



Fakultät Informatik

**Entwicklung eines VHDL-basierten Applikationsbeispiels
zur Ablösung des FM350-2 Moduls durch das TM FAST
Modul inklusive Integration in das TIA Portal und
Aspekte des Software Engineerings**

Bachelorarbeit im Studiengang Informatik

vorgelegt von

Jan-Eric Gedicke

Matrikelnummer 3643446

Erstgutachter: Prof. Dr. Meitner

Betreuer: Conrad, Maximilian

© 2025

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung und Motivation	1
1.2	Zielsetzung der Arbeit	1
1.3	Methodisches Vorgehen und Struktur der Arbeit	2
1.4	Erwarteter Nutzen und Anwendungsbereich / Passend?	2
2	Grundlagen	4
2.1	SPS-Technologie und Module im industriellen Umfeld / Überblick über SIMATIC S7 bei Thomas oder Andreas	4
2.2	Überblick über das FM350-2 Modul	7
2.3	Eigenschaften und Aufbau des TM FAST von Andreas (wichtige sachen für software architektur)	7
2.4	VHDL und FPGA-Technologie (maybe als software zusammenfas- sen) / erwähnen aber evtl wo anders besser aufgehoben	8
2.5	TIA Portals und dessen Bedeutung für die Integration (maybe als software zusammenfassen)	8
2.6	Maybe	8
3	Analyse der Ablösung des FM350-2 durch das TM FAST Modul	9
3.1	Funktionsweise und Limitierungen des FM350-2	9
3.2	Einsatzgebiete der FM 350-2	11
3.3	Vorteile und neue Möglichkeiten durch das TM FAST Modul	12
3.4	Anforderungen an die Neuentwicklung	12
4	Entwicklung des VHDL-basierten Applikationsbeispiels	14
4.1	Konzept und Architektur des neuen Applikationsdesigns	14
4.2	Umsetzung der Funktionalität in VHDL	15
4.3	Test und Validierung der VHDL-Implementierung	15
4.4	Vergleich mit der bisherigen Lösung	15

5	Integration in das TIA Portal?	16
5.1	Herausforderungen der Einbindung des TM FAST Moduls	16
5.2	Anpassungen an die Projektumgebung	16
5.3	Implementierung und Tests im TIA Portal	16
5.4	Evaluation der neuen Lösung im industriellen Kontext	16
6	Aspekte des Software Engineerings	17
6.1	Best Practices in der VHDL-Entwicklung	17
6.2	Modularisierung und Wiederverwendbarkeit	17
6.3	Teststrategien und Debugging-Techniken	17
6.4	Dokumentation und Wartung der Software	17
7	Fazit und Ausblick	18
7.1	Zusammenfassung der Ergebnisse	18
7.2	Kritische Reflexion und Herausforderungen	18
7.3	Potenzielle Weiterentwicklungen und zukünftige Anwendungen . . .	18
8	Plan	20
9	Fragen	23
10	Tests	25
11	Passwörter	26

1 Einleitung

1.1 Problemstellung und Motivation

Die Produktserie S7-300, insbesondere das FM350-2 Modul, sind nicht mehr in der aktiven Vermarktung und dadurch nicht mehr bestellbar. Das FM350-2 Modul bietet 8 Kanäle für hochkanalige Dosieranwendungen und ist nur noch als Ersatzteil verfügbar. Ein alternatives Lösungskonzept ist der Einsatz des TM FAST Moduls, eines freiprogrammierbaren Hochgeschwindigkeits-Moduls, welches eine erweiterte Kanalanzahl von 10 bis 12 Kanälen ermöglicht. Ziel dieser Bachelorarbeit ist die Entwicklung eines VHDL-basierten Applikationsbeispiels zur Demonstration der Einsatzmöglichkeiten des TM FAST Moduls in hochkanaligen Dosieranwendungen. Gleichzeitig soll die Integration in das TIA Portal erfolgen, um eine reibungslose Nutzung in industriellen Automatisierungsumgebungen zu gewährleisten.

Als dualer Student bei Siemens verfüge ich bereits über Erfahrung in der industriellen Automatisierungstechnik. Diese Arbeit bietet mir die Gelegenheit, mein Wissen in den Bereichen Embedded Systems, Industrieautomation und Software Engineering weiter zu vertiefen. Ich freue mich darauf, dieses praxisnahe Thema in Kooperation mit Siemens zu bearbeiten und einen wertvollen Beitrag zur Weiterentwicklung industrieller Automatisierungslösungen zu leisten.

1.2 Zielsetzung der Arbeit

Ziel dieser Bachelorarbeit ist die Entwicklung eines VHDL-basierten Applikationsbeispiels zur Ablösung des FM350-2 Moduls durch das TM FAST Modul. Dabei soll die Kanalanzahl von 8 auf 10 oder 12 erhöht werden, um eine flexiblere Nutzung in hochkanaligen Dosieranwendungen zu ermöglichen. Ein wesentlicher

Bestandteil der Arbeit ist zudem die Integration des entwickelten Applikationsbeispiels in das TIA Portal, um eine nahtlose Einbindung in bestehende Automatisierungslösungen zu gewährleisten. Neben der Implementierung des VHDL-Codes werden bewährte Methoden des Software Engineerings angewendet, um die Code-Qualität und Skalierbarkeit sicherzustellen. Schließlich soll eine umfassende Dokumentation erstellt werden, die eine detaillierte Beschreibung der Nutzung und Inbetriebnahme des Moduls umfasst.

1.3 Methodisches Vorgehen und Struktur der Arbeit

Zur Umsetzung der Ziele werden zunächst die bestehenden Funktionalitäten des FM350-2 Moduls analysiert und an den aktuellen Erfordernissen der Kunden gespiegelt. Dabei werden sowohl die Hardware- als auch die Software-Architektur untersucht und relevante Funktionen für die Dosieranwendung identifiziert. Anschließend erfolgt die Entwicklung des VHDL-basierten Applikationsbeispiels, wobei die ausgewählten Funktionen des FM350-2 Moduls in VHDL umgesetzt und die Kanalanzahl auf 10 oder 12 erweitert wird. Danach wird die Integration in das TIA Portal vorgenommen, indem das TM FAST Modul an das Automatisierungssystem angebunden und geeignete Schnittstellen sowie Datenstrukturen entwickelt werden. Qualitätssicherungsmaßnahmen spielen ebenfalls eine zentrale Rolle. Hierbei werden bewährte Entwicklungswerkzeuge wie Intel Quartus Prime und das TIA Portal genutzt, Teststrategien zur Funktionssicherung implementiert und eine modulare Softwarearchitektur entworfen. Abschließend wird die Dokumentation erstellt, die eine umfassende Beschreibung der Implementierung und Nutzung des Moduls sowie Anwendungsbeispiele und Testberichte enthält. Die Ergebnisse werden in einer Abschlusspräsentation vorgestellt.

1.4 Erwarteter Nutzen und Anwendungsbereich / Passend?

Das entwickelte Applikationsbeispiel bietet eine zukunftssichere Alternative zur Ablösung des FM350-2 Moduls und ermöglicht eine flexible Anpassung an in-

dustrielle Anforderungen. Ein konkretes Anwendungsbeispiel ist die Nachrüstung bestehender Anlagen, um eine Umstellung auf neue Hardware und Software zu erleichtern. Dies könnte insbesondere für Hersteller von Lebensmitteln oder pharmazeutischen Produkten von Interesse sein, die eine präzise Dosierung von Zutaten wie Aromen, Farbstoffen oder Vitaminen erfordern.

2 Grundlagen

2.1 SPS-Technologie und Module im industriellen Umfeld / Überblick über SIMATIC S7 bei Thomas oder Andreas

Ich:

Speicherprogrammierbare Steuerungen (SPS) sind durch ihre freie Programmierbarkeit, vielseitig einsetzbar und dadurch ein essenzieller Bestandteil der industriellen Automatisierungstechnik. Sie werden zum steuern und regeln von Ventilen, Antrieben, Maschinen, Pumpen und Sensoren verwendet.

ChatGPT: Speicherprogrammierbare Steuerungen (SPS) sind ein essenzieller Bestandteil der industriellen Automatisierungstechnik. In diesem Abschnitt werden die grundlegenden Prinzipien einer SPS sowie deren Aufbau und Funktionsweise erläutert. Zudem werden verschiedene Modularten vorgestellt, darunter digitale und analoge Ein-/Ausgabemodule, Zählermodule sowie Kommunikationsmodule. Ein weiteres Thema ist die Anbindung von SPS-Systemen an industrielle Netzwerke mittels gängiger Protokolle wie PROFINET, PROFIBUS und Modbus.

Thomas: Speicherprogrammierbare Steuerungen – im Englischen „Programmable Logic Controllers“ (PLCs) – wie die SIMATIC S7-Serie von Siemens steuern und regeln unter anderem Maschinen, Antriebe, Ventile, Pumpen und Sensoren. Diese Steuerungen sind frei programmierbar und dadurch universell einsetzbar [10, S. 146]. Im Folgenden wird ein Überblick über den Aufbau, die Funktionsweise und die Programmierung eines SIMATIC S7-Systems gegeben.

2.1.1 aufbau eines simatic s7

Ein SIMATIC S7-System ist modular aufgebaut und besteht aus mehreren Komponenten. Im Folgenden werden die Hauptkomponenten und ihre Funktionen erläutert. **Controller (CPU)** Die CPU der Steuerung wird auch als Controller bezeichnet und bildet die zentrale Verarbeitungseinheit des Systems. Die Hauptaufgabe der CPU ist es, das Anwenderprogramm auszuführen, das vom Nutzer entwickelt wurde, um spezifische Steuerungs- und Regelungsaufgaben zu übernehmen [11, S. 25]. Die CPU verarbeitet Signale von Ein- und Ausgabemodulen, führt Berechnungen durch und gibt Steuerbefehle an angeschlossene Geräte, wie Sensoren und Aktoren, aus. Controller der SIMATIC S7 Serie verfügen über integrierte Kommunikationsschnittstellen, wie Ethernet/PROFINET oder PROFIBUS, die den Datenaustausch mit anderen Steuerungen, Bediengeräten oder übergeordneten Systemen (z. B. SCADA oder MES) ermöglichen [12, S. 14]. **Weitere Baugruppen** Um eine Verbindung zu einer Maschine oder Anlage herzustellen, werden Signalbaugruppen verwendet, welche es als Ein- bzw. Ausgabebaugruppen sowohl für Digitale als auch analoge Signale gibt wie Spannungen oder Ströme gibt. Technologiebaugruppen sind Signalverarbeitende Baugruppen, welche eingehende Signale aufbereiten und entweder direkt an den Prozess oder zur weiteren Verarbeitung an die CPU gegeben. Ein möglicher Einsatzzweck ist z.B. das zählen von Impulsen, wofür die CPU meist zu langsam ist. Zur Erweiterung der Kommunikationsfähigkeit der CPU bezüglich Protokollen oder Funktionen kommen Kommunikationsbaugruppen zum Einsatz [11, S. 25].

2.1.2 Funktionsweise einer Speicherprogrammierbaren Steuerung

Grundsätzlich arbeitet eine Speicherprogrammierbare Steuerung nach dem Eingabe-Verarbeitung-Ausgabe-Prinzip (EVA). Der damit verbunden Prozess erfolgt zyklisch und lässt sich wie folgt darstellen: Über Eingangsbaugruppen werden zunächst die Zustände von angeschlossener Sensorik erfasst. Diese Eingangssignale werden anschließend in einem Prozessabbild gespeichert, welches eine Momentaufnahme der aktuellen Signalzustände darstellt. Änderungen der Signale, die während eines laufenden Zyklus auftreten, werden somit ignoriert [10, S. 147], [13, S. 39]. Das Prozessabbild dient nun als Datengrundlage für die Verarbeitung durch das Anwenderprogramm. Das Anwenderprogramm ist modular aufgebaut

und arbeitet mit sogenannten Bausteinen. Das Hauptprogramm wird durch den Organisationsbaustein1 (OB1) repräsentiert, welcher aus konkreten Anweisungen aber auch Funktions- oder Unterprogrammaufrufen besteht. Die Abarbeitung des OB1 entspricht einem Verarbeitungszyklus, dessen Dauer üblicherweise im zweistelligen Millisekunden Bereich liegt [13, S. 39]. Nach der Verarbeitung des Anwenderprogramms erstellt die SPS ein neues Prozessabbild, das die Ergebnisse der Programmverarbeitung enthält. Dieses Abbild wird schließlich an die Ausgangsbaugruppen übertragen, woraufhin die angeschlossenen Aktoren angesteuert werden. Mit diesem Schritt ist der Zyklus abgeschlossen, und die SPS beginnt unmittelbar mit der Erfassung der neuen Eingangs-Signalzustände für den nächsten Zyklus [10, S. 147], [13, S. 39].

2.1.3 Programmierung einer PLC mit TIA-Portal

Das Totally Integrated Automation Portal (TIA Portal) ist ein Framework für die Projektierung, Programmierung und Inbetriebnahme von SIMATIC S7-Steuerungen und deckt unter anderem Hardware-Konfiguration, Softwareentwicklung und Fehlerdiagnose ab [14, S. 11]. Die Programmierung folgt dabei der internationalen Norm IEC 61131-3, die den Standard für die Softwareentwicklung in speicherprogrammierbaren Steuerungen (SPS) definiert. Diese Norm legt die Struktur und die Programmiersprachen fest, die in Automatisierungsprojekten eingesetzt werden können, um eine einheitliche und herstellerübergreifende Programmierbarkeit zu gewährleisten. Nach IEC 61131-3 werden fünf standardisierte Programmiersprachen für SPS definiert: KOP (Kontaktplan): Eine grafische Sprache, die sich an elektrischen Schaltplänen orientiert. FUP (Funktionsplan): Eine grafische Sprache zur Darstellung von Logik und Funktionen. AWL (Anweisungsliste): Eine textbasierte Sprache, die ähnlich wie Assembler arbeitet. SCL (Structured Control Language): Eine textbasierte, hochsprachliche Sprache ähnlich zu Pascal. AS (Ablaufsprache): Eine grafische Sprache zur Darstellung von Ablaufsteuerungen [10, S. 153]. Für die Programmierung im Rahmen der Bachelorarbeit sind insbesondere die Sprachen FUP und SCL relevant. **Funktionsplan (FUP)** ist eine grafische Programmiersprache. Sie basiert auf der Darstellung von Funktionsbausteinen, die durch Verbindungen miteinander verknüpft werden. Ein typisches Beispiel in FUP ist die Implementierung von UND-, ODER- oder NICHT-Verknüpfungen, bei denen Ein- und Ausgangssignale miteinander verbunden werden. Die Funktionsbausteine enthalten logische Funktionen, mathematische Operationen oder

auch spezifische Steuerungsaufgaben, die als grafische Symbole dargestellt sind [15, S. 91 f.]. **Structured Control Language (SCL)** SCL ist eine textbasierte Programmiersprache, die der Hochsprache Pascal ähnelt. Sie ist besonders für komplexere Algorithmen und mathematische Berechnungen geeignet. SCL wird in der IEC 61131-3 als Strukturierte Text bezeichnet und unterstützt eine strukturierte Programmierung mit Schleifen, Verzweigungen und Funktionsaufrufen [10, S. 153].

2.2 Überblick über das FM350-2 Modul

ChatGPT: Das FM 350-2 Modul ist ein spezielles Zähl- und Messmodul für SPS-Systeme. Hier werden die technischen Spezifikationen und die grundlegende Funktionsweise erläutert. Typische Anwendungen, wie die Messung von Drehzahlen oder das Zählen von Impulsen, werden beschrieben. Darüber hinaus werden die Einbindung in SPS-Systeme, die verfügbaren Steuer- und Leseoperationen sowie die Limitierungen des Moduls thematisiert.

2.3 Eigenschaften und Aufbau des TM FAST von Andreas (wichtige sachen für software architektur)

ChatGPT: Das TM FAST Modul stellt eine Weiterentwicklung gegenüber dem FM 350-2 dar und ist für hochdynamische Anwendungen konzipiert. In diesem Abschnitt werden die technologischen Unterschiede sowie die Vorteile des TM FAST Moduls in Bezug auf Echtzeitfähigkeit und Signalverarbeitung betrachtet. Außerdem wird die Kompatibilität des Moduls mit bestehenden SPS-Systemen analysiert.

2.4 VHDL und FPGA-Technologie (maybe als software zusammenfassen) / erwähnen aber evtl wo anders besser aufgehoben

ChatGPT: FPGAs (Field Programmable Gate Arrays) bieten eine leistungsstarke Alternative zu klassischen SPS-Modulen. Dieser Abschnitt behandelt die Grundlagen der VHDL-Programmierung und die Architektur von FPGAs. Ein besonderer Fokus liegt auf den Unterschieden zwischen FPGAs, Mikrocontrollern und herkömmlichen SPS-Modulen sowie deren jeweiligen Vor- und Nachteilen in der Automatisierungstechnik. Abschließend werden praxisnahe Beispiele für die Implementierung von Zähl- und Messfunktionen mit VHDL vorgestellt.

2.5 TIA Portals und dessen Bedeutung für die Integration (maybe als software zusammenfassen)

ChatGPT: Das TIA Portal ist eine zentrale Entwicklungsumgebung für SPS-Programmierung und Systemintegration. Hier werden der Aufbau und die Struktur des TIA Portals beschrieben, einschließlich der unterstützten Programmiersprachen wie KOP (Kontaktplan), FUP (Funktionsplan) und SCL (Structured Control Language). Weiterhin wird die Integration von Modulen wie FM 350-2 oder TM FAST erläutert. Schließlich werden Diagnosemöglichkeiten und Optimierungsstrategien für SPS-Programme im TIA Portal vorgestellt.

2.6 Maybe

-Automatisierungspyramide -Echtzeitsysteme? übergang zu was ich mache (ist das wichtig? wahrscheinlich nicht) -verwendete Software:Questa, Quartus, TIA ... (Bei Umsetzung mit 3 Sätzen)

3 Analyse der Ablösung des FM350-2 durch das TM FAST Modul

3.1 Funktionsweise und Limitierungen des FM350-2

ChatGPT: Die FM 350-2 ist ein Zähl- und Messmodul, das speziell für schnelle Zählprozesse und Frequenzmessungen entwickelt wurde. In diesem Abschnitt werden die wesentlichen Funktionen, deren Anwendung sowie die Einschränkungen des Moduls beschrieben.

3.1.1 Grundlegende Funktionen der FM 350-2

Die FM 350-2 bietet eine Vielzahl von Funktionen zur Steuerung und Überwachung von Zählprozessen. Hierzu gehören unter anderem das Initialisieren der Zähler, das Steuern von Digitalausgängen sowie das Laden und Auslesen von Zählwerten.

3.1.1.1 Steuerung der Baugruppe mit FC CNT2_CTR

(FC2)

Die Funktion extttFC CNT2_CTR ermöglicht die Steuerung der Digitalausgänge der FM 350-2 sowie die Verwaltung der Software-Tore. Zudem können Rückmeldesignale ausgelesen werden.

Hauptfunktionen:

- Initialisierung des Zähler-DBs
- Auslesen der Rückmeldesignale und Speicherung in der Struktur CHECKBACK_SIGNALS

- Übertragen der Steuersignale aus der Struktur CONTROL_SIGNALS zur FM 350-2

Die Funktion muss zyklisch (z. B. im OB1 oder im Weckalarm OB35 der S7-300) aufgerufen werden, um eine kontinuierliche Überwachung und Steuerung zu gewährleisten.

3.1.1.2 Laden von Zählerständen, Grenz- und Vergleichswerten (extttFC3 / extttFB3)

Mit der Funktion extttFC CNT2_WR oder dem Funktionsbaustein extttFB CNT2WRPN können Zählerstände und Vergleichswerte neu geladen werden. Diese Funktion sollte nur verwendet werden, wenn im laufenden Betrieb neue Werte in das Modul eingespielt werden müssen.

3.1.1.3 Auslesen von Zähl- und Messwerten (extttFC4 / extttFB4)

Die extttFC CNT2_RD / extttFB CNT2RDPN Funktion dient zum zyklischen Auslesen der aktuellen Zählwerte. Falls keine Leseaufträge benötigt werden, kann auf diese Funktion verzichtet werden, um die Prozesslast zu minimieren.

3.1.1.4 Diagnosedaten lesen (FC5)

Im Falle eines Diagnosealarms ermöglicht die Funktion FC DIAG_RD das Laden der Diagnosealarmdaten in den Zähler-DB, um Fehlerursachen zu analysieren und entsprechende Maßnahmen einzuleiten.

3.1.2 Einschränkungen der FM 350-2

Trotz ihrer vielseitigen Einsatzmöglichkeiten weist die FM 350-2 einige Limitierungen auf:

- Begrenzte Flexibilität in der Anpassung an komplexe Steuerungsaufgaben
- Kein FPGA-basierter Aufbau, wodurch individuelle Anpassungen nicht möglich sind

- Begrenzte Echtzeitfähigkeit im Vergleich zu modernen Zählmodulen

3.2 Einsatzgebiete der FM 350-2

3.2.1 Haupteinsatzgebiete

Die FM 350-2 wird hauptsächlich dort eingesetzt, wo Signale gezählt und schnelle Reaktionen auf vorgegebene Zählerstände erforderlich sind. Besonders relevant ist sie für Anwendungen in der Fertigungs- und Automatisierungstechnik.

3.2.2 Typische Anwendungen

Typische Einsatzbereiche sind:

- Verpackungsanlagen
- Sortieranlagen
- Dosieranlagen
- Drehzahlregelungen und Überwachung von Gasturbinen

3.2.3 Anwendungsbeispiel: Kartonabfüllung

Ein typisches Beispiel für die Nutzung der FM 350-2 ist die Abfüllung von Teilen in Kartons:

- Kanal 0 zählt die Teile und steuert das Abfüllventil.
- Kanal 1 steuert den Transport der Kartons und erfasst die Anzahl der gefüllten Kartons.
- Das System stellt sicher, dass exakt die vorgegebene Anzahl an Teilen pro Karton abgefüllt wird.

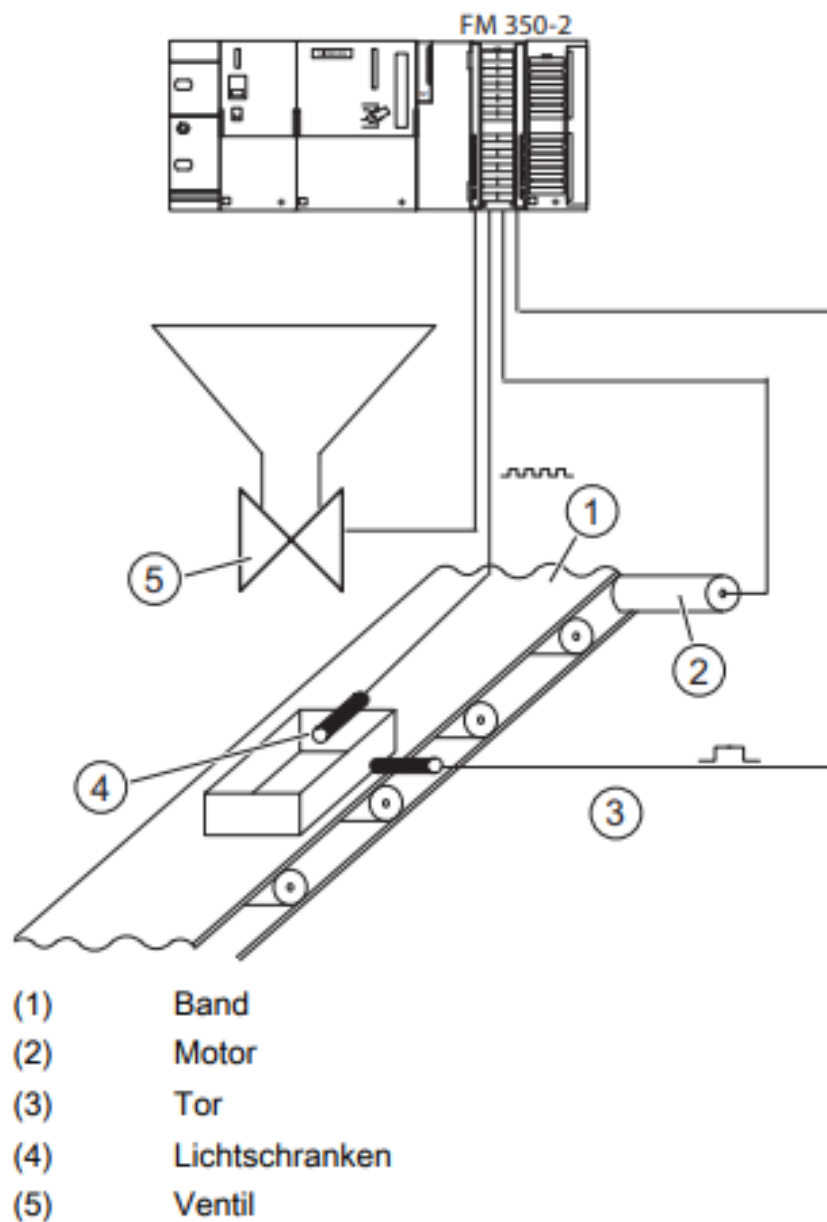


Abbildung 3.1: Beispiel für den Einsatz einer FM 350-2 in der S7-300 [7]

3.3 Vorteile und neue Möglichkeiten durch das TM FAST Modul

3.4 Anforderungen an die Neuentwicklung

Kriterium	FM350-2
Unterschiede	
Architektur	<p>Reines Zähler- und ohne programmierbare Hardware-plattform</p> <p>FPGA-Technologie</p>
Funktionalität	<p>Spezialisiert auf Zählen, Prozessungs- und Überwachungsaufgaben über-nehmen</p> <p>und Drehzahl-gesung</p>
Prozessierung	<p>Keine Integration</p> <p>direkt in den Produk-tions-prozess ein-geordnet und</p> <p>Zähler- und Mess-wer-te, die von der über-geord-neten</p>

4 Entwicklung des VHDL-basierten Applikationsbeispiels

4.1 Konzept und Architektur des neuen Applikationsdesigns

Byte Zuweisung für FB_IF (32 Byte)

- 3 Byte für Kanalobergrenze = 4095 (Standardventil ist 1 Impuls/ 1 ml -> 4l abfüllung) => bis zu 10 Kanäle möglich
- 4 Byte für Kanalobergrenze = 65535 (65l abfüllung) => nur 8 Kanäle möglich

Andere Ideen: Vielleicht auf TMFASTUserWriteREC ausweichen? (Bis zu 128 Byte) asynchron (was ist wenn daten mitten im ablauf geändert werden?)

Algorithmus, der im ersten Zyklus die Kanalobergrenzen 1–4 von der CPU an das TMFast sendet, im zweiten Zyklus 5–8 usw. ist synchron und eignet sich besser die Steuerschnittstelle, da über die im Betrieb auch taktsynchron neue Werte geschrieben werden können. Der Nachteil des Write-Records ist die Asynchronität zum Zyklus.

Zähler -> Unter- und Obergrenze einstellbar

CPU Stop beachten:

Questa DQ fehler beheben /evtl Hello world datei probieren um fehler zu finden

4.2 Umsetzung der Funktionalität in VHDL

4.3 Test und Validierung der VHDL-Implementierung

4.4 Vergleich mit der bisherigen Lösung

5 Integration in das TIA Portal?

5.1 Herausforderungen der Einbindung des TM FAST Moduls

5.2 Anpassungen an die Projektumgebung

5.3 Implementierung und Tests im TIA Portal

5.4 Evaluation der neuen Lösung im industriellen Kontext

6 Aspekte des Software Engineerings

6.1 Best Practices in der VHDL-Entwicklung

6.2 Modularisierung und Wiederverwendbarkeit

6.3 Teststrategien und Debugging-Techniken

6.4 Dokumentation und Wartung der Software

7 Fazit und Ausblick

7.1 Zusammenfassung der Ergebnisse

7.2 Kritische Reflexion und Herausforderungen

7.3 Potenzielle Weiterentwicklungen und zukünftige Anwendungen

Literaturverzeichnis

- [1] Ernst Klett Verlag (2004): *Infoblatt Siemens AG*. Verfügbar unter: <https://www.klett.de/alias/1036900>. Zugriff: 5. Januar 2025.
- [2] Merkur (2022): *Siemens – Geschichte, Aktie und Tätigkeitsfelder*. Verfügbar unter: <https://www.merkur.de/wirtschaft/siemens-geschichte-aktie-und-taetigkeitsfelder-91268844.html>. Zugriff: 19. Januar 2024.
- [3] Siemens, 2023, zitiert nach de.statista.com: *Umsatz von Siemens SE seit 2005*. Verfügbar unter: <https://de-statista-com.thn.idm.oclc.org/statistik/daten/studie/73827/umfrage/umsatz-von-siemens-seit-2005/>. Zugriff: 5. Januar 2025.
- [4] ResearchGate: Verfügbar unter: https://www.researchgate.net/figure/SA-88s-physical-model-and-an-example-production-allocation_fig2_308384237/. Zugriff: 11. Januar 2025.
- [5] Siemens (2022): *Pflichtenheft-SmartFactory*. Internes Dokument.
- [6] Liam Bee (2022): *PLC and HMI development with Siemens TIA Portal: develop PLC and HMI programs using standard methods and structured approaches with TIA Portal V17*. Packt Publishing, Limited.
- [7] Siemens AG (2011): *Zählerbaugruppe FM 350-2 Gerätehandbuch*. Verfügbar unter: https://cache.industry.siemens.com/dl/files/178/1105178/att_20226/v1/s7300_fm350_2_operating_instructions_de_de-DE.pdf. Zugriff 11. März 2025.
- [8] Siemens AG (2023): *TMFast Programmeier Handbuch*. Verfügbar unter: https://cache.industry.siemens.com/dl/files/088/109816088/att_1144842/v2/s71500_tm_fast_programming_manual_de-DE_de-DE.pdf. Zugriff 12. März 2025.

8 Plan

Hier sind einige Schritte, die Ihnen dabei helfen können, eine VHDL-Applikation für das TM FAST-Modul zu entwickeln, um das alte FM350-2-Modul zu ersetzen:

Anforderungsanalyse

- Analysieren Sie sorgfältig die Funktionalität und Schnittstellen des FM350-2-Moduls.
- Identifizieren Sie die Kernfunktionen, die das neue TM FAST-Modul erfüllen muss.
- Definieren Sie klar die Eingangs- und Ausgangssignale, die Kommunikationsprotokolle und andere relevante Spezifikationen.

Architekturentwurf

- Entwerfen Sie eine modulare Architektur für Ihre VHDL-Applikation.
- Unterteilen Sie die Funktionalität in logische Blöcke wie Steuerungslogik, Kommunikationsschnittstellen, Datenpfade usw.
- Legen Sie die Schnittstellen zwischen den einzelnen Modulen fest.

VHDL-Entwicklung

- Beginnen Sie mit der Implementierung der einzelnen VHDL-Module basierend auf Ihrem Architekturentwurf.
- Achten Sie auf eine strukturierte und modulare Codeerstellung, um die Wartbarkeit und Erweiterbarkeit zu erleichtern.
- Verwenden Sie VHDL-Sprachkonstrukte wie Prozesse, Signale, Typen und Pakete, um Ihre Applikation übersichtlich und effizient zu gestalten.

Simulation und Test

- Erstellen Sie umfangreiche Testbänke, um Ihre VHDL-Applikation gründlich zu überprüfen.
- Führen Sie Simulationen durch, um das korrekte Verhalten der einzelnen Module und der Gesamtapplikation zu verifizieren.
- Testen Sie die Applikation schrittweise, um mögliche Fehler frühzeitig zu erkennen und zu beheben.

Integration und Validierung

- Integrieren Sie Ihre VHDL-Applikation in die Gesamtarchitektur des TM FAST-Moduls.
- Testen Sie die Applikation in der realen Umgebung, um ihre Kompatibilität und Funktionalität zu validieren.
- Passen Sie die Applikation gegebenenfalls an, um Anforderungen und Spezifikationen des TM FAST-Moduls zu erfüllen.

Dokumentation und Versionskontrolle

- Erstellen Sie eine ausführliche Dokumentation, die die Architektur, Schnittstellen, Konfiguration und Verwendung Ihrer VHDL-Applikation beschreibt.
- Nutzen Sie ein geeignetes Versionskontrollsystem, um Änderungen an Ihrer Applikation nachzuverfolgen und zu verwalten.

Durch diese strukturierte Vorgehensweise können Sie sicherstellen, dass Ihre VHDL-Applikation für das TM FAST-Modul die Anforderungen des FM350-2-Moduls erfüllt und in die Gesamtarchitektur des TM FAST-Systems nahtlos integriert wird.

9 Fragen

Müssen Alarme umgesetzt werden? -> Müssen nicht berücksichtigt werden
Müssen Hardwareconfigs von der CPU oder von TMFast gehandhabt werden? -> hardwareconfig
der (z.B. Ventile) ist in VHDL (andere Kommunikationsprotokolle und co pro z.B.
Ventile) Wie verhalten sich die In- und Outputs?

- Sollen alle Eingänge Zähl- und Richtungseingang sein oder Aufteilung zwischen den beiden? Nur Zähler
-

wie funktionieren Ventile überhaupt?

bit an die CPU senden

Nachlauf bei Inkrement zum Abfüllen

CPU macht - TM Fast macht Ventil zu

auf ist 1 zu ist low für Ventil

Abfüllende Menge muss übergeben werden

VHDL auf hoher Ebene darstellen (als Baustein)

Synchron oder asynchron (für Zähler oder generell) -> Alles Synchron (TM Fast arbeitet (einstellbar mit 50mHz) schneller als ein Ventil) -> dadurch zählt es schneller als Inkremente des Ventils

clk wird bei Änderung aktiviert (High->Low / Low->High)

Steuerung des Ventils über PLC oder TMFAST? Erstmal Plc mit bool senden (Überlauf für Counter (Justierung)) / TMFast erfordert Kommunikations-Protokolle
Filter?

Anwendung in mehreren oder einem Process in VHDL?

soll der timer nur zählen, wenn die flanke positiv wird? oder auch wenn sie negativ wird? -> nur bei positiv

Allgemein: Schriftgröße und Seitenränder? Zitieren von Handbüchern (Graue Literatur) und ChatGPT? Mit Disclaimer

ES IST KEIN PRAXISBERICHT (KEIN PROZESS BESCHREIBEN) / EVTL: VORNACHTEILANALYSE

Vielleicht auf WR_REC ausweichen, wenn FB_IF nicht groß genug?

10 Tests

rückgabe über das (fb_if) feedback_interface

EncEmu encoder für impulse

schwierigkeit: di1 oder di0 zählen

RD_Rec sind azyklisch und nur selten benutzt

library test 8 counter -> name mit test cases Jürgen, ..., Kollege aus Prag

Muss ich nicht allein schreiben

Mathias fragen wenn funktionierendes Programm -> in den Nightly test muss in den nightly build

-systemlogic updaten auf v2 / DONE -zähler des sea teams verwenden / -Questa Simulation / (Questa DUT=Design under Test)

- 0000_0000_0000_0000_0000_0000_0000_0001 -> FB_IF(0)(0)<='1'; - /->Q0.0 (FB_IF(0))

- FB_IF(1)(0) == QD4

-mathias jupiter.tfs.siemens.net für nightly build (muss ins git)

-adress register verbessern

-quellcode verwaltung?

11 Passwörter

PLC: Siemens123

Masterarbeit von Andreas Kick Bachelorarbeit von Thomas Weisel Prof. Meitner
Zoom Link