und LS4 verwendet. Die logische Zuordnung der Hardware zu den Ein- und Ausgängen der SPS ist in einer Tabelle dargestellt. Die Steuerung der Bandgeschwindigkeit erfolgt analog, indem die Steuerung über AW260 einen festen Digitalwert im Bereich von 0 bis 27648 sendet. Die Analogausgabebaugruppe wandelt diese Digitalwerte in ein analoges Spannungssignal um.

1.3.2 Steuerungshardware

Die Steuerung basiert auf einer Siemens SPS vom Typ SIMATIC S7-1500. Sie besteht aus einer Stromversorgung, der Zentralbaugruppe CPU 1516-3 DP/PN, die sowohl den Anschluss von PROFIBUS DP als auch von PROFINET ermöglicht, sowie mehreren Signalbaugruppen mit analogen und digitalen Ein- und Ausgängen. Die Kommunikation zwischen der CPU und der Anlage erfolgt über ein dezentrales Interfacemodul vom Typ ET200S mit PROFIBUS DP Anbindung.

1.3.3 Bussysteme PROFIBUS DP

PROFIBUS DP ist ein dezentraler Feldbus, der häufig in der Fertigungstechnik verwendet wird. Er ermöglicht die Anbindung von Modulen (DP-Slaves) nahe am Anlagenprozess und verbindet sie über ein serielles Bussystem miteinander und mit einer zentralen Steuerung. Die wichtigsten Eigenschaften von PROFIBUS DP sind:

Aufbau

PROFIBUS DP ist ein System, das aus einem Mono-Master-System besteht. Es hat einen aktiven DP-Master (z. B. CPU), der möglicherweise wechseln kann, und eine Anzahl von passiven DP-Slaves (z. B. Messumformer, E/A-Baugruppen).

Topologie

PROFIBUS DP verwendet eine Linienstruktur für die Übertragung, die auf einem elektrischen Übertragungsmedium basiert. Bei Verwendung von Lichtwellenleitern (LWL) werden Zweipunktverbindungen eingesetzt. Das System kann bis zu 32 Teilnehmer (TN) unterstützen. Durch den Einsatz von maximal 9 Repeatern kann die maximale Teilnehmerzahl auf 126 erhöht werden, wodurch eine Baumstruktur entsteht. Es ist wichtig, Abschlusswiderstände einzusetzen, die dem Wellenwiderstand der Leitungen entsprechen.

Buszugriffsverfahren

PROFIBUS DP verwendet Token Passing mit einem untergeordneten Master-Slave-Prinzip.

Übertragungstechnik nach RS485-Standard

PROFIBUS DP verwendet symmetrische Übertragung mit verdrehten Leitungen, um elektromagnetische Störungen zu reduzieren. Die Spannungspegel werden über die Pins 8 und 3 gesteuert. Die Bitcodierung der UART-Zeichen erfolgt mittels NRZ (Non Return to Zero). Jedes Datenbyte, unabhängig davon, ob es ein Datum oder Teil des Telegrammrahmens ist, wird mit einem vorangestellten Startbit (Belegung "0"), einem angehängten Stoppbit (Belegung "1") und einem Paritätsbit für gerade Parität als vorletztes Bit codiert (wobei die Quersumme des Bytes auf einen geraden Wert ergänzt wird). Der Betrieb erfolgt im Halb-Duplex-Modus. Die Übertragungsraten und maximalen Leitungslängen hängen von der Art des elektrischen Übertragungsmediums ab (ohne Einsatz von Repeatern).

Telegrammaufbau

PROFIBUS DP umfasst verschiedene Arten von Telegrammen. Ein Beispiel ist das Datentelegramm mit variabler Länge:

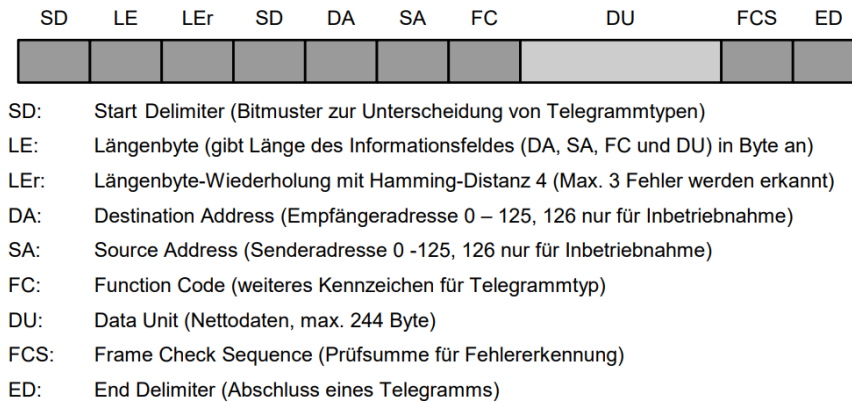


Abbildung 1.1: Telegrammaufbau

Abbildung 1.2: Der Aufbau eines PROFIBUS DP Telegramms

1.4 Vorbereitende Arbeiten

1.5 Vorbereitung

Zur Vorbereitung soll nun ein Grundverständnis für das bestehende Steuerungsprogramm geschaffen werden. Dazu wird das Programm in AWL analysiert und die einzelnen Schritte erklärt.

1.5.1 OB100

Dieser Organisationsbaustein wird nur einmalig zur Initialisierung aufgerufen.

```

1 SET
2 S M4.0
3 L 0
4 T AW260

```

Hier wird der Notaus, der in der Variable *M4.0* gespeichert ist, gesetzt. Dadurch ist der Notaus nicht gedrückt, da er drahtbruchsicher implementiert ist. Darüber hinaus wird die Bandgeschwindigkeit, die in *AW260* ist, auf 0 gesetzt.

1.5.2 OB1

Dieser Organisationsbaustein wird zyklisch aufgerufen und ist in Netzwerke unterteilt.

Netzwerk 1

```

1 % Wenn Notaus gedruickt ist, dann Motor aus
2 UN M4.0
3 SPBN M001
4 L 0
5 T AW260
6 SPB M004
7 M001: NOP 0

```

Dieses Netzwerk überprüft, ob der Notaus gedrückt ist. Wenn dies der Fall ist, wird der Motor ausgeschaltet. Wenn der Notaus nicht gedrückt ist, bleibt der Zustand des Motors unverändert.

Netzwerk 2

```
1 % Wenn Lichtschanke L1 unterbrochen ist, dann Motor umkehren
2 UN E0.2
3 R A0.1
4 SPBN M002
5 L 10000
6 T AW260
7 M002: NOP 0
```

Dieses Netzwerk überprüft die Lichtschanke *L1*. Wenn diese unterbrochen ist, wird die Drehrichtung des Motor umgekehrt, sodass das Werkstück nun wieder in Richtung der anderen Lichtschanke fährt. Wenn die Lichtschanke nicht unterbrochen ist, bleibt der Zustand des Motors unverändert. Wenn der Motor zu Anfang noch nicht gestartet war, wird die volle Drehgeschwindigkeit eingestellt und der Motor gestartet.

Netzwerk 3

```
1 % Wenn Lichtschanke L4 unterbrochen ist, dann Motor umkehren
2 UN E0.3
3 S A0.1
4 SPBN M003
5 L 10000
6 T AW260
7 M003: NOP 0
```

Dieses Netzwerk überprüft die Lichtschanke *L4*. Wenn diese unterbrochen ist, wird die Drehrichtung des Motor umgekehrt, sodass das Werkstück nun wieder in Richtung der anderen Lichtschanke fährt. Wenn die Lichtschanke nicht unterbrochen ist, bleibt der Zustand des Motors unverändert. Wenn der Motor zu Anfang noch nicht gestartet war, wird die volle Drehgeschwindigkeit eingestellt und der Motor gestartet.

Netzwerk 4

```
1 %Ende des Programms
2 M004: NOP 0
```

Dieses Netzwerk stellt das Ende des Programms dar. Jeder Sprung zu diesem Netzwerk führt zum Ende des Programms.

1.6 Aufbau

Zunächst wird der PC gestartet und die Software *TIA-Portal* gestartet. Das vorgefertigte Projekt wird geöffnet und die Variablen sowie die OBs aus der Vorbereitung hinzugefügt. Nun wird die Konfiguration getestet und das Programm auf die CPU geladen. Sollte das Programm nun vollständig funktionieren, werden nun die folgenden Schritte zur Analyse von Telegrammen auf einem PROFIBUS DP Netzwerk durchgeführt.

1.7 Aufgabe 5: Inbetriebnahme

Die Inbetriebnahme erfolgt nun mit dem übersetzen der Steuerung und dem Starten der CPU. Legt man nun ein Werkstück in eine der beiden Lichtschanken, bewegt sich das Band in die Richtung des gegenüberliegenden Sensors. Wird das Werkstück nun in die andere Lichtschanke gelegt, bewegt sich das Band wieder in die andere Richtung. Diese Änderungen können nun über die Beobachtungstabelle eingesehen werden. Steuert man nun über diese Tabelle den Merker für den Notaus auf 0, wird die Anlage, wie gewünscht, angehalten.

1.8 Aufgabe 6: ProfiTrace

Im Folgenden sollen nun die Telegramme, die zwischen der Haupt-SPS und der dezentralen Peripherie ausgetauscht werden, untersucht werden. Startet man die ProfiTrace-Software und bekommt man

von jedem Zyklus die entsprechenden Telegramme präsentiert. Jeder Zyklusdurchlauf besteht in unserem Aufbau aus dem Suchen eines zweiten Masters, die jedes Mal nicht erfolgreich ist, dann einem TokenPassing zwischen der SPS und sich selbst. Danach folgt ein Datentelegramm variabler Länge, das wie in Abbildung ?? zu sehen ist, aufgebaut ist. Die Datenübertragung erfolgt synchron und zyklisch in einem Request/Response-Schema, bei dem die SPS ein Request an die DP schickt und die DP mit einer Response antwortet. Zum Schluss wird der Zyklus mit einer Service-Übertragung synchronisiert und beendet. Dieser Zyklus wiederholt sich nun immer wieder. (siehe Abbildung ??)

Rahmennr.	Zeitstempel	Linie/Rahmen	Adr.	Dienst	Rel.typ	Req/Res/SPS	Daten	Daten	Interpret.	Station
0	0.10 ms	SD2	2->85	SD2 N10K	Data Exchange	Req	7	00 00 00 00 00 00 00		SD
1	0.23 ms	SD2	2->85	SD2 N10K	Data Exchange	Res	7	1E 44 00 2F 9C 00 00		08
2	0.37 ms	SD1	2->119	FDL Status		Req				49
3	0.62 ms	SD4	2->2	Token pass	Pass token					
4	0.67 ms	SD2	2->64	SD2 N10K	Data Exchange	Req	2	00 00		7D
5	0.76 ms	SD2	2->64	SD2 N10K	Data Exchange	Res	2	00 00		08
6	0.86 ms	SD2	2->85	SD2 N10K	Data Exchange	Req	7	00 00 00 00 00 00 00		7D
7	0.99 ms	SD2	2->85	SD2 N10K	Data Exchange	Res	7	1E 44 00 2F 9C 00 00		08
8	1.13 ms	SD1	2->120	FDL Status		Req				49
9	1.33 ms	SD4	2->2	Token pass	Pass token					
10	1.42 ms	SD2	2->64	SD2 N10K	Data Exchange	Req	2	00 00		SD
11	1.51 ms	SD2	2->64	SD2 N10K	Data Exchange	Res	2	00 00		08
12	1.62 ms	SD2	2->85	SD2 N10K	Data Exchange	Req	7	00 00 00 00 00 00 00		SD
13	1.74 ms	SD2	2->85	SD2 N10K	Data Exchange	Res	7	1E 44 00 2F 9C 00 00		08
14	1.85 ms	EN1	2->131	FDL Status		Req				49

Abbildung 1.3: ProfiTrace-Ausgabe nach der Inbetriebnahme

Aus den Zeitstempeln der Telegramme lässt sich eine Zykluszeit von ungefähr $42ms$ errechnen. Das stimmt mit der voreingestellten maximalen Zykluszeit von $150ms$ überein. Die Data Unit in den Datentelegrammen stellen Informationen und Anweisungen die von und zu der SPS fließen. Eine Änderung der Laufrichtung beispielsweise setzt andere Bits als die vorherige Richtung. Auch ein Ändern des Notauses, wie in Aufgabe 5, setzt in der Request der SPS andere Bits als im normalen Zustand.

1.9 Aufgabe 7: Busfehler

1.9.1 Aufgabe 7.1: Ausfall einer dezentralen Peripherie

Es gibt verschieden Möglichkeiten, wie eine dezentrale Peripherie ausfallen kann. Zunächst werden wir die Adresse unseres DP-Moduls softwareseitig ändern, sodass diese nicht mehr von der SPS gefunden werden kann. Dazu wird in der Hardwarekonfiguration der SPS die Adresse des DP-Moduls geändert. Nun wird die Hardwarekonfiguration neu geladen und die SPS neu gestartet. Nun wird die SPS die dezentrale Peripherie nicht mehr finden und es wird ein Busfehler auftreten. Nach dem Korrigieren der Adresse funktioniert die Anlage wieder wie im Normalbetrieb. Damit ein solches Problem auch außerhalb der roten LED an den Bauteilen angezeigt wird, wird nun der Ausgang A8.0 gesetzt, wenn ein Busfehler auftritt. Im OB86 folgt daher:

```

1 SET
2 S A8.0

```

Der OB86 wird immer genau dann aufgerufen, wenn ein Busfehler auftritt. In diesem Fall wird nun der Ausgang A8.0 gesetzt. Im Betrieb wird dieser Vorgang nun getestet, indem der SUB-DP-Stecker bei laufender CPU gezogen wird. Es kann festgestellt werden, dass der Ausgang korrekt gesetzt wird.

1.9.2 7.2: Abschlusswiderstand

Ein Bussystem basiert stark auf ausreichende Abschlusswiderstände, die eine Reflexion der Signale am Eingang einer Baugruppe verhindern. Es wird nun simuliert, was passiert, wenn der Abschlusswiderstand in unserem Versuchsaufbau fehlt. Dazu wird der Abschlusswiderstand am SUB-D-Stecker ausgeschaltet. Durch unsere geringen Leitungslängen, funktioniert das System weiterhin fehlerfrei.

Verwendet man jedoch die Ersatzleitung, die eine Leitungslänge von ca. 150m simuliert, konnte festgestellt werden, dass die Eigenschaften des Bussystems sich drastisch ändern. Bei ausgeschaltetem Widerstand ist eine Übertragung nicht möglich und der ProfiTrace-Output besteht ausschließlich aus Fehlermeldungen. Bei eingeschaltetem Abschlusswiderstand sind die Übertragungsraten jedoch stark

Rahmennr	Zeitstempel	Hinweis	Rahmen	Adr.	Dienst	Teil.typ	Req/Res	SAPS	Datenlen	Daten	Interpreter	FC	Station
0	0.25 ms		SD4	2->2	Token pass	Pass token							
1	0.30 ms		SD2	2->85	SRD HIGH	Data Exchange	Req	7	02 00 00 00 00 00 00			5D	
2	0.42 ms		SD2	2<-85	SRD HIGH	Data Exchange	Res	7	1E 44 00 47 F9 00 00			08	
3	0.55 ms		SD1	2->49	FC Status	Pass token	Req						49
4	0.61 ms		SD4	2->2	Token pass	Pass token							
5	0.66 ms		SD2	2->85	SRD HIGH	Data Exchange	Req	7	02 00 00 00 00 00 00			7D	
6	0.98 ms		SD2	2<-85	SRD HIGH	Data Exchange	Res	7	1E 44 00 47 F9 00 00			08	
7	1.13 ms		SD1	2->49	FC Status	Pass token	Req						49
8	1.37 ms		SD4	2->2	Token pass	Pass token							

Abbildung 1.4: ProfiTrace-Ausgabe mit mehreren Teilnehmern

eingeschränkt. So ist eine Übertragung mit 12MBit/s nicht möglich, da eine höhere Übertragungsrate, genauso wie eine längere Leitungslänge Reflexionen und Interferenzen an den Bussignalen begünstigt. Es folgen inkorrekt gesetzte Bytes für die korrekte Übertragung von Telegrammen und eine folglich Lahmlegung der Kommunikation zwischen Master und Slave. Bei einer Übertragungsrate von 1,5MBit/s ist eine Übertragung jedoch möglich, da die Reflexionen und Interferenzen bei dieser Übertragungsrate nicht mehr so stark sind, dass sie die Übertragung verhindern.

1.10 Aufgabe 8: Einbinden eines neuen DP-Moduls

1.10.1 Aufgabe 8.1: Einbinden ohne Hardwarekonfiguration

Zunächst wird das neue DP-Modul an den SUB-D-Stecker angeschlossen, ohne dabei eine software-seitige Änderung durchzuführen. Nach dem Starten wird das neuen Modul nicht gefunden und es nimmt kein direkten Einfluss auf den Programmablauf. Lediglich auf dem Busteilnehmer, der noch nicht in das System eingebunden war, wurde ein Busfehler auf der Zustands-LED angezeigt. Dieser Fehler wird durch das Fehlen der Hardwarekonfiguration verursacht.

1.10.2 Aufgabe 8.2: Einbinden mit Hardwarekonfiguration

Um nun das neue Modul in den Programmablauf einzubinden, wird zunächst über die ProfiTrace-Software das Bussystem nach neuen Teilnehmern durchsucht und die jeweilige Adresse herausgefunden. Nun kann eine Baugruppe aus dem Katalog im TIA-Portal ausgewählt werden, die mit der physischen Baugruppe übereinstimmt und in die Netzwerkübersicht gezogen. Über die Oberfläche kann nun die Verbindung zwischen den Baugruppen gezogen und die vorher ermittelte Adresse eingetragen werden. Nun kann die Hardwarekonfiguration neu geladen werden und die SPS neu gestartet werden. Nun wird das neue Modul gefunden und kann in den Programmablauf eingebunden werden.

1.10.3 Aufgabe 8.3: ProfiTrace-Ausgabe

Betrachtet man nun die ProfiTrace-Ausgabe, wird einem schnell klar, dass der Zyklus um ein Request/Response-Schema erweitert wurde. Es gibt zwar weiterhin ein Suchen nach einem zweiten Master und ein TokenPassing, aber die Datenübertragung erfolgt nun erst zu der ersten DP und dann zu der gerade neu eingebundenen DP. Die Data Unit der zweiten Peripherie ist jedoch deutlich kleiner, da es kaum Daten gibt, die übertragen werden müssen. (siehe Abbildung 1.4)

1.10.4 Aufgabe 8.4: Zugriff auf die Baugruppe im Programmablauf

Um nochmal einen Zugriff auf die Baugruppe im Programmablauf zu simulieren, wird nun der Adressbereich der DP in den Einstellungen im TIA-Portal auf den Adressbereich 100 gelegt. Über die Beobachtungstabelle kann nun die Adresse 100.0 angesprochen werden. In der ProfiTrace-Software ist nun zu sehen, wie Bytes in dem Payload der Request geändert wurden.

1.11 Fazit

Nach Abschluss des Labor lässt sich sagen, dass PROFIBUS DP mit dem richtigen Wissen und Equipment eine mächtige Technologie ist, die mit Recht in der Industrie eingesetzt wird. Es ist vergleichsweise einfach ein System aus dezentralen Peripherien aufzubauen, die gut miteinander kommunizieren um auch komplizierte Aufgaben zu lösen. Trotzdem gibt es viele kleine Stolperfallen und das System bietet viel Platz für Optimierung, wenn man weiß, wie diese durchzuführen sind. Die Aufgaben haben außerdem gezeigt, dass PROFIBUS DP ein sehr robustes Bussystem ist, das auch bei fehlenden Abschlusswiderständen noch funktioniert. Es ist jedoch zu beachten, dass die Übertragungsrate bei fehlenden Abschlusswiderständen stark eingeschränkt ist. Aus einer Software-Perspektive kann man sagen, dass das Arbeiten mit PROFIBUS DP eine auch einige Herausforderung darstellen kann, da es eine spezielle Hardware-Konfiguration erfordert und eine eigene Kommunikationsprotokoll-Schicht hat. Um mit PROFIBUS DP zu arbeiten, benötigt man eine spezielle Software, die in der Lage ist, mit dem Protokoll zu kommunizieren und die Daten zu verarbeiten. Es gibt jedoch auch Bibliotheken und Frameworks, die die Arbeit mit PROFIBUS DP erleichtern können. In der Regel müssen Entwickler auch über Kenntnisse in der Automatisierungstechnik verfügen, um die Anforderungen der Anlage und die Kommunikation mit der SPS zu verstehen.