BERGISCHE
UNIVERSITÄT
WUPPERTAL

Fakultät für Elektrotechnik, Informationstechnik und Medientechnik
der Bergischen Universität Wuppertal

Motion-Control-Architektur für Industrie 5.0

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

Dissertation
von
M. Sc. Timo Wilkening

Erstgutachter: Prof. em. Dr.-Ing. Joachim Holtz
Zweitgutachter: Prof. Dr.-Ing. Jens Onno Krah

Mündliche Prüfung am: 12.04.2024

Vorwort

Die vorliegende Dissertation ist das Ergebnis meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik an der Technischen Hochschule Köln sowie meiner Promotion an der Fakultät für Elektrotechnik, Informationstechnik und Medientechnik der Bergischen Universität Wuppertal. Die Möglichkeit, an diesen renommierten Instituten zu forschen und zu lernen, hätte ohne die Unterstützung vieler Personen und der Firma SEW-EURODRIVE nicht realisiert werden können.

Mit großer Dankbarkeit und Wertschätzung möchte ich all jenen meinen aufrichtigen Dank aussprechen, die mich während meines Promotionsvorhabens begleitet und unterstützt haben.

An erster Stelle gilt mein Dank Herrn Prof. em. Dr.-Ing. Joachim Holtz für die Unterstützung meines Promotionsvorhabens. Der angenehme und konstruktive fachliche Austausch mit ihm haben zum erfolgreichen Abschluss meiner Arbeit beigetragen.

Mein großer Dank gilt Herrn Prof. Dr.-Ing. Jens Onno Krah, der nicht nur als Zweitgutachter eine wichtige Rolle spielte, sondern auch als engagierter Betreuer und vertrauensvoller Mentor meine wissenschaftliche Entwicklung maßgeblich geprägt hat. Seine fachliche Expertise, seine aufmunternden Worte sowie seine inspirierenden Hinweise und generell seine Unterstützung während der gesamten Zeit meiner Arbeit habe ich sehr zu schätzen gelernt.

Ein bedeutender Dank gilt der Firma SEW-EURODRIVE. Durch ihre finanzielle Unterstützung konnte ich meine Forschungstätigkeit in vollem Umfang entfalten. Die Firma SEW-EURODRIVE hat sich nicht nur als Förderer meiner akademischen Bemühungen erwiesen, sondern auch als Partner, der die Brücke zwischen Wissenschaft und Industrie schlägt.

Ebenfalls möchte ich mich bei meinen geschätzten Kollegen von der Technischen Hochschule Köln, SEW-EURODRIVE und Intel bedanken. Ihre Mitarbeit, anregenden Diskussionen und interessanten Gespräche haben die Arbeitsatmosphäre bereichert und den Wissensaustausch gefördert. Insbesondere möchte ich mich bei Tobias Schmidt, Evgenios Mentventev, Joschka Randerath, Karl Erik Mertens, Christian Mühlfeld, Stefan Grambach, Michael Schmidt und Monica Lemke für die wertvollen Beiträge und die gute Zusammenarbeit bedanken.

Meiner Familie und Freunden, insbesondere meinen Eltern, meinen beiden Schwestern und meiner lieben Frau Caroline, gilt mein besonderer Dank. Euer Glaube an mich und eure bedingungslose Unterstützung haben mir die Kraft gegeben, meinen Weg zu gehen und meine Ziele zu erreichen.

Kurzfassung

Die direkte Zusammenarbeit zwischen Menschen und intelligenten Maschinen wie Robotern als nächste Evolutionsstufe der industriellen Fertigung wird als Industrie 5.0 bezeichnet. Ausgehend von Industrie 4.0 steht nun die Interaktion zwischen Mensch und Maschine im Mittelpunkt. Aufbauend auf der mit Industrie 4.0 erfolgten Vernetzung von Komponenten in der Fertigungs- und Maschinenautomatisierung, beschäftigt sich diese Arbeit mit dem menschenzentrierten Ansatz von Industrie 5.0 und der damit einhergehenden sicheren Mensch-Maschine-Kollaboration (MMK).

Ein Teilaspekt bei der Kollaboration ist der funktional sichere Betrieb von elektrischen Antrieben. Gerade bei Robotikapplikationen sind für die sichere Einhaltung von vorgegeben Geschwindigkeiten und Kräften komplexe Algorithmen sicherheitsbezogen zu verarbeiten. Durch menschenzentrierte Lösungen für die MMK, die durch den stark wachsenden Einsatz mobiler Assistenzsysteme weiter vorangetrieben werden, nimmt die Menge an funktional sicherer Software stetig zu, hat aber noch nicht das Niveau von Automobilanwendungen erreicht. Lange Zeit wurden Sicherheitsmethoden in den verschiedenen Branchen unabhängig voneinander entwickelt. Aus normativer Sicht weisen die Sicherheitsnormen in den verschiedenen Branchen jedoch viele Gemeinsamkeiten und ähnliche Ziele auf. In diesem Zusammenhang ergeben sich Verbesserungsmöglichkeiten für die Fertigungs- und Maschinenautomatisierung durch die Verwendung etablierter Komponenten einschließlich ihrer Dokumentation aus dem Automobilsektor. Sicherheitsrelevante Halbleiter und Softwarealgorithmen, die in beiden Bereichen eingesetzt werden können, eröffnen insbesondere für Motion-Control-Anwendungen neue Möglichkeiten.

Die vorliegende Arbeit stellt eine neu entwickelte Motion-Control-Architektur für Industrie 5.0 vor. Als Teil der Motion-Control-Architektur wird ein Industrie-PC (IPC) mit einem Multi-Core System-on-a-Chip (SoC) vorgestellt. Der IPC wird als speicherprogrammierbare Steuerung (SPS) für gemischt-kritische Industrieanwendungen konfiguriert. Darüber hinaus werden eine kompakte, energiesparende und kostengünstige Vernetzung, Steuerung und funktional sichere Überwachung von Antrieben für autonome Maschinen vorgestellt. Der sicherheitsbezogene Teil des SoCs basiert auf dem aus dem Automobilsektor bekannten Lockstep-Ansatz. Ein Konzept, bei dem komplexe Sicherheitsfunktionen sowie Diagnosetests für die angebundene Peripherie zentral im sicherheitsbezogenen Teil des IPCs ausgeführt werden, rundet die Architektur ab. Es wird Hardwarevirtualisierung verwendet, um eine Trennung von Anwendungen unterschiedlicher Kritikalität (sicherheitsbezogenen und nicht sicherheitsbezogen) auf derselben Hardware zu ermöglichen. Um die Komplexität und die Kosten des Systems zu reduzieren, werden bewährte Methoden aus den Bereichen Betriebstechnik (engl.: Operational Technology, OT) und Informationstechnik (IT) eingesetzt.

Der Nachweis der Funktions- und Leistungsfähigkeit des in dieser Arbeit konzipierten und realisierten Systems wird anhand von Anwendungen für einen Industrieroboter erbracht. Ein Delta-Parallel-Roboter demonstriert den Nutzen der erarbeiteten Konzepte.

Abstract

The direct collaboration between humans and intelligent machines such as robots as the next stage in the evolution of industrial manufacturing is referred to as Industry 5.0. Based on Industry 4.0, the focus is now on human-machine interaction. Building on the networking of components in manufacturing and machine automation that took place with Industry 4.0, this thesis focuses on the human-centric approach of Industry 5.0 and the associated safe human-machine collaboration.

One aspect of collaboration is the safety-related operation of electric drives. In robotic applications in particular, complex algorithms must be processed in a safety-related manner to ensure that specified speeds and forces are safely maintained. Due to human-centric solutions for human-machine collaboration, which are further driven by the rapidly growing use of mobile assistance systems, the amount of safety-related software is steadily increasing, but has not yet reached the level of automotive applications. For a long time, safety methods were developed independently in the various industries. However, from a normative point of view, the safety standards in the different industries have many commonalities and similar goals. In this context, there is room for improvement for manufacturing and machine automation by using established components including their documentation from the automotive sector. Safety-related semiconductors and software algorithms that can be used in both sectors open up new possibilities, especially for motion control applications.

This work presents a newly developed motion control architecture for Industry 5.0. As part of the motion control architecture, an industrial PC (IPC) with a multi-core system-on-a-chip (SoC) is presented. The IPC is configured as a programmable logic controller for mixed-critical industrial applications. In addition, a compact, energy-saving and cost-effective connection, control and safety-related monitoring of drives for autonomous machines is presented. The safety-related part of the SoC is based on the lockstep approach known from the automotive sector. The architecture is rounded off by a concept in which complex safety functions and diagnostics of the connected peripherals are executed centrally in the safety-related part of the IPC. Hardware virtualization is used to allow separation of applications of different criticality (safe and non-safe) on the same hardware. To reduce the complexity and cost of the system, proven methods from the fields of operational technology (OT) and information technology (IT) are used.

The functional and performance capabilities of the system designed and implemented in this thesis are demonstrated by applications on an industrial robot. A delta parallel robot demonstrates the advantages of the developed concepts.

Inhalt

Abbildungsverzeichnis

Abkürzungsverzeichnis

1oo2 ..... *one out of two*
1oo2D ..... *one out of two with diagnostics*
AMQP ..... *Advanced Message Queuing Protocol*
AMR ..... *Autonomer mobiler Roboter*
C2C ..... *Controller-to-Controller*
CCF ..... *Fehler gemeinsamer Ursache (engl.: Common Cause Failure)*
CFC ..... *Continuous Function Chart*
CNC ..... *Computerized Numerical Control*
Cobot ..... *Kollaborativer Roboter (engl.: Collaborative Robot)*
CPU ..... *Central Processing Unit*
CRC ..... *Zyklische Redundanzprüfung (engl.: Cyclic Redundancy Check)*
CT ..... *Stromwandler (engl.: Current Transducer)*
DC ..... *Diagnosedeckungsgrad (engl.: Diagnostic Coverage)*
DC_{avg} .*Durchschnittlicher Diagnosedeckungsgrad (engl.: Average Diagnostic Coverage)*
DDS ..... *Data Distribution Service*
DIN ..... *Deutsches Institut für Normung*
DMIPS ..... *Dhrystone Millionen Instruktionen pro Sekunde*
E/A ..... *Eingabe und Ausgabe*
ECC ..... *Error Correction Code*
EL ..... *Privilegstufe (engl.: Exception Level)*
EMV ..... *Elektromagnetische Verträglichkeit*
ERP ..... *Enterprise Resource Planning*
EU ..... *Europäische Union*
EVC ..... *Exchange Variable Connection*
EVL ..... *Exchange Variable List*
FB ..... *Funktionsbaustein*
FBD ..... *Function Block Diagram*
FPGA ..... *Field Programmable Gate Array*
FPU ..... *Gleitkommaeinheit (engl.: Floating-Point Unit)*
FSoE ..... *Failsafe over EtherCAT*
FTF ..... *Fahrerloses Transportfahrzeug*
FTS ..... *Fahrerloses Transportsystem*
FVL ..... *Programmiersprache mit nicht eingeschränktem Sprachumfang (engl.: Full Variability Language)*
GbE ..... *Gigabit-Ethernet*
GPU ..... *Graphics Processing Unit*

HG ..... *Handführung (engl.: Hand Guiding)*
HMI ..... *Human Machine Interface*
Hypercall ..... *Hypervisoraufruf (engl.: Hypervisor Call)*
I²C ..... *Inter-Integrated-Circuit*
IEC ..... *International Electrotechnical Commission*
IFA ..... *Institut für Arbeitsschutz der Deutschen gesetzlichen Unfallversicherung*
Intel VMX ..... *Intel Virtual-Machine Extensions*
Intel VT ..... *Intel Virtualization Technology*
IoT ..... *Internet of Things*
IP ..... *Internet Protocol*
IPC ..... *Industrie-PC*
ISI ..... *Intel Safety Island*
ISO ..... *International Organization for Standardization*
IT ..... *Informationstechnik*
KI ..... *Künstliche Intelligenz*
LD ..... *Ladder Diagram*
LiDAR ..... *Light Detection and Ranging*
LVL ..... *Programmiersprache mit eingeschränktem Sprachumfang (engl.: Limited Variability Language)*
M2M ..... *Machine-to-Machine*
MES ..... *Manufacturing Execution System*
MMK ..... *Mensch-Maschine-Kollaboration*
MMU ..... *Memory Management Unit*
MPU ..... *Memory Protection Unit*
MQTT ..... *Message Queueing Telemetry Transport*
MRK ..... *Mensch-Roboter-Kollaboration*
NTP ..... *Network Time Protocol*
OMG ..... *Object Management Group*
OPC UA ..... *Open Platform Communications Unified Architecture*
OpenCL ..... *Open Computing Language*
OT ..... *Betriebstechnik (engl.: Operational Technology)*
PAA ..... *Prozessabbild der Ausgänge*
PAE ..... *Prozessabbild der Eingänge*
PCI ..... *Peripheral Component Interconnect*
PFH_D ..... *Gefährlicher Ausfall je Stunde (engl.: Probability of a Dangerous Failure per Hour)*
PFL ..... *Leistungs- und Kraftbegrenzung (engl.: Power and Force Limitation)*
PI ..... *PROFIBUS & PROFINET International*
PL ..... *Performance Level*

PLL ..... *Phasenregelschleife (engl.: Phase-Locked Loop)*
PMAC ..... *Permanent Magnet Alternating Current*
POI ..... *Point of Interest*
PTP ..... *Precision Time Protocol*
PubSub ..... *Publisher-Subscriber*
PWM ..... *Pulsweitenmodulation*
ROS 2 ..... *Robot Operating System 2*
RTOS ..... *Echtzeitfähiges Betriebssystem (engl.: Realtime Operating System)*
SBC ..... *Sichere Bremsenansteuerung (engl.: Safe Brake Control)*
SCADA ..... *Supervisory Control and Data Acquisition*
SCARA ..... *Selective Compliance Assembly Robot Arm*
SCL ..... *Sicherheitskommunikationsschicht (engl.: Safety Communication Layer)*
SD ..... *Secure Digital*
Sercos ..... *Serial Realtime Communication System*
SIL ..... *Sicherheitsanforderungsstufe (engl.: Safety Integrity Level)*
SLA ..... *Sicher begrenzte Beschleunigung (engl.: Safely-Limited Acceleration)*
SLP ..... *Sicher begrenzte Position (engl.: Safely-Limited Position)*
SLS ..... *Sicher begrenzte Drehzahl (engl.: Safely-Limited Speed)*
SLT ..... *Sicher begrenztes Drehmoment (engl.: Safely-Limited Torque)*
SMARC ..... *Smart Mobility Architecture*
SoC ..... *System-on-a-Chip*
SPDU ..... *Safety Protocol Data Unit*
SPI ..... *Serial Peripheral Interface*
SPS ..... *Speicherprogrammierbare Steuerung*
SRASW .... Sicherheitsbezogene Anwendungssoftware (engl.: *Safety-Related Application Software)*
SRESW ..... *Sicherheitsbezogene Embedded-Software (engl.: Safety-Related Embedded Software)*
SRMS ..... *Sicherheitsbewerteter überwachter Stillstand (engl.: Safety-Rated Monitored Stop)*
SRP/CS ..... *Sicherheitsbezogener Teil von Steuerungen (engl.: Safety-Related Part of a Control System)*
SS1 ..... *Sicherer Stopp 1 (engl.: Safe Stop 1)*
SS2 ..... *Sicherer Stopp 2 (engl.: Safe Stop 2)*
SSD ..... *Solid-State Drive*
SSM ..... *Geschwindigkeits- und Abstandsüberwachung (engl.: Speed and Separation Monitoring)*
ST ..... *Structured Text*
STL ..... Selbsttest-Bibliothek (engl.: *Self-Test-/Software-Test-Library)*

STO ..... *Sicher abgeschaltetes Drehmoment (engl.: Safe Torque Off)*
Syscall ..... *Systemaufruf (engl.: System Call)*
TCP ..... *Werkzeugarbeitspunkt (engl.: Tool Center Point)*
TCP/IP ..... *Transmission Control Protocol/Internet Protocol*
TE ..... *Testeinrichtung*
TS ..... *Technical Specification*
TSN ..... *Time-Sensitive Networking*
UDP ..... *User Datagram Protocol*
VDA ..... *Verband der Automobilindustrie*
VDMA ..... *Verband Deutscher Maschinen- und Anlagenbauer*
VDW ..... *Verein Deutscher Werkzeugmaschinenfabriken*
VM ..... *Virtuelle Maschine*
VMM ..... *Virtual-Machine-Monitor*
VPN ..... *Virtual Private Network*
vUART ..... *virtueller Universal Asynchronous Receiver Transmitter*
WLAN ..... *Wireless Local Area Network*
ZVEI ..... *Zentralverband Elektrotechnik und Elektroindustrie e.V.*

Symbole

K_T	Drehmomentkonstante
T_a	Abtastzeit
T_cyc	Zykluszeit
T_Geber	Übertragungszeit vom Drehgeber-Winkel zur sicheren Logik
T_Logik	Verarbeitungszeit von Sicherheitsfunktionen in der sicheren Logik
T_SPDU	Verzögerung durch das sicherheitsbezogene Protokoll
T_SSPS	Zykluszeit der Sicherheitsanwendung
T_STO	Ausführungszeit der Reaktionsfunktion STO im Antriebssystem
T_t	Totzeit
a_max	Maximalbeschleunigung
i_q	Augenblickswert der drehmomentbildenden Stromkomponente
i_u, i_v, i_w	Augenblickswert der Phasenströmen U, V, W
i_α, i_β	Augenblickswert der α- und β-Komponente der Phasenströme in Raumzeigerdarstellung
$s_0, \vec{s}_0$	Bereits zurückgelegte Strecke zum Startzeitpunkt t_0
s_Reaktion	Strecke, die aufgrund der Reaktionszeit zusätzlich zurückgelegt wird
$t_\mathrm{DDS\,delay}$	Latenz bei der DDS-Kommunikation über Ethernet
$t_\mathrm{DDS\,Sub}$	Zeitstempel beim Empfang von DDS-Daten
t_k	Abtastzeitpunkt
t_Reaktion	Reaktionszeit zum Auslösen von Reaktionsfunktionen
$t_\mathrm{RX\,delay}$	Gesamtverzögerung bei der DDS-Kommunikation
t_SPS	Zeitstempel in der SPS-Task
t_Sync	Alter eines Prozessdatums in der Empfänger-SPS
$v_0, \vec{v}_0$	Geschwindigkeit zum Startzeitpunkt t_0
$v_\mathrm{k}, \vec{v}_\mathrm{k}$	Geschwindigkeit zum Zeitpunkt t_k
v_SLS	Geschwindigkeitsgrenzwert für die sicher reduzierte Geschwindigkeit
φ_e	Kommutierungswinkel
φ_k	Winkel zum Zeitpunkt t_k
$\varphi_{\mathrm{k}_\mathrm{pub}}$	Publizierter Winkel zum Zeitpunkt t_k
$\varphi_{\mathrm{k}_\mathrm{sub}}$	Berechneter Winkel in der Empfänger SPS
ω_k	Winkelgeschwindigkeit zum Zeitpunkt t_k
Δt	Differenz zweier t-Werte
$\Delta\varphi$	Differenz zweier φ-Werte

$a, \vec{a}$	Augenblickswert der translatorischen Beschleunigung von beweglichen Maschinenteilen
m	Augenblickswert des Drehmoments
$p, \vec{p}$	Augenblickswert der Position von beweglichen Maschinenteilen
$s, \vec{s}$	Augenblickswert der Strecke
t	Zeit
$v, \vec{v}$	Augenblickswert der translatorischen Geschwindigkeit von beweglichen Maschinenteilen
φ	Augenblickswert des Winkels
ω	Augenblickswert der Winkelgeschwindigkeit

1 Einleitung

1.1 Motivation

Mit Arbeitsplätzen und Wohlstand leistet die Industrie den größten Beitrag zur europäischen Wirtschaft. Damit die europäische Industrie weiterhin Wohlstand schaffen kann, muss sie sich durch ständige Innovationen an die sich wandelnden Herausforderungen anpassen. Die Industrie muss dabei ihre Effizienz an verschiedenen Stellen der Wertschöpfungskette verbessern und die Flexibilität sowie die Effizienz ihrer Produktionssysteme steigern, um den sich schnell ändernden Anforderungen der globalen Verbraucher gerecht zu werden [1].

Der zunehmende Bedarf nach agilen und dynamischen Produktionssystemen im Kontext von Paradigmen wie Industrie 4.0 ruft die Anforderung nach neuen Technologien hervor. Die Digitalisierung der Produktion definiert dabei neue Funktionen für die Komponenten der Automatisierungspyramide, die hinsichtlich Software und Hardware modifiziert werden müssen, um den neuen Anforderungen gerecht zu werden [2]. Die Industrie 4.0 strebt dabei eine erhöhte Digitalisierung der Automatisierungspyramide an [3]. Um dies zu erreichen, sind neue Technologien erforderlich, die entweder in der Automatisierungstechnik entwickelt oder aus der Informationstechnik (IT) in die Automatisierungstechnik integriert werden müssen. So finden beispielsweise heute schon Technologien aus der IT wie z. B. das Internet of Things (IoT), Cloud Computing, cyber-physische Systeme und das Publisher-Subscriber (PubSub)-Modell den Einzug in die Automatisierungstechnik und übernehmen eine entscheidende Funktion für die Vernetzung und den Informationsaustausch von Komponenten [2]. Mit der zunehmenden Integration von IT-Technologien in die Automatisierungspyramide verschwimmen die Grenzen zwischen IT und der Betriebstechnik (engl.: Operational Technology, OT). Im Zentrum dieser Entwicklungen stehen neue Steuerungssysteme, die die Grenze zwischen der IT und der OT bilden. Sie sind für die operative Prozessführung verantwortlich und verbinden die Feldebene mit dem oberen IT-Teil der Automatisierungspyramide.

Im Mittelpunkt der industriellen Revolutionen standen bisher immer Allzwecktechnologien. Wie die drei ersten industriellen Revolutionen zeichnet sich auch Industrie 4.0 durch technologische Fortschritte aus, die erhebliche positive Auswirkungen auf den Gewinn haben, während die ökologischen und sozialen Aspekte vernachlässigt werden [4]. Seit der Veröffentlichung des Whitepapers der Europäischen Union (EU) [1] im Jahr 2021,

welches das Zeitalter der Industrie 5.0 ankündigt, hat sich der Forschungstrend zur Industrie 5.0 verstärkt. Im Vergleich zu früheren industriellen Revolutionen, die den Schwerpunkt eher auf den wirtschaftlichen Aspekt der Nachhaltigkeit betonten, orientiert sich die Vision der Industrie 5.0 am Menschen und den gesellschaftlichen Bedürfnissen. Die Industrie 5.0 wird im Allgemeinen durch drei Kernelemente beschrieben: Menschenzentrierung, Nachhaltigkeit und Widerstandsfähigkeit.

Menschenzentrierte Lösungen und Technologien für die Mensch-Maschine-Kollaboration (MMK) sollen nach [1] darauf abzielen, die Stärken von Menschen und Maschinen miteinander zu verbinden und zu kombinieren. Dabei sollen Technologien zum Einsatz kommen, die sich dem Arbeitnehmer anpassen und nicht umgekehrt. Durch die individuellen Anforderungen in der Industrie kann der Mensch nicht vollständig durch Maschinen ersetzt werden [5], [6]. Ein Grund dafür ist, dass die Anwesenheit des Menschen dem System eine höhere Fehlertoleranz verleiht [7]. Der menschenzentrierte Ansatz ist dabei eine Voraussetzung für Fertigungsanlagen, die Flexibilität, Agilität und Robustheit gegenüber Störungen anstreben [8].

Bereits seit den 1980er Jahren werden in der Industrie Roboter im größeren Umfang eingesetzt. Solche Industrieroboter werden bis heute noch ohne direkten menschlichen Eingriff hinter trennenden Schutzeinrichtungen betrieben. Sie sind zwar in der Lage, Aufgaben mit hoher Geschwindigkeit und Traglast auszuführen, eine Zusammenarbeit mit dem Menschen ist jedoch nicht vorgesehen. Im Gegensatz dazu werden immer mehr spezielle Robotersysteme eingesetzt, die sich im direkten Umfeld des Menschen aufhalten. Diese kollaborativen Roboter (engl.: Collaborative Robot, Cobot), fahrerlosen Transportsysteme (FTS) bzw. fahrerlosen Transportfahrzeuge (FTF) und autonomen mobilen Roboter (AMR) sind sehr vielfältig und ihr ständiger technologischer Fortschritt ermöglicht weitere Anwendungsbereiche, die verschiedene Automatisierungsanforderungen erfüllen [9]. Ein Nachteil von Cobots ist, dass sie mit reduzierter Geschwindigkeit und Kraft arbeiten, unabhängig davon, ob ein Mensch in der Nähe ist oder nicht. Für eine effizientere Fertigung und Produktion ist es vorteilhaft, wenn Industrieroboter autonom mit hoher Geschwindigkeit und Traglast arbeiten können, während sie in der Nähe von Menschen ebenso langsam und ungefährlich wie Cobots agieren.

Ein maßgeblicher Aspekt bei der Mensch-Roboter-Kollaboration (MRK) ist der funktional sichere Betrieb elektrischer Antriebe, wie er in der internationalen Norm 61800-5-2 der International Electrotechnical Commission (IEC) beschrieben ist [10]. Insbesondere bei Robotikanwendungen mit nichtlinearen kinematischen Transformationen müssen

komplexe Algorithmen zur Einhaltung vorgegebener Geschwindigkeiten und Kräfte sicherheitsbezogen verarbeitet werden. Innovationen wie Mikrocontroller, Kommunikationsnetzwerke und PCs haben schnell Einzug in die Automatisierung gehalten und einen enormen Einfluss ausgeübt. Dies führte zu einer höheren Flexibilität und Verfügbarkeit bei gleichzeitig reduzierten Kosten. In den Sicherheitsnormen waren diese Innovationen zu Beginn untersagt und Komponenten der Sicherheitstechnik wurden zunächst fest verdrahtet. Heutzutage haben sich Mikrocontroller und Software in unzähligen Anwendungen bewährt und somit Voraussetzungen geschaffen, um in internationalen Sicherheitsnormen berücksichtigt zu werden.

Lange Zeit wurden die Sicherheitsmethoden in den verschiedenen Branchen unabhängig voneinander entwickelt. Aus normativer Sicht ist die IEC 61508 branchenübergreifend zu betrachten [11] . Die jeweiligen Normen der International Organization for Standardization (ISO), die ISO 26262 für Automobilanwendungen und die ISO 13849 für die Maschinenautomatisierung, bauen auf der IEC 61508 auf und haben somit viele Gemeinsamkeiten und ähnliche Ziele [12], [13]. Dies hat zur Folge, dass Komponenten für den Automobilsektor zusammen mit der zugehörigen Dokumentation den Aufwand für die Entwicklung und die Zertifizierung der funktionalen Sicherheit auch für andere Anwendungsbereiche reduzieren können [14].

Im Bereich der Automatisierung nimmt der Anteil sicherheitsrelevanter Software kontinuierlich zu, ist aber noch weit von dem Anteil entfernt, der für Automobilanwendungen typisch ist. Mit dem schnell wachsenden Einsatz von Robotern, Cobots, FTFs und AMRs in der Industrie ändert sich dies derzeit. Automobil- und Industrieanwendungen, einschließlich ihrer Sicherheitsalgorithmen, werden sich immer ähnlicher. Die Tatsache, dass sicherheitsbezogene Halbleiter und Algorithmen in beiden Bereichen eingesetzt werden können, eröffnet insbesondere für Motion-Control-Anwendungen völlig neue Perspektiven [15], [16].

1.2 Zielsetzung

Im Rahmen dieser Arbeit wird eine neu entwickelte Motion-Control-Architektur für Industrie 5.0 vorgestellt. Ein Industrie-PC (IPC), basierend auf einem leistungsfähigen Multi-Core System-on-a-Chip (SoC), konfiguriert als speicherprogrammierbare Steuerung (SPS) für gemischt-kritische Industrieanwendungen, bildet den Kern der Motion-Control-Architektur. Es wird zudem eine kompakte, energiesparende und kostengünstige Vernetzung, Steuerung und funktional sichere Überwachung von Antrieben für Cobots, Roboter und ähnliche autonome Maschinen vorgestellt. Darüber hinaus wird ein Konzept entwickelt, bei dem komplexe Sicherheitsfunktionen sowie die zugehörige Diagnose zentral im sicherheitsbezogenen Teil des IPCs ausgeführt werden.

Der IPC als Steuerung für Maschinenmodule soll zu den beiden Domänen IT und OT kompatibel sein und die Feldebene der Automatisierung mit der IT-Welt verbinden. Da Automatisierungssysteme immer leistungsfähiger und komplexer werden, soll die Komplexität der Systeme durch Modularisierung und den Einsatz bewährter Technologien reduziert werden. Es werden im Folgenden Anforderungen und Aufgaben definiert, die vom IPC umgesetzt werden sollen:

- Paralleler Betrieb von Applikationen mit unterschiedlicher Kritikalität: sicherheitsbezogene und funktionale Anwendungen werden auf der gleichen Hardware ausgeführt. Diese Steuerung für gemischt-kritische Anwendungen wird als Compound-SPS bezeichnet und besteht aus einer Standard-SPS und einer Sicherheits-SPS.
- Berechnung komplexer Sicherheitsfunktionen mit nichtlinearer Mathematik und Gleitkomma-Arithmetik in kurzen Zykluszeiten, um die Kollaboration von Menschen und Maschinen zu ermöglichen.
- Unterstützung gängiger OT-Feldbusprotokolle für die Steuerung von Mehrachs-Systemen wie Roboter, Cobots, FTFs und AMRs.
- Einsatz bewährter Technologien auf Basis eines Linux-Betriebssystems für den nicht sicherheitsbezogenen Teil der Steuerung.
- Einsatz von IPC-basierten Hardware- und Softwarekomponenten für eine skalierbare und flexible Automatisierungslösung.
- Virtualisierung soll eine effiziente Ressourcennutzung und Isolation der Anwendungen ermöglichen.
- Effiziente Kommunikation mit PubSub-Methoden aus dem IT-Sektor.

- Sicherheitsbezogene Kommunikation auf IT- und OT-Ebene.

Um die genannten Anforderungen zu erfüllen, wird ein Konzept vorgestellt, das durch seine Architektur die beschriebenen Anforderungen grundsätzlich unterstützt. Die vorgestellte Systemarchitektur ermöglicht eine Trennung und Partitionierung der Komponenten auf derselben Hardware durch Virtualisierung. Es werden Anwendungen logisch voneinander getrennt, indem ihnen die zugrunde liegenden Ressourcen wie Rechenleistung (Prozessorkerne), Arbeitsspeicher, nichtflüchtiger Speicher, Eingabe und Ausgabe (E/A)-Peripherie und Netzwerkschnittstellen zugewiesen werden. Abbildung 1 veranschaulicht die vorteilhafte Kombination von Technologien aus verschiedenen Sektoren für die Automatisierung von Maschinenmodulen.

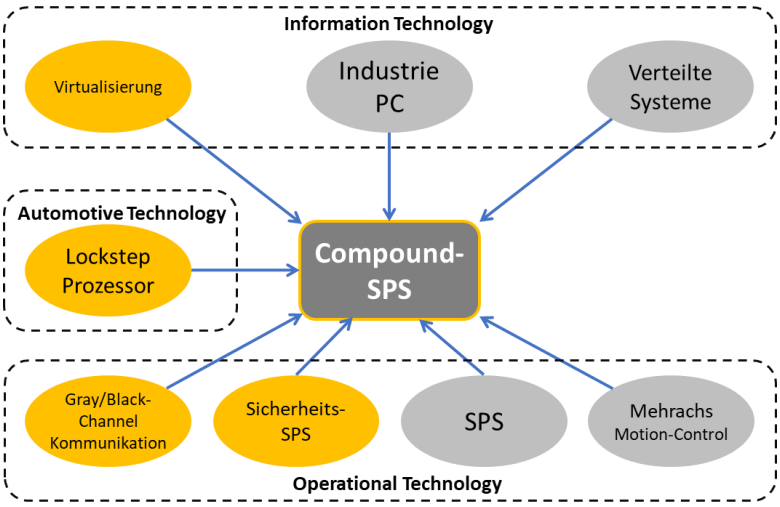


Abbildung 1: Einsatz bewährter Technologien für eine Compound-SPS.

1.3 Gliederung der Arbeit

Nach der Motivation und der Zielsetzung folgt ein Überblick über den Aufbau der vorliegenden Arbeit. Dazu wird der Inhalt der folgenden Kapitel kurz zusammengefasst.

In Kapitel 2 wird der Stand der Technik und die notwendigen Grundlagen für diese Arbeit vorgestellt. Dabei werden zunächst Steuerungssysteme in der Fertigungs- und Maschinenautomatisierung, ihre verwendeten Prozessoren und die eingesetzten Kommunikationsmethoden beschrieben. Anschließend wird der Stand der Technik von Steuerungen für die funktionale Sicherheit in der Fertigungs- und Maschinenautomatisierung aufgezeigt und die für diese Arbeit relevanten Sicherheitsnormen vorgestellt. Des Weiteren wird die Hardwarevirtualisierung erläutert und die verschiedenen Virtualisierungstechnologien beschrieben und verglichen.

In Kapitel 3 wird ein neues Sicherheitskonzept mit zentraler Sicherheitssteuerung vorgestellt. Dieser Ansatz basiert auf einer hohen Rechenleistung heutiger Sicherheitssteuerungen und schnellen ethernetbasierten Feldbussen mit sicherheitsbezogener Datenübertragung. Es wird beschrieben, wie dieser Ansatz die Kollaboration von Menschen und Robotern vereinfacht und die Komplexität sowie die hohen Kosten von sicherheitsbezogenen Komponenten reduzieren kann, indem Sicherheitsfunktionen und Diagnosetests zentral ausgeführt werden.

In Kapitel 4 wird eine effiziente Vernetzung von Automatisierungskomponenten mit bewährten Kommunikationsmethoden aus den Bereichen OT und IT gezeigt. Eine Modulsteuerung (eine SPS für ein Maschinenmodul), die die Grenze zwischen IT und OT bildet, vernetzt Komponenten der Feldebene mit Komponenten und Systemen aus der IT über verschiedene Kommunikationsmethoden.

In Kapitel 5 werden verschiedene Ansätze einer Systemarchitektur für eine gemischt-kritische Compound-SPS für die Steuerung, Vernetzung und funktional sichere Überwachung von autonomen Maschinen vorgestellt. Es wird eine skalierbare und flexible Architektur entwickelt, die den hohen Anforderungen von Industrie 5.0 und dem damit verbundenen menschenzentrierten Ansatz gerecht wird.

In Kapitel 6 wird die Implementierung der vorgestellten Systemarchitektur auf Basis verwendeter Hardware- und Softwarekomponenten beschrieben.

In Kapitel 7 wird die Validierung der Konzepte für die Sicherheitslösung mit zentraler Diagnose, MRK und effizienter Vernetzung mit OT- und IT-Kommunikationsmethoden gezeigt. Der Nachweis der Funktions- und Leistungsfähigkeit des in dieser Arbeit entworfenen und realisierten Systems wird anhand von Anwendungen für einen Industrieroboter erbracht.

In Kapitel 8 werden die Ergebnisse und Erkenntnisse der Arbeit abschließend zusammengefasst und mit einem Ausblick bewertet.

2 Grundlagen und Stand der Technik

2.1 Steuerungssysteme

2.1.1 Speicherprogrammierbare Steuerungen

Richard E. Morley stellte 1969 mit der Modicon 084 die erste SPS als halbleiterbasierendes sequentielles Logiksystem vor. Die Programmierung erfolgte grafisch mit einer stromlaufplanähnlichen Darstellung, dem inzwischen in der IEC 61131-3 [17] standardisierten Ladder Diagram (LD). Durch die höhere Flexibilität hat die SPS die zuvor übliche fest verdrahtete Anordnung von Relais abgelöst. Der Übergang von der verbindungsprogrammierten Steuerung zu einer frei programmierbaren Steuerung wird inzwischen als dritte industrielle Revolution (Industrie 3.0) bezeichnet.

Um die bis dahin übliche aufwendige Parallelverdrahtung binärer Signale und die hinsichtlich der elektromagnetischen Verträglichkeit (EMV) anfällige analoge Signalübertragung durch eine flexibel konfigurierbare digitale Übertragungstechnik zu ersetzen, wurde in den 80er Jahren die erste Generation der Feldbustechnik entwickelt. Eine weltweite Standardisierung der verschiedenen Feldbusse erfolgte 1999 mit der Norm IEC 61158. Während die erste Feldbus-Generation in der Regel auf RS-485-Transceivern basierte, verwendet die zweite Feldbus-Generation Ethernet. Die Automatisierung mit SPSen und Feldbussen zur Vernetzung wird heute als OT bezeichnet.

In der zweiten Hälfte der 1980er Jahre nutzte u. a. Beckhoff Automation erstmals PC-Technik, um damit IPCs mit SPS-Funktionalität anzubieten. PCs mit Intel Prozessoren boten aufgrund der viel höheren Stückzahlen in der IT vergleichsweise viel Rechenleistung mit Gleitkomma-Arithmetik zu günstigen Kosten. Die IPCs konnten SPS, Motion-Controller, Feldbusanschaltung und Human Machine Interface (HMI) in einer Automatisierungskomponente zusammenfassen: Aus ehemals getrennten Hardwarekomponenten wurden Softwaremodule, die auf einer skalierbaren Hardware ausgeführt werden. Während bei den ersten IPCs Bildschirm mit Tastatur und Maus oft als kostengünstiges User Interface genutzt wurden, ist dies inzwischen nicht mehr üblich, da eine remote Anbindung zusätzliche Flexibilität ermöglicht. IPCs können heute flexibel erweitert werden. Nahezu alles, was über standardisierte Schnittstellen an PCs angeschlossen werden kann, lässt sich somit auch in Automatisierungssysteme einbinden.

Die ersten SPSen hatten kein eigenes Betriebssystem, die SPS-Software interagierte direkt mit der zugrunde liegenden Hardware. Bei den später eingeführten IPCs wurde zunächst

das jeweils aktuelle PC-Betriebssystem von Microsoft mit einer proprietären Echtzeiterweiterung für OT erweitert. Die Software-SPS-Erweiterung für den IPC, die sogenannte Soft-SPS, wurde mit höchster Priorität vom Prozessor ausgeführt. Das Betriebssystem konnte die noch verbleibende Prozessorzeit für andere Aufgaben nutzen. Inzwischen werden die ehemals dominierenden PC-Betriebssysteme von Microsoft sukzessive durch echtzeitfähige Betriebssysteme (engl.: Realtime Operating System, RTOS) mit Unix bzw. Linux ersetzt.

2.1.2 Robotersteuerungen

Heutzutage wird Flexibilität von Produktionsanlagen gefordert, um schnell auf Kundenwünsche reagieren und Produktion sowie Prozesse anpassen zu können, ohne die Effizienz zu beeinträchtigen. In diesem Zusammenhang sind Industrieroboter aufgrund ihrer Vielseitigkeit, mit der sie flexible und konfigurierbare Fertigungsaufgaben ausführen können, ein wichtiges Werkzeug für moderne Fabriken. Die Steuerung und Programmierung von Industrierobotern ist jedoch ein limitierender Faktor für deren effektiven Einsatz, insbesondere in dynamischen Produktionsumgebungen oder bei komplexen Anwendungen [18]. Die meisten heute eingesetzten Industrieroboter sind vollständig von den Softwareplattformen ihrer Hersteller abhängig. Speziell die Programmierung der Roboter basiert auf spezifischen, proprietären Robotersprachen, die von jedem Roboterhersteller angeboten werden [19]. Die Syntax dieser herstellerspezifischen Programmiersprachen unterscheidet sich zum Teil erheblich, auch wenn sie in ihren Grundzügen an Hochsprachen wie PASCAL oder C angelehnt sind. Die Versuche, die Roboterprogrammiersprachen zu standardisieren, z. B. mit der Deutschen Institut für Normung (DIN)-Norm 66312 [20], sind als gescheitert zu betrachten, weil die Roboterhersteller die Normen in ihren Robotersteuerungen nicht umsetzen [21]. Die fehlende Interoperabilität erschwert die Entwicklung einer interaktiven Steuerung mehrerer Industrieroboter verschiedener Hersteller. Dies macht sich nicht nur durch die Programmierung in unterschiedlichen Programmiersprachen bemerkbar, sondern auch durch die verwendete Projektierungssoftware, die speziell auf das System des jeweiligen Herstellers zugeschnitten ist. Der Programmcode lässt sich dadurch nicht herstellerübergreifend nutzen.

Neben den Robotersteuerungen können heutzutage auch IPCs mit Soft-SPS-Erweiterung Robotik-Aufgaben übernehmen. Die in der IEC 61131-3 standardisierten Programmiersprachen werden von allen SPS-Herstellern übergreifend genutzt und finden eine hohe Akzeptanz in der Branche. Das PLCopen Komitee für Motion-Control übernimmt die Funktionalitäten aus der Robotik und überträgt sie in die SPS-Welt, indem sie verschiedene IEC 61131-3 Funktionsbausteine (FB) für eine koordinierte Bewegungssteuerung

von Mehrachs-Systemen wie Robotern definiert [22]. Damit kann die SPS- und Motion-Control-Funktionalität in Form eines Baukastensystems in eine IEC 61131-3 Entwicklungsumgebung für SPSen integriert werden.

2.1.3 Prozessoren

In der Anfangszeit waren beispielhafte Kennwerte für Industriesteuerungen die nutzbare Speicherkapazität und die Ausführungsgeschwindigkeit. Mit zunehmender Nutzung von IPCs, war es üblich stattdessen den in der SPS eingesetzten Prozessor als Kennwert zu nennen. Typischerweise werden in der Industrie nur solche Prozessoren genutzt, die auch über einen längeren Zeitraum verfügbar sind. Während es in der IT üblich ist, dass jedes Jahr Geräte mit leistungsfähigeren Prozessoren vorgestellt werden, betragen die Innovationszyklen in der OT typischerweise drei bis fünf Jahre. Prozessoren für Industriesteuerungen können grob in drei Gruppen klassifiziert werden:

1. Weniger leistungsstarke Prozessoren, z. B. ARM Cortex-M, Cortex-R oder Infineon TriCore, werden in Antriebssteuerungen und kleineren (Sicherheits-)SPSen eingesetzt. Außerdem basieren kostengünstige IPCs oft auf ARM Cortex-A Multi-Core Prozessoren, die auch im Raspberry Pi eingesetzt werden. Ein Beispiel hierfür sind die ctrlX CORE Steuerungen von Bosch Rexroth [23].
2. Intel Atom Prozessoren bieten einen kostengünstigen Kompromiss zwischen Prozessorleistung und Leistungsaufnahme. IPCs auf Basis dieser Prozessoren benötigen zwar einen geeigneten Kühlkörper, ein Lüfter ist aber meist nicht erforderlich. Ein Beispiel hierfür ist der MOVI-C CONTROLLER progressive von SEW-EURODRIVE mit dem Intel Atom E3845 Prozessor [24].
3. Intel Core i Prozessoren sind leistungsstark, da sie mit hohen Taktraten betrieben werden. Diese Prozessoren benötigen in der Regel ein Kühlsystem mit Lüfter und sind für mobile, akkubetriebene Anwendungen nur bedingt einsetzbar.

2.1.4 Kommunikationsmethoden

Die Automatisierungspyramide bietet ein Modell zur Kategorisierung der Aufgaben und Funktionalitäten von Techniken und Systemen und stellt diese in verschiedenen Ebenen dar. Die Abbildung 2a zeigt die Automatisierungspyramide mit den verschiedenen Ebenen: Unternehmensebene (Enterprise Resource Planning, ERP), Betriebsleitebene (Manufacturing Execution System, MES), Prozessleitebene (Supervisory Control and Data Acquisition, SCADA), Steuerungsebene und die Feldebene. Wie in Abbildung 2a dargestellt,

ist es für den oberen Teil der Automatisierungspyramide üblich, Kommunikationstechnologien aus dem Bereich der IT zu verwenden. Eine gängige Lösung ist beispielsweise das Transmission Control Protocol/Internet Protocol (TCP/IP) auf der Basis von Gigabit-Ethernet (GbE), typischerweise in einer Server-Client-Architektur mit Open Platform Communications Unified Architecture (OPC UA). Die als OT bezeichnete Vernetzung von Steuerungen mit Komponenten der Feldebene mit Feldbussystemen stellt die untere Hälfte der Automatisierungspyramide dar. Abbildung 2b zeigt die Steuerungsebene der Automatisierungspyramide im Detail, die in drei Unterebenen aufgeteilt ist: Maschinenautomatisierung, Modulautomatisierung und Motorsteuerung. Im OT-Bereich sind PROFINET und EtherNet/IP weit verbreitet, für die Motorsteuerungsebene sind jedoch Feldbusse mit synchroner Abtastung der Peripherie besser geeignet. Die meisten Geräte der Feldebene nutzen bisher statt GbE Fast Ethernet mit 100 Mbit/s und den damit einhergehenden geringeren Kosten. In der Abbildung 2b stellen nummerierte Pfeile die Kommunikationsverbindungen zwischen den Automatisierungsgeräten dar. Abbildung 2c zeigt verschiedene Kommunikationsmethoden, die in der Automatisierungstechnik in den verschiedenen Ebenen verwendet werden. Die Zykluszeit der verschiedenen Anwendungen innerhalb der Steuerungsebene nimmt dabei nach oben hin zu.

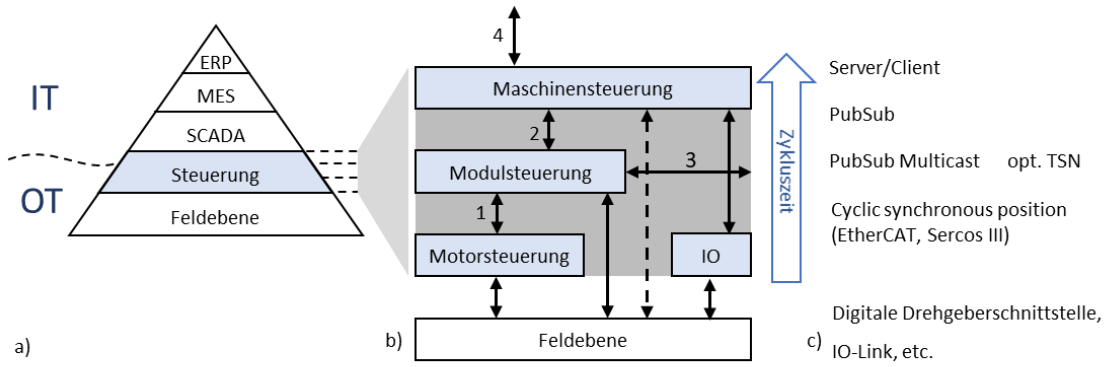


Abbildung 2: Kommunikationsmethoden in der Automatisierung.

Die synchrone zyklische Abtastung für die Regelung von Mehrachsbewegungen ist ein bewährtes Verfahren, das ursprünglich für Computerized Numerical Control (CNC)-Maschinen entwickelt wurde. Speziell bei Werkzeugmaschinen muss die Abtastung mikrosekundengenau erfolgen. Der Verein Deutscher Werkzeugmaschinenfabriken (VDW) hat 1987 zusammen mit dem Zentralverband Elektrotechnik- und Elektronikindustrie e.V. (ZVEI) eine offene Schnittstellenspezifikation entwickelt, die für digitale Antriebssysteme geeignet ist. Die daraus resultierende Spezifikation mit dem Namen Serial Realtime Communication System (Sercos) wurde 1995 als IEC 61491 mit dem Schwerpunkt auf

Motion-Control-Systemen veröffentlicht. Die mit dem ursprünglich glasfaserbasierten Sercos Interface eingeführte Synchronisation der Antriebe bis hin zur Pulsweitenmodulation (PWM) und die daraus resultierenden Betriebsarten (z. B. cyclic synchronous position), bei denen die Fahrbefehle zyklisch und synchron an die Antriebe übertragen werden, sind heute in vielen verschiedenen ethernetbasierten Feldbussen wie z. B. EtherCAT realisiert.

EtherCAT ist der am weitesten verbreitete offene Feldbus für herstellerunabhängige Motion-Control-Systeme in der Automatisierung [25]. In Abbildung 2b zeigt Pfeil 1 eine Verbindung zwischen einer Modulsteuerung und einem Antrieb (Motorsteuerung), wo EtherCAT eingesetzt werden kann. Bei EtherCAT werden Ausgangsdaten vom EtherCAT-Master (üblich in der SPS oder im Motion-Controller) in einem gemeinsamen Ethernet-Frame an mehrere EtherCAT-Slave-Geräte (z. B. Antriebssysteme) gesendet. Die Kommunikation erfolgt in der Regel synchron mit der gleichen Zykluszeit wie die Abarbeitung der Motion-Control-Trajektorie durch den EtherCAT-Master in der SPS. Die mit EtherCAT vernetzten Antriebe synchronisieren ihre schnelleren Regelkreis-Aktualisierungsraten mit einer Phasenregelschleife (engl.: Phase-Locked Loop, PLL) auf den Motion-Controller bzw. den EtherCAT-Master-Buszyklus mit der Distributed-Clock-Funktionalität [26].

Für die Regelung von Motoren werden hochauflösende Winkel in der Antriebssteuerung benötigt. Dafür werden Motorfeedbacksysteme über digitale Schnittstellen mit den Antriebssteuerungen verbunden [27]. In den meisten Fällen handelt es sich dabei um RS-485-basierte Schnittstellen, mit Ausnahme von DRIVE-CLiQ, der internen Systemschnittstelle von Siemens zwischen den Motion-Control-Komponenten des Antriebssystems Sinamics S120.

Eine effiziente Anbindung intelligenter Sensoren und Aktoren bietet heute das in der IEC 61131-9 genormte Kommunikationssystemprotokoll IO-Link. Im Gegensatz zu einem Feldbus wird bei IO-Link nur ein Gerät (Sensor oder Aktor) über eine Punkt-zu-Punkt-Verbindung an einen IO-Link-Master-Port angeschlossen. Die IO-Link-Schnittstelle ist bezüglich der Spannungen zu den traditionellen 24 V E/As, wie sie in der IEC 61131-2 definiert sind, aufwärtskompatibel. Prozessdaten werden zyklisch übertragen, während Service- und Diagnosedaten nicht zyklisch übertragen werden.

Heutzutage ist es üblich, dass SPS-Programme zyklisch ausgeführt werden, unabhängig davon, ob sich Eingangssignale überhaupt verändern. Das gilt für alle fünf Programmiersprachen nach dem IEC 61131-3 Standard für Applikationsentwicklung [17]. Zusätzlich

ist es bei Mehrachs-Motion-Control-Anwendungen sinnvoll, dass die Feldbuskommunikation und die SPS nicht nur zyklisch, sondern idealerweise auch taktsynchron ausgeführt werden. Das Sercos III Interface und später auch EtherCAT nutzen die Bandbreite von Fast Ethernet (100 Mbit/s, 2 Leitungspaare) effizient und eignen sich für die taktsynchrone zyklische Steuerung. Bei PROFINET sind z. B. nur die bisher wenig verbreiteten Versionen Isochronous Real Time (IRT) und Time-Sensitive Networking (TSN), jeweils basierend auf GbE, für anspruchsvolle Motion-Control-Anwendungen geeignet. PROFINET IRT oder TSN sind für optimierte Robotikanwendungen zu komplex und zu teuer, da zusätzliche Netzwerkhardware, spezielle Switches, zusätzliche Konfigurations- und Optimierungsmaßnahmen sowie ein komplexer Softwarestack erforderlich sind. Nicht ohne Grund setzen z. B. ABB und KUKA EtherCAT für die interne Vernetzung ein, auch wenn die Roboter mit einer Standard PROFINET-Schnittstelle in eine Fertigungsautomatisierung eingebunden werden. Die streng zyklische Ausführung von SPS-Programmen nach IEC 61131-3 hat sich bei prozessnahen SPSen in der OT bewährt. Antriebe können dabei ohne Einschränkungen für den Anwender kostengünstig über EtherCAT vernetzt werden.

Wenn es bei der Vernetzung im industriellen Bereich um zuverlässige und stabile Verbindungen geht, sind kabelgebundene Lösungen bisher die erste Wahl. Wird jedoch eine hohe Mobilität gefordert, z. B. durch autonome mobile Systeme wie FTFs oder AMRs, empfiehlt sich der Einsatz einer funkbasierten Kommunikation. Neben Wireless Local Area Network (WLAN) bietet sich die Mobilfunktechnik 5G im industriellen Umfeld an. 5G steht für die fünfte Mobilfunkgeneration und verspricht schnelles mobiles Breitband, hochzuverlässige Netze mit sehr geringer Latenz und erfüllt die hohen Anforderungen der Vernetzung von mobilen Robotersystemen [28], [29]. Für die Steuerung von FTFs wurde in einer Zusammenarbeit von dem Verband der Automobilindustrie (VDA) und dem Verband Deutscher Maschinen- und Anlagenbauer (VDMA) mit der VDA5050 [30] ein Standard für die Kommunikation zwischen einem FTF-Flottenmanager und allen konformen FTFs entwickelt. Die Kommunikation erfolgt über drahtlose Netzwerke (5G, WLAN) mit Message Queueing Telemetry Transport (MQTT) als Nachrichtenprotokoll. Die Teilnehmer kommunizieren über das aus der IT bekannte PubSub-Modell, indem die Teilnehmer des Netzwerks die für sie relevanten Informationen bei einem Broker abonnieren (subscribe). Die einzelnen Fahrzeuge übertragen ihren Status (publish) bei einer Änderung. Bei der Leitsteuerung zur Orchestrierung einer FTF-Flotte ist eine zyklische Abarbeitung der Algorithmen nicht erforderlich. Vielmehr soll die Leitsteuerung auf die eingehenden Ereignisse der einzelnen FTFs schnell reagieren. Diese PubSub-Kommunikation erfolgt ereignisgesteuert, wodurch auch bei einer großen Anzahl von Fahrzeugen die Bandbreite der drahtlosen Kommunikation effizient genutzt wird.

Auch das Robot Operating System 2 (ROS 2), als Framework für die Entwicklung und Steuerung von Robotern und autonomen Systemen, kommuniziert ereignisgesteuert. Das von der Object Management Group (OMG) spezifizierte Data Distribution Service (DDS) wird zur PubSub-Kommunikation bei ROS 2 als Middleware eingesetzt [31]. MQTT und DDS harmonieren mit ihrem ereignisgesteuerten Ansatz mit der Norm IEC 61499 [32] für verteilte Steuerungssysteme. Für die Vernetzung von Submodulen als verteilte Automatisierungssysteme setzen sich zunehmend PubSub-Kommunikationsverfahren aus der IT durch und können in Zukunft Feldbusse wie PROFINET oder EtherNet/IP ersetzen [33]. In Abbildung 2b zeigen die Pfeile 2 und 3 eine solche Controller-to-Controller (C2C) Vernetzung und Pfeil 4 die Kommunikation von der Steuerungsebene zu höheren Ebenen.

2.2 Sicherheitsbezogene Steuerungen

2.2.1 Normen

Zunächst ist anzumerken, dass im englischen Sprachgebrauch beim Begriff Sicherheit zwischen Safety und Security unterschieden wird. Während Safety den Schutz von Menschen und Umwelt vor Maschinen behandelt, befasst sich Security mit dem Schutz von Systemen und Daten vor unberechtigtem Zugriff, Sabotage und Diebstahl. Im Rahmen dieser Arbeit wird der Begriff Sicherheit im Kontext von Safety verwendet.

Alle Maschinen, die im europäischen Wirtschaftsraum in Betrieb genommen werden, müssen seit Januar 1995 die grundlegenden Anforderungen der Maschinenrichtlinie einhalten [34]. Normen haben zunächst keine unmittelbare Rechtswirkung, erhalten diese aber durch die Veröffentlichung im Amtsblatt der EU oder durch nationale Gesetze und Verordnungen, die auf diese Normen verweisen. Für die funktionale Sicherheit ist insbesondere die Norm DIN EN ISO 12100 zu nennen, da hier eine Methodik zur Erarbeitung einer geeigneten Sicherheitsstrategie für die Maschinenkonstruktion vorgestellt wird, die auch die Gestaltung sicherheitsbezogener Teile von Steuerungen (engl.: Safety-Related Part of a Control System, SRP/CS) einbezieht [35]. Bei SRP/CS wird auf sicherheitsbezogene Eingänge reagiert und sicherheitsbezogene Ausgänge werden generiert. Es handelt sich dabei um den Teil des Steuerungssystems einer Maschine, der gefährliche Zustände verhindert. SRP/CS sind für die Ausführung von Sicherheitsfunktionen ausgelegt und müssen unter allen vorhersehbaren Bedingungen ordnungsgemäß funktionieren. Eine Sicherheitsfunktion ist eine Funktion einer Maschine, deren Ausfall zu einer unmittelbaren Erhöhung des Risikos führen kann [35]. Eine Sicherheitsfunktion, die durch ein SRP/CS einer Maschine realisiert wird, hat die Aufgabe, die Maschine in Bezug auf bestimmte

Gefährdungen in einem sicheren Zustand zu halten. Ein SRP/CS in Form einer Sicherheitssteuerung ist in der Regel eine SPS mit speziellen Eigenschaften, um das geforderte Maß an Sicherheit und Verfügbarkeit zu gewährleisten. Im Folgenden wird der Begriff Sicherheits-SPS verwendet, wenn ein SRP/CS speziell als Sicherheitssteuerung betrachtet wird.

Die Normenreihe IEC 61508 und ihre Sektornorm IEC 62061 (Stand 2016) für die Maschinenindustrie beziehen sich auf elektrische, elektronische und programmierbare elektronische Systeme [36]. Als Schema zur Klassifizierung von Sicherheitsfunktionen verwenden die beiden Normen die Sicherheitsanforderungsstufe (engl.: Safety Integrity Level, SIL). Aufbauend auf der grundlegenden Norm DIN EN ISO 12100 beschreibt die Nachfolgenorm der EN 954, die ISO 13849-1, die erforderliche Risikominderung bei der Entwicklung und der Integration von SRP/CS und Schutzeinrichtungen. Diese können elektrisch, elektronisch, hydraulisch, pneumatisch oder mechanisch sein. Mit der ISO 13849-1 wird also eine allgemein anwendbare Systematik für ein SRP/CS vorgelegt. Die Norm behandelt die Risikominderungsstrategie der ISO 12100 und beschreibt den Beitrag des sicherheitsbezogenen Steuerungssystems zur Risikominderung. Um die erforderliche Risikominderung zu erreichen, stellt das SRP/CS Sicherheitsfunktionen mit einem Performance Level (PL) bereit. Im Gegensatz zu den Normen IEC 62061 und 61508 wird in der ISO 13849-1 der Begriff SIL nicht verwendet. Der in der Norm beschriebene PL erweitert den aus der EN 954 bekannten Kategorienbegriff. Die sicherheitsbezogenen Architekturen sind nun vielfältiger einsetzbar, da mit dem PL Kombinationen verschiedener Steuerungsstrukturen mit unterschiedlichen Technologien einfacher zu realisieren sind.

Sowohl die ISO 13849-1 als auch die IEC 62061 behandeln sicherheitsbezogene Steuerungssysteme. Beide Normen können miteinander kombiniert werden, da sie trotz unterschiedlicher Verfahren zu ähnlichen Ergebnissen führen. Der Ansatz der IEC-Normen zur funktionalen Sicherheit, IEC 61508 und IEC 62061, Ausfallwahrscheinlichkeiten und nicht Strukturen als Kenngrößen zu definieren, erscheint zunächst allgemeingültiger. Bis 2023 war die ISO 13849-1 die einzige harmonisierte Norm für den Maschinensektor, die dem Anwender ermöglicht hat, Sicherheitsfunktionen von der Sensorebene bis hin zur Aktorebene technologieunabhängig zu konzipieren und zu evaluieren [37]. Die aktuelle Version der IEC 62061 (Stand 2023) hat dies aufgegriffen und bietet nun ebenfalls den technologieunabhängigen Ansatz [38].

Sowohl die ISO 13849-1 als auch die IEC 62061 sind harmonisierte Normen im Sinne der Maschinenrichtlinie. Obwohl die Teile 1 bis 4 der IEC 61508 aus Sicht der IEC den Status

einer Sicherheitsgrundnorm haben, kann sie nicht unter der Maschinenrichtlinie harmonisiert werden [37]. Um mit der IEC-Welt kompatibel zu sein und eine langfristige Zusammenführung beider Normenwelten zu ermöglichen, unterstützt die ISO 13849-1 sowohl den deterministischen Ansatz der Kategorien und Architekturen als auch mit der PL-Definition den Ansatz der sicherheitstechnischen Zuverlässigkeit. Die vereinfachte quantitative Herangehensweise, unter Verwendung der vorgestellten Architekturen, stellen aus Anwendungssicht Gründe dar, die für die Entscheidung zugunsten der ISO 13849-1 zur Umsetzung von Sicherheitsfunktionen im Maschinensektor sprechen. Werden Sicherheitskomponenten auch für andere Branchen als der Maschinenautomatisierung entwickelt, kann neben der ISO 13849-1 die IEC 61508 als Grundlage für eine Entwicklung herangezogen werden.

2.2.2 Sicherheitssteuerungen

Auch lange nach der Erfindung der SPS wurden die Sicherheitsfunktionen noch fest verdrahtet. Eine SPS galt dabei als nicht ausreichend zuverlässig in Bezug auf Sicherheitsfunktionen. 1987 brachte Pilz das erste Not-Halt-Relais PNOZ 1 zum Schutz von Menschen und Maschinen auf den Markt. Das Sicherheitsschaltgerät überwacht neben der klassischen Not-Aus-Funktion auch Schutztüren, Lichtschranken, Zweihandbetätigungen und viele weitere Sicherheitsfunktionen. Die kontinuierliche Entwicklung führte von dedizierten Sicherheitsgeräten zur 2002 eingeführten frei konfigurierbaren Sicherheits-SPS PNOZmulti. Erstmals war es möglich, Sicherheitsfunktionen mit Redundanz einfach per Drag&Drop über eine grafische Benutzeroberfläche zu erstellen [39].

Trennung von Anwendungen unterschiedlicher Kritikalität

Sicherheitsfunktionen sind heute integraler Bestandteil von Automatisierungslösungen. Für ein Automatisierungskonzept mit funktionaler Sicherheit werden üblicherweise getrennte Steuerungen für sicherheitsbezogene und nicht sicherheitsbezogene Anwendungen eingesetzt. Dabei ist es üblich, dass das Anwendungsprogramm auf einer Standard-SPS ausgeführt wird und über einen oder mehrere Standard-Feldbusse mit anderen Komponenten kommuniziert. Gleichzeitig wird die Sicherheitsanwendung auf einer separaten Sicherheits-SPS ausgeführt und die sicherheitsbezogenen Komponenten können über einen zusätzlichen sicherheitsbezogenen Feldbus angebunden werden. Dieser Ansatz bringt einige Nachteile mit sich, zu denen unter anderem eine komplexe Struktur, der hohe Verdrahtungs- und Installationsaufwand sowie erhöhte Kosten gehören.

Aufgrund der genannten Nachteile und den steigenden Anforderungen an die funktionale Sicherheit, ist es sinnvoll Systeme zu entwickeln, bei denen das Sicherheits- und das Anwendungsprogramm auf dem gleichen Gerät ausgeführt werden. Mit dem Gray- bzw. Black-Channel-Prinzip und somit der Datenübertragung von sicherheitsbezogenen Prozessdaten über Standard-Feldbusse, ist es möglich innerhalb eines gemeinsamen Feldbus-Netzwerks mit sicherheitsbezogenen und nicht sicherheitsbezogenen Komponenten zu kommunizieren. Abbildung 3 zeigt auf der linken Seite das gängige Sicherheitskonzept, bei dem die sicherheitsbezogenen und nicht sicherheitsbezogenen Komponenten voneinander getrennt sind. Auf der rechten Seite wird zum Vergleich das integrierte Sicherheitskonzept gezeigt, bei dem eine sogenannte Compound-SPS sowohl das Sicherheits- als auch das Anwendungsprogramm ausführt. Die Kommunikation mit den sicherheitsbezogenen Automatisierungskomponenten basiert hier auf dem Gray- bzw. Black-Channel-Prinzip, wie es beispielsweise bei EtherCAT/Failsafe over EtherCAT (FSoE) oder PROFINET/PROFIsafe verwendet wird.

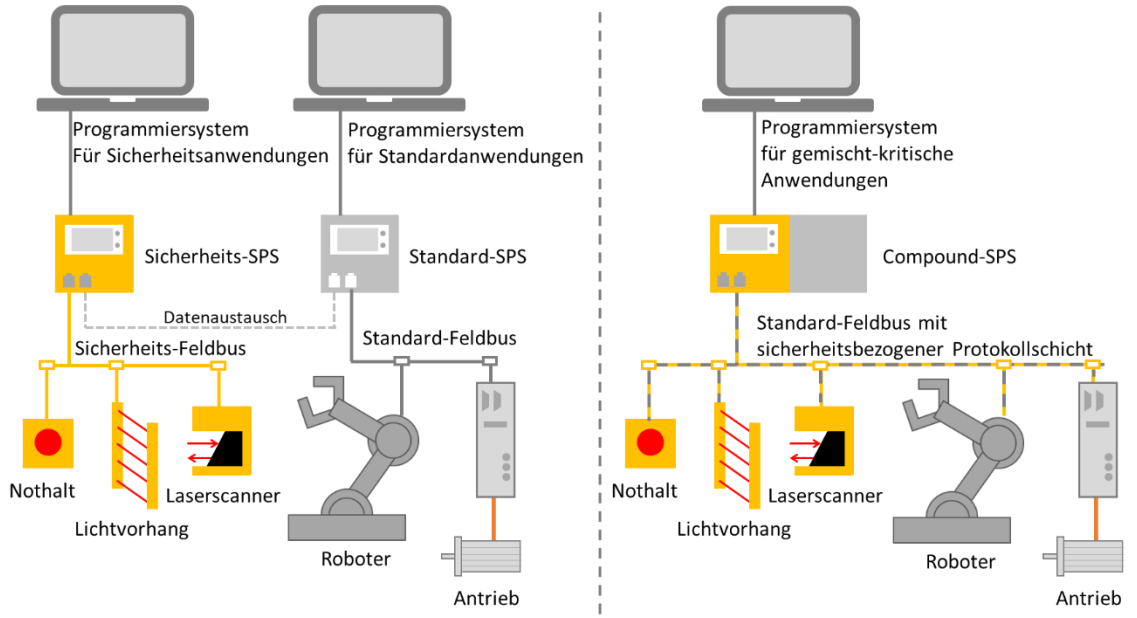


Abbildung 3: Vergleich des gängigen Sicherheitskonzepts (links) und des integrierten Sicherheitskonzepts (rechts).

Bei einer Compound-SPS wird in der Regel das gleiche Programmiersystem verwendet, um sowohl die sicherheitsbezogene als auch die nicht sicherheitsbezogene Applikation zu programmieren [40]. Die Rückwirkungsfreiheit zwischen nicht sicherheitsbezogener Standard-SPS und der Sicherheits-SPS ist für die Realisierung einer Compound-SPS von großer Bedeutung. Ein klassischer Ansatz ist die Integration von zwei getrennten Steuerungen unterschiedlicher Kritikalität in einem Gehäuse.

Architekturen von Sicherheitssteuerungen

Die Ansätze der Hersteller von sicherheitsbezogenen Automatisierungskomponenten sind nach wie vor unterschiedlich, wobei jeder Ansatz nach den Normen IEC 61508, 62061 oder ISO 13849-1 ausgelegt sein muss, um drei Sicherheitsmaßnahmen zu erfüllen: technische Zuverlässigkeit, Redundanz und Diagnosedeckungsgrad (engl.: Diagnostic Coverage, DC). Die Toleranz gegenüber Fehlern wird durch die Struktur oder Architektur eines SRP/CS bestimmt und stellt die Basis dar, auf dem alle quantifizierbaren Aspekte aufbauen, um den PL oder SIL des SRP/CS zu bestimmen. Die Anwendungen zeigen, dass es nur wenige Grundtypen von SRP/CS im Maschinenbau gibt, auf die sich der Großteil aller Steuerungen abbilden lässt. Zu den Grundtypen von SRP/CS gehören nach der ISO 13849-1 einkanalige ungetestete Systeme mit unterschiedlich zuverlässigen Bauteilen, das durch Tests aufgewertete System und schließlich das zweikanalige getestete hochwertige System.

Eine klassische, oft verwendete Sicherheitsarchitektur für Steuerungssysteme ist die zweikanalige Ausführung mit homogener Redundanz, bei der in der Regel zwei identische Mikrocontroller eingesetzt werden [41]. Das zweikanalige Blockschaltbild für Kategorie 3 ist in der Abbildung 4 dargestellt. Unter der IEC 61508 wird eine solche zweikanalige Architektur als 1oo2 (aus dem Englischen: one out of two) bzw. mit erhöhter Diagnose 1oo2D bezeichnet. 1oo2 beschreibt ein fehlertolerantes System, bei dem ein Fehler in einem Kanal nicht zum Verlust der Sicherheitsfunktion führt.

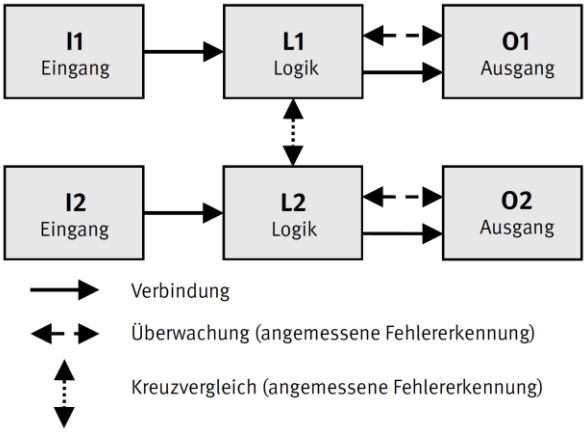


Abbildung 4: Zweikanalige Kategorie 3 Architektur nach ISO 13849-1 [37].

Bei einem Kategorie 3 System wird nach der ISO 13849-1 ein durchschnittlicher DC (DC_{avg}) von niedrig (60 % bis 90 %) bis mittel (90 % bis 99%) gefordert. Um den geforderten DC_{avg} zu erreichen, wird unter anderem eine Ausführung einer Prozessor Selbsttest-Bibliothek (engl.: Self-Test-/Software-Test-Library, STL) verwendet [37], [41], [42]. Für

Entwickler von Sicherheitssteuerungen, die auf komplexen Prozessorsystemen basieren, ist es ohne detaillierte Kenntnisse des internen Aufbaus kaum möglich, diese STL selbst zu erstellen. Aus diesem Grund bieten die Hersteller in der Regel eine entsprechende sicherheitszertifizierte STL zusammen mit den notwendigen Sicherheitskennzahlen an. Die Anforderungen an die Ausführung der Selbsttests hängen von der Anwendung und den Anforderungen an die funktionale Sicherheit des Systems ab. Im Allgemeinen sollen die Selbsttests jedoch in der Lage sein, die Funktion des Systems zu überwachen und fehlerhafte Operationen zu erkennen und ggf. zu korrigieren. Die Ausführung der Selbsttests kann je nach System- und Sicherheitsanforderungen zyklisch oder im Hintergrund in Zeitscheiben erfolgen. In Systemen mit Echtzeitanforderungen können die Selbsttests im Hintergrund ausgeführt werden, um den Betrieb des Systems nicht zu beeinträchtigen. In diesem Fall werden die Tests über mehrere Zyklen durchgeführt [41].

Der klassische Ansatz eines Kategorie 3 Systems mit zwei redundanten Kanälen wird z. B. von den Kooperationspartnern Berghof Automation, KEB Automation, Kendrion Kuhnke Automation und Lenze genutzt, die gemeinsam eine Sicherheits-SPS auf Basis des CODESYS SIL3 Laufzeitsystems entwickelt haben [43]. Abbildung 5 zeigt eine mögliche Architektur einer Compound-SPS mit einem zweikanaligen sicherheitsbezogenen Teil auf Basis von zwei Mikrocontrollern.

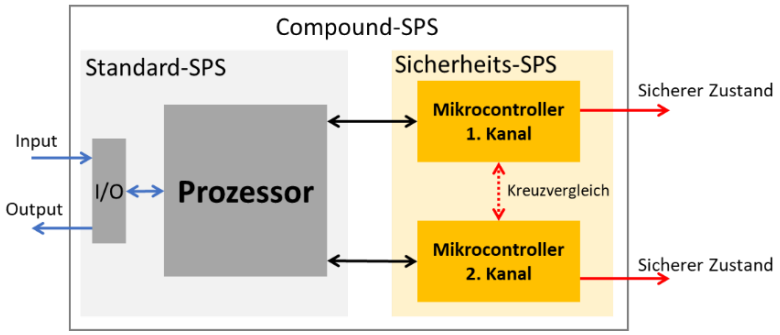


Abbildung 5: Compound-SPS mit zwei Steuerungen unterschiedlicher Kritikalität.

Das Sicherheits-SoC HICore1 der Firma HIMA bietet ein komplettes zertifiziertes sicherheitsbezogenes System auf einem einzigen Chip [44]. Um SIL3 Anforderungen zu erfüllen, besteht der HICore1 im Wesentlichen aus einem Sicherheitssystem und einem Kommunikationssystem. Während das Sicherheitssystem auf einer redundanten 1oo2D-Zweiprozessorarchitektur mit einer On-Chip-Diagnoseeinheit basiert, verwendet das nicht zertifizierte Kommunikationssystem einen dritten Prozessor, der als Black-Channel dient.

Sicherheitsbezogene Steuerungen in der Industrie bestehen auch aus Standardkomponenten [37]. Dies ist insbesondere dann möglich, wenn die zweikanalige

Sicherheitsstruktur mit der diversitären Redundanz realisiert wird [45]. Diese Standardkomponenten sind z. B. nach ISO 9001 qualitätsgesichert und müssen nicht sicherheitszertifiziert werden. Die diversitäre Redundanz erhöht die Fehlererkennung auf Systemebene im Vergleich zur homogenen Redundanz erheblich. Diversitäre Technologien in beiden Kanälen verringern die Wahrscheinlichkeit eines gefährlichen Ausfalls des SRP/CS aufgrund eines Hardware- oder Softwarefehlers. Die Einführung von Diversität in Software und Hardware verringert das Risiko von Fehlern gemeinsamer Ursache (engl.: Common Cause Failure, CCF) und ein Kreuzvergleich von diversitären Kanälen erhöht den DC. Das Institut für Arbeitsschutz der Deutschen gesetzlichen Unfallversicherung (IFA) listet in [37] Beispiele auf, in denen die technologische Diversität gegeben ist:

- Ein Kanal enthält Komponenten mit Embedded-Software, der zweite Kanal enthält nur Komponenten ohne Embedded-Software wie z. B. mechanische, pneumatische oder hydraulische Bauteile.
- Beide Kanäle bestehen aus Komponenten mit diversitärer Embedded-Software, beispielsweise zwei verschiedenen Betriebssystemen.
- Beide Kanäle verwenden unterschiedliche Hardware, beispielsweise unterschiedliche Prozessoren.

Mit diversitärer Redundanz unter Verwendung von qualitätsgesicherten Standardkomponenten lassen sich sicherheitsbezogene Systeme bis PL d ohne eine aufwendige Zertifizierung realisieren. PL e ist ohne eine Zertifizierung nicht möglich. Ein Beispiel mit diversitärer Redundanz ist die Compound-SPS RFC 4072S von Phoenix Contact. Sie basiert beim sicherheitsbezogenen Teil auf einer zertifizierten Kategorie 4 Architektur mit zwei diversitären Mikrocontrollern (ARM Cortex-A9 und ARM Cortex-A8) [46].

Auf der Automatica 2012 stellte der Industrieroboter Hersteller KUKA die Kompaktsteuerung KR C4 für Kleinroboter vor. Ein einziger Intel Core 2 Duo Prozessor wurde verwendet, um komplexe Sicherheitsalgorithmen zusammen mit nicht sicherheitsbezogenen Roboteralgorithmen mit einer Aktualisierungsrate von 8 kHz auszuführen. Ein Betriebssystem für das HMI und ein RTOS teilen sich einen Kern des Dual-Core Prozessors unter der Echtzeit-Virtualisierungstechnologie von KUKA [47]. Die integrierte Sicherheitssteuerung nutzt den anderen Kern des Prozessors für beide sicherheitsbezogenen Kanäle. Nach inzwischen geltenden Sicherheitsnormen kann diese Architektur so nicht mehr als Kategorie 3 oder äquivalent zertifiziert werden. Daher ist dieser Ansatz für Robotikanwendungen heute nicht mehr anwendbar.

Ein weiteres Beispiel von diversitärer Redundanz in der Robotik ist das Drei-Wege-Vergleichsverfahren von ABB corporate research, bei dem ein Multi-Core Prozessor mit integrierter Graphics Processing Unit (GPU) in Verbindung mit einer externen Testeinrichtung (TE) eingesetzt wird [48]. Die GPU als diversitärer Co-Prozessor erhöht hierbei die Sicherheitsintegrität. Neben der Programmiersprache C wird die Open Computing Language (OpenCL) verwendet, ein Framework für Software, das auf heterogener Hardware wie einer Multi-Core Central Processing Unit (CPU), GPU oder digitalen Signalprozessoren ausgeführt werden kann.

In [49] wird Coded Processing vorgestellt. Die notwendigen kodierten Anweisungen des Coded Processings werden durch einen zertifizierten Compiler automatisch in einen Kanal der Anwendungssoftware eingefügt. Dadurch entsteht eine diversitäre Software, die auf einer Standard-IPC-Hardware ausgeführt werden kann. Bei Siemens wird durch die Integration des Coded Processings eine sicherheitsbezogene SPS sowie eine Standard-SPS auf einer Siemens-IPC Hardware bis SIL3 realisiert.

Auf der SPS – Smart Production Solutions 2022 in Nürnberg stellte Silistra gemeinsam mit CODESYS eine Lösung vor, bei der das SIL3 Laufzeitsystem von CODESYS auf einer Standard-Hardware mit Coded Processing ausgeführt wird [50]. Sicherheits-E/As können nur über ein sicherheitsbezogenes Protokoll, z. B. PROFINET/PROFIsafe oder EtherCAT/FSoE angebunden werden. Ein Nachteil des Coded Processings ist der erhebliche zusätzliche Rechenaufwand, gemessen als Performance Overhead [51]. Dies mindert bei komplexen Algorithmen die Energieeffizienz der Sicherheitsfunktionen.

In der Automobilindustrie, wo sicherheitsrelevante Systeme mit hoher Rechenleistung erforderlich sind, werden in der Regel Dual-Core Lockstep Prozessoren mit integrierter Gleitkommaeinheit (engl.: Floating-Point Unit, FPU) in Kombination mit einer externen TE (auf separatem Silizium-Die) eingesetzt [52], [53]. Bei Lockstep Prozessoren, die ursprünglich 2008 von Texas Instruments entwickelt wurden, handelt es sich um spezielle Multi-Core Prozessoren, um die Fehlererkennung und Fehlertoleranz zu erhöhen. Die Lockstep-Methode leitet sich vom englischen Begriff lockstep marching (Gleichschritt) ab und bedeutet, dass zwei Kerne des Multi-Core Prozessors exakt das gleiche Programm ausführen. Die so erzielte Redundanz hilft bei der Ermittlung von hardwareseitigen Ausfällen in einem der Prozessorkerne. Um CCFs entgegen zu wirken, können die einzelnen Kerne des Prozessors um einige Taktzyklen versetzt betrieben werden. So wirkt sich der CCF in unterschiedlichen Zuständen der einzelnen Kerne aus und lässt sich anschließend in der Kontrolle und Bewertung der Ergebnisse beider Kerne feststellen. Die für den Au-

tomobilsektor entwickelten Lockstep Prozessoren werden demzufolge auch nach der Sicherheitsnorm ISO 26262 für sicherheitsrelevante elektrische/elektronische Systeme in Kraftfahrzeugen und der Sicherheitsgrundnorm IEC 61508 zertifiziert [54].

Lockstep Prozessoren lassen sich nicht auf die vorgesehenen Architekturen der ISO 13849-1 abbilden, da ein klassisches zweikanaliges System aus zwei getrennten CPUs oder Mikrocontrollern besteht. Bei einem Lockstep Prozessor, bei dem die Redundanz auf einem Chip realisiert wird (Single-Chip), sind zusätzliche Maßnahmen notwendig, um eine äquivalente Risikoreduzierung wie ein Kategorie 3, PL d System zu erreichen. Bei einem SRP/CS mit einem Lockstep Prozessor und externer TE wird für eine äquivalente Risikoreduzierung gemäß Kategorie 3, PL d ein DC_{avg} von hoch (ab 99 %) anstelle von niedrig bis mittel (60 % bis 99 %) gefordert [55]. Abbildung 6 zeigt die Sicherheitsarchitektur bestehend aus einem Lockstep Prozessor mit On-Chip Sicherheitsmaßnahmen wie einer STL und einer externen Versorgungs- und Überwachungseinrichtung, im Folgenden ebenfalls als TE bezeichnet.

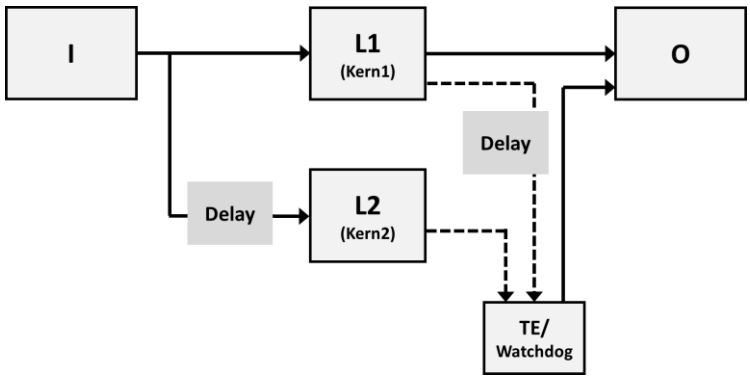


Abbildung 6: Architektur eines Lockstep Prozessors mit externer Testeinrichtung.

Die integrierte hardwarebasierte Fehlererkennung des Lockstep Prozessors decken die in den Normen geforderten Anforderungen ab und ermöglichen die Entwicklung von SRP/CS auch für andere Anwendungsfelder wie der Industrie und der Medizintechnik [56]. Die Fehlererkennung, die durch die Hardware erfolgt, reduziert zusätzlich die erforderlichen Sicherheitsmaßnahmen durch Softwarealgorithmen wie einer STL im zyklischen Betrieb. Lockstep Prozessoren gibt es von vielen Herstellern, unter anderem Infineon, Texas Instruments, Renesas und Xilinx. Xilinx bietet mit dem Zynq Ultrascale+ ein Multi-Prozessor SoC mit einem Dual-Core Lockstep ARM Cortex-R5F Prozessor, einem Quad-Core ARM Cortex-A53 Prozessor und einem Field Programmable Gate Array (FPGA). Das Multi-Prozessor SoC ist bis SIL2 zertifiziert und bietet sich aufgrund der unterschiedlichen Prozessoren für den Einsatz in einer Compound-SPS an [15].

Für die Realisierung von Sicherheitssteuerungen werden heute überwiegend ARM Cortex Prozessoren eingesetzt. Dies ist unter anderem darauf zurückzuführen, dass sie von vielen verschiedenen Herstellern angeboten werden und sich Unternehmen bei der Entwicklung von SRP/CS nicht von einem Hersteller abhängig machen wollen.

Intel bietet mit dem Intel Atom x6427FE Quad-Core SoC mit dem Codenamen Elkhart Lake eine Single-Chip-Lösung, die für die funktionale Sicherheit bis SIL2, Kategorie 3 und PL d zertifiziert ist [57]. Bei Verwendung von zwei x6427FE SoCs (Dual-Chip) kann SIL 3, Kategorie 4 und PL e erreicht werden. Das Quad-Core SoC zeichnet sich dadurch aus, dass es mit einem vergleichbaren Lockstep-Ansatz zwei Kerne für die funktionale Sicherheit und zwei Kerne für nicht sicherheitsbezogene Anwendungen verwendet. Das SoC ist daher für den Einsatz in einer Compound-SPS geeignet. Ebenso wie beim Lockstep Prozessor ist eine externe TE auf einem separaten Silizium-Die erforderlich, die das System im Fehlerfall in den sicheren Zustand überführt. Die Single-Chip-Lösung des x6427FE lässt sich ebenfalls nicht direkt auf eine Architektur der ISO 13849-1 abbilden. Um eine äquivalente Risikoreduzierung nach Kategorie 3, PL d zu erreichen, ist ein DC_{avg} von hoch (mind. 99%) erforderlich [58]. Die Architektur des Intel x6427FE Quad-Core SoCs für gemischt-kritische Anwendung ist in der Abbildung 7 dargestellt.

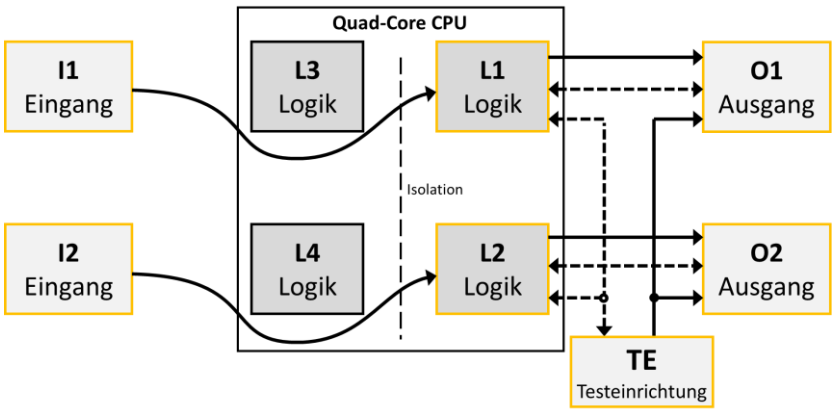


Abbildung 7: Architektur eines Quad-Core SoCs für gemischt-kritische Anwendungen.

Sicherheitsbezogene Rechenleistung

Die Leistung verschiedener Sicherheitssteuerungen und deren Architektur sind schwer zu vergleichen, da Benchmark-Ergebnisse von den Herstellern meist nicht veröffentlicht werden. Die weit verbreitete Sicherheits-SPS PNOZmulti 2 von Pilz verarbeitet sicherheitsbezogene digitale Signale mit einer Zykluszeit von 10 ms [39]. Die Feldbuszykluszeit der Servoverstärker-Sicherheitskarte AX5805 beträgt 15 ms [59]. Lange Zeit wurden solche Zykluszeiten in der Automatisierung als ausreichend angesehen, um das sichere Abschalten des Drehmoments beispielsweise über Lichtvorhänge oder einen Not-Halt zu

steuern. Jedoch ändern sich die Anforderungen an sicherheitsbezogene Software durch den menschenzentrierten Ansatz und dem einhergehenden Einsatz von FTFs, AMRs und Cobots.

Benchmarks nach Dhrystone gibt es für fast alle Prozessorarchitekturen [60]. Die Angabe des Benchmarks in Dhrystone Millionen Instruktionen pro Sekunde (DMIPS) ist heute weit verbreitet. Eine andere Darstellung ist DMIPS/MHz bzw. Dhrystone Instructions per cycle, was einen einfacheren Vergleich von Prozessorarchitekturen ermöglicht.

Tabelle 1 zeigt gemäß [61] die Schätzung der verfügbaren sicherheitsbezogenen DMIPS für verschiedene Prozessorarchitekturen. Die Prozessorarchitektur steht in der ersten Spalte. Die DMIPS-Werte pro Kern in der vierten Spalte werden aus dem Produkt der Taktrate (zweite Spalte) und der spezifischen Rechenleistung pro Prozessorkern in DMIPS/MHz (dritte Spalte) berechnet. In der fünften Spalte sind die Lockstep-Eigenschaften vermerkt. Die letzte Spalte der Tabelle 1 zeigt die verfügbare Rechenleistung für komplexe Sicherheitsfunktionen in sicherheitsbezogenen DMIPS. Der Wert errechnet sich aus dem Wert in der vierten Spalte und berücksichtigt ggf. die Zeit für die Ausführung der STL. Die Zeilen der Tabelle 1 sind so sortiert, dass die Werte der verfügbaren Sicherheits-DMIPS in aufsteigender Reihenfolge stehen.

Die AURIX-Prozessorfamilie von Infineon wurde für Anwendungen im Automobilbereich entwickelt und ist mit vielen Sicherheitsmerkmalen ausgestattet, darunter auch die Lockstep-Erweiterung. Die für Echtzeitanwendungen entwickelte ARM Cortex-R5 Prozessorarchitektur ist auch optional mit Lockstep-Erweiterung z. B. von Texas Instruments und Xilinx erhältlich.

Bei Systemen, die in regelmäßigen Abständen neu gestartet werden, ist es in der Regel ausreichend, wenn die Selbsttests nur in der Boot-Sequenz ausgeführt werden [41]. Fahrzeuge werden regelmäßig aufgetankt oder aufgeladen, die Fahrer benötigen eine Pause oder das Ziel wird erreicht und die Fahrt beendet. Steuerungssysteme in der Industrie hingegen werden teilweise 24 Stunden am Tag, 7 Tage die Woche betrieben. Da bei Lockstep Prozessoren ein Großteil der Fehlererkennung durch die Hardware erfolgt, reduzieren sich die zusätzlich erforderlichen Sicherheitsmaßnahmen durch Software, wie z. B. die Implementierung einer während des Betriebs zyklisch ausgeführten STL. In industriellen Anwendungen ist die Implementierung der STL wesentlich komplizierter, da sie während des Betriebs zyklisch ausgeführt werden muss. Tabelle 1 zeigt in der vorletzten Spalte einen typischen Wert von ca. 30 % der Rechenzeit für die STL im Betrieb [56].

Der Intel Core i7 nimmt eine Sonderstellung ein, da bisher alle Implementierungen in der Automatisierung auf Coded Processing für Sicherheitsfunktionen setzen. Nach [51] muss

für Coded Processing im Durchschnitt ein Overhead-Faktor von 17 berücksichtigt werden. Für Anwendungen mit einem signifikanten Anteil komplexer sicherheitsrelevanter Algorithmen ist Coded Processing mit Hochleistungsprozessoren nicht sinnvoll, da bei vergleichbar sicherheitsrelevanten DMIPS die Hardware teurer ist und auch deutlich mehr Leistung verbraucht. Der Intel Core i7 Prozessor benötigt aufgrund der erhöhten Leistungsaufnahme in der Regel einen Lüfter, der im industriellen Umfeld nicht erwünscht ist. Das Intel x6427FE SoC bietet mit 7980 DMIPS die höchste sicherheitsrelevante Rechenleistung und eignet sich als Single-Chip-Lösung für gemischt-kritische Anwendungen. Der gemessene DMIPS Benchmark für einen Kern des Intel x6427FE SoCs basiert auf der GCC Compilerversion 11.4 mit dem Linux Kernel in der Version 5.4.143.

Tabelle 1: Sicherheitsbezogene Rechenleistung unterschiedlicher Prozessoren.

CPU Architektur	Taktrate [MHz]	DMIPS/ MHz pro Kern	DMIPS pro Kern	Lockstep	Selbsttest-Bibliothek (STL)	Sicherheits-DMIPS pro Kern
Cortex-M4 [62]	180	1,25	225		30 %	157,5
AURIX [63]	300	2	600	x	-	600
Intel Core i7	2900	~ 10	29000		30 %	1075
Cortex-A9 [64]	800	2,5	2000		30 %	1400
Cortex-R5 [65]	800	2,1	1680	optional	-	1680
Intel x6427FE	1900	6	11400	vergleichbar	30 %	7980

2.2.3 Kommunikation

Die IEC 61158 und ihre Begleitnormen IEC 61784-1 und 61784-2 definieren gemeinsam Kommunikationsprotokolle, die die dezentrale Steuerung von Applikationen in der Automatisierungstechnik ermöglichen. Die Feldbustechnik sowie der Einsatz von Mikrocontrollern hat sich inzwischen in der Praxis bewährt und hat dabei in der Automatisierungstechnik hohe Akzeptanz gefunden.

Die IEC 61784-3 definiert zunächst grundlegende Prinzipien zur Erfüllung der Anforderungen der Hauptnorm IEC 61508 für die funktional sichere Übertragung. Dabei werden

Übertragungsfehler berücksichtigt und Maßnahmen getroffen, um diese aufzudecken. Diese Maßnahmen zur Einrichtung einer sicherheitsbezogenen Datenübertragung werden in einer zusätzlichen Sicherheitskommunikationsschicht (engl.: Safety Communication Layer, SCL) ausgeführt [66]. Dieser SCL hat die Aufgabe, die sicherheitsbezogenen Daten in eine Safety Protocol Data Unit (SPDU) zu kodieren und diese an den Black-Channel weiterzugeben bzw. eine SPDU vom Black-Channel zu empfangen, diese zu dekodieren, um die sicherheitsbezogenen Daten zu extrahieren und der Sicherheitsanwendung zur Verfügung zu stellen (siehe Abbildung 8).

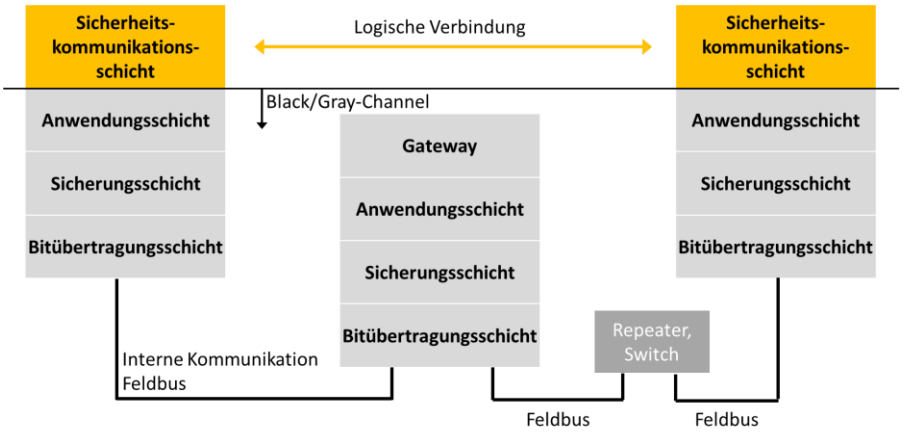


Abbildung 8: Funktionsweise vom Black-Channel.

Der Black-Channel-Ansatz hat den Vorteil, dass das sicherheitsbezogene Protokoll im Standard-Feldbus integriert ist. Dies ist für den Anwender besonders komfortabel, da über einen Feldbus sowohl sicherheitsbezogene als auch Standard-Daten übertragen werden können. Der Black-Channel besteht aus der zugrunde liegenden Kommunikationsschicht des Feldbusses unterhalb des SCL. Fehler innerhalb des Channels können verschiedene Ursachen haben. Beispiele sind Veränderung von Bits in den Nachrichten, zufällige Hardware Fehler oder systematische Fehler in der Elektronik. Ein sicherheitsbezogenes Protokoll muss nachweisen, dass für jeden möglichen Fehler mindestens eine geeignete Sicherheitsmaßnahme zur Fehlererkennung vorhanden ist [41], [66].

Beim White-Channel-Ansatz ist für sicherheitsbezogene Daten ein eigener sicherheitsbezogener Feldbus notwendig, der parallel zum Standard-Feldbus implementiert wird. Die gesamte Datenübertragung beim White-Channel wird nach der IEC 61508 implementiert.

Die Bitfehlerwahrscheinlichkeit aus der IEC 61784-3 von 0,01 (jedes 100. Bit kann falsch sein) sind für verschlüsselte, komprimierte oder modulierte Kommunikationswege nicht ausreichend, weshalb die geplante fünfte Ausgabe der IEC 61784-3 zwischen der Black-

und Gray-Channel-Kommunikation unterscheidet [67]. Für die zukünftige Black-Channel-Übertragung soll eine deutlich erhöhte Bitfehlerwahrscheinlichkeit von 0,5 (jedes zweite Bit kann falsch sein) angenommen werden [67], [68]. Die bisher geforderte Bitfehlerwahrscheinlichkeit von 0,01 soll in Zukunft für die neue Gray-Channel-Übertragung mit folgenden Einschränkungen verwendet werden können:

- Keine Verschlüsselung (z. B. keine Virtual Private Networks (VPN))
- Keine Modulation (z. B. kein WLAN)
- Keine Kompression (z. B. kein ZIP)

2.2.4 Bewegungsüberwachung

Zur Drehzahlregelung von elektrischen Antrieben in Maschinen werden seit vielen Jahren Antriebssteuerungen wie Gleichstromsteller, Frequenzumrichter oder Servoregler eingesetzt. Diese Antriebssteuerungen sind in der Regel mit gefahrbringenden Bewegungen an den Maschinen verbunden. Im Automatikbetrieb wird der Zugang zum Gefahrenbereich durch trennende oder berührungslos wirkende Schutzeinrichtungen überwacht. Es gibt jedoch Situationen, wie z. B. den Einrichtbetrieb an einer Werkzeugmaschine, in denen auch bei laufender Maschine und damit bei aufgehobener Funktion der trennenden Schutzeinrichtungen gearbeitet wird. Auch bei der Kollaboration von Menschen und Robotern wird im direkten Umfeld einer laufenden Maschine gearbeitet. Hierbei sind Maßnahmen notwendig, die dem Bediener in solchen Situationen ausreichend Schutz bieten [69].

Das Ziel der Sicherheitstechnik ist stets die Verhinderung von gefahrbringenden Bewegungen. Demzufolge war es ein logischer Schritt, die Sicherheitstechnik eng mit der Bewegungssteuerung zu verknüpfen. Ein Frequenzumrichter oder Servoregler ist in der Regel zunächst eine nicht sicherheitsrelevante Komponente. Die Sicherheit wird hier durch zusätzliche sicherheitsbezogene Komponenten gewährleistet, die den Motor im Fehlerfall in einen drehmomentlosen Zustand versetzen bzw. die Bewegung des Motors sicher überwachen. Der Stand der Technik ist, diese zusätzlichen sicherheitsbezogenen Komponenten für die Bewegungsüberwachung dezentral in den Antrieb zu integrieren. Damit ergibt sich eine sicherheitsbezogene Bewegungssteuerung aus der Kombination einer sicherheitsbezogenen Bewegungsüberwachung, einer sicheren Trennung des Motors von der Energiezufuhr und einer nicht sicherheitsbezogenen Bewegungserzeugung. Dieser Stand der Technik ist historisch gewachsen, da bei der Entwicklung sicherheitsbezogener Protokolle wie PROFIsafe der zugehörige Feldbus PROFIBUS mit seinen Zykluszeiten vergleichsweise langsam war. Ein weiterer Aspekt ist die sicherheitsbezogene Rechenleis-

tung von Sicherheits-SPSen, die früher nicht ausreichend hoch war, um komplexere Sicherheitsfunktionen auszuführen. Daher werden sicherheitsbezogene Bewegungsfunktionen heute noch meist dezentral antriebsintegriert als sicherheitsbezogenes Teilsystem, beschränkt auf Einzelachsfunktionen, realisiert [70], [71].

Die IEC 61800-5-2 definiert Sicherheitsfunktionen für Leistungsantriebe mit einstellbarer Drehzahl. Die Sicherheitsfunktionen lassen sich dabei in zwei Kategorien gliedern: Stopp-Funktionen und Überwachungsfunktionen. Nachfolgend werden relevante Stopp-Funktionen, die auch als Fehlerreaktionsfunktionen bezeichnet werden, beschrieben. Bei der Sicherheitsfunktion sicher abgeschaltetes Drehmoment (engl.: Safe Torque Off, STO) nach Stoppkategorie 0 [72] wird verhindert, dass dem Motor drehmomenterzeugende Energie zugeführt wird. Dies kann beispielsweise durch Motorschütze, Netzschütze oder eine Impulssperre realisiert werden [69]. Die Sicherheitsfunktionen sicherer Stopp 1 (engl.: Safe Stop 1, SS1) nach Stoppkategorie 1 und sicherer Stopp 2 (engl.: Safe Stop 2, SS2) nach Stoppkategorie 2 realisieren hingegen ein gesteuertes Stillsetzen.

Neben den Stopp-Funktionen definiert die IEC 61800-5-2 auch Überwachungsfunktionen, die eine Bewegung oder festgelegte physikalische Größen von Antriebssystemen überwachen. Es werden im Folgenden drei Überwachungsfunktionen vorgestellt. Die Sicherheitsfunktion sicher begrenzte Position (engl.: Safely-Limited Position, SLP) verhindert, dass die Applikation eine festgelegte Lagebegrenzung überschreitet. Die Sicherheitsfunktion sicher begrenzte Drehzahl (engl.: Safely-Limited Speed, SLS) dient dazu, die Geschwindigkeit eines Motors auf ein Maximum zu überwachen. Bei der Sicherheitsfunktion sicher begrenzte Beschleunigung (engl.: Safely-Limited Acceleration, SLA) wird verhindert, dass ein festgelegter Grenzwert der Beschleunigung überschritten wird. Voraussetzung für eine antriebsintegrierte sicherheitsbezogene Bewegungsüberwachung ist die Implementierung einer vollwertigen, meist redundanten Sicherheitslogik in der Antriebssteuerung. Dies wird typischerweise in Form einer zusätzlichen Sicherheitskarte [71] realisiert, die auf einer Architektur mit zwei Mikrocontrollern nach Kategorie 3 oder 4 basiert. Aufgrund der Steckverbinder und der zusätzlichen Hardware gehen mit diesem Ansatz erhebliche Kosten einher.

2.2.5 Roboter, kollaborative Roboter und fahrerlose Transportsysteme

Bei der MRK überschneiden sich Arbeitsräume von Menschen und Maschinen zeitlich und räumlich. Diesbezüglich befasst sich die Norm ISO 10218 mit den Sicherheitsanforderungen an Industrieroboter und deren Integration in eine automatisierte Produktion [73]. Die Norm beschreibt allgemeine Anforderungen an die Sicherheit von Robotersystemen und spezielle Anforderungen an die MRK. Dazu gehören unter anderem die Definition von Sicherheitsabständen, die Festlegung von Not-Aus-Funktionen und die Anforderungen an die Bedienelemente von Robotern. Während sich die Sicherheitsfunktionen der IEC 61800-5-2 auf Leistungsantriebe und die Überwachung einzelner Achsen beziehen, werden in der ISO 10218 geeignete Sicherheitsmaßnahmen für die Überwachung vom gesamten Robotersystem definiert. Die Sicherheitsmaßnahme Sicherheitsbewertete reduzierte Geschwindigkeit beschreibt beispielsweise die Überwachung der Geschwindigkeit des Werkzeugarbeitspunkts (engl.: Tool Center Point, TCP) und die notwendige Reaktion in Form eines Sicherheitshalts.

Mit den immer höheren Anforderungen waren die Normen nicht ausreichend, um die Kollaboration von Personen und Robotern sicher umzusetzen. Daher wurde die ISO/Technical Specification (TS) 15066 erarbeitet, die vier Kollaborationsarten als Schutzprinzipien definiert, um die zeitliche und räumliche Überschneidung der Arbeitsräume von Mensch und Maschine sicher zu ermöglichen (siehe Abbildung 9) [74]. Das erste Schutzprinzip ist der sicherheitsbewertete überwachte Stillstand (engl.: Safety-Rated Monitored Stop, SRMS). Bei diesem Schutzprinzip hat der Mensch nur Zugang zum stillstehenden Roboter, der in der Regel hinter Schutzzäunen betrieben wird. Der Roboter darf dabei nicht selbstständig und unerwartet wieder anlaufen. Das zweite Schutzprinzip ist die Handführung (engl.: Hand Guiding, HG), bei dem der Mensch ebenfalls nur Zugang zum eingezäunten stillstehenden Roboter hat. Der Unterschied zu SRMS ist, dass dem Menschen durch eine manuelle Bestätigung einer Zustimmeinrichtung die Handführung des Robotersystems ermöglicht wird. Das dritte Schutzprinzip der ISO/TS 15066 beschreibt die Geschwindigkeits- und Abstandsüberwachung (engl.: Speed and Separation Monitoring, SSM). Hierbei wird permanent der Abstand zwischen Menschen und Roboter sicherheitsbezogen überwacht. Das Robotersystem bewegt sich mit sicher reduzierter Geschwindigkeit. Je näher der Mensch dem Roboter kommt, desto langsamer wird der Roboter, bis letztendlich ein Stillsetzen ausgelöst wird.

Die ersten drei Schutzmethoden gewährleisten die Sicherheit durch den Abstand zwischen Personen und Maschine, eine Kollision ist nicht zulässig. Das vierte Schutzprinzip, die

Leistungs- und Kraftbegrenzung (engl.: Power and Force Limitation, PFL), beschreibt, dass unter bestimmten Umständen der direkte Kontakt zwischen Personen und Robotern möglich ist. Es muss jedoch sichergestellt werden, dass die Kollision keine Gefahr für den Menschen darstellt. Dies wird insbesondere durch die Leistungs- und Kraftbegrenzung des Robotersystems sichergestellt.

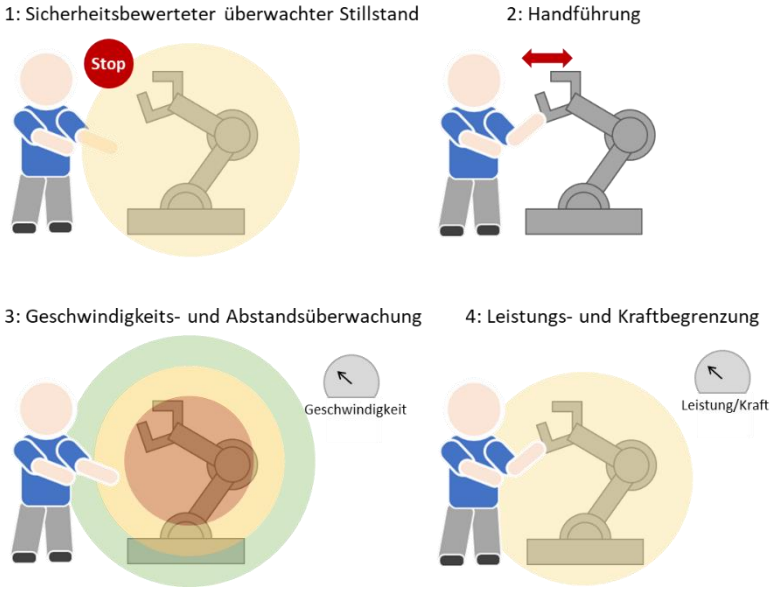


Abbildung 9: Schutzprinzipien für die MRK nach DIN/ISO TS 15066.

Neben den klassischen Industrierobotern gibt es heute speziell für die jeweilige Kollaborationsart konzipierte Cobots. Sie sind in der Regel durch ihre gewichtssparende Bauweise kostengünstiger als herkömmliche Industrieroboter, haben jedoch im Vergleich einige Einschränkungen vorzuweisen:

- Geringere Tragfähigkeit: Cobots sind in der Regel nicht in der Lage, schwere Lasten zu tragen, da sie aufgrund ihres kollaborativen Ansatzes und ihrer gewichtssparenden Bauweise nicht über die gleiche Kraft wie herkömmliche Industrieroboter verfügen. Dies bedeutet, dass sie bevorzugt in Produktionsumgebungen eingesetzt werden, in denen keine schweren Lasten bewegt werden.
- Geringere Geschwindigkeit: Cobots sind in der Regel langsamer als herkömmliche Industrieroboter, da sie auf eine für die Zusammenarbeit mit Menschen sichere Geschwindigkeit eingestellt sind.

Aufgrund der genannten Einschränkungen von Cobots im Vergleich zu herkömmlichen Industrierobotern kann abgeleitet werden, dass Cobots nicht den gleichen Anwendungs-

bereich wie Industrieroboter abdecken können. Für viele Anwendungen sind Industrieroboter aufgrund ihrer höheren Traglast und Geschwindigkeit besser geeignet. Dazu gehören z. B. Handhabungs-, Montage- und Verpackungsaufgaben.

Eine ähnliche Entwicklung ist bei den FTFs und AMRs zu beobachten. Auch hier liegt der Fokus auf der Zusammenarbeit zwischen Personen und den Fahrzeugen. Die ISO 3691-4 beschreibt Sicherheitsanforderungen und die Mittel zu deren Verifizierung für fahrerlose Flurförderzeuge [75]. FTFs und AMRs sind beide in der Norm mit inbegriffen. Die ISO 3691-4 beschreibt verschiedene Betriebsarten der Fahrzeuge und definiert Sicherheitsmaßnahmen, die während den unterschiedlichen Betriebsarten aktiv sein müssen. Es wird beispielsweise gefordert, dass Personenerkennungseinrichtungen so ausgelegt sein müssen, dass die Fahrzeuge anhalten, bevor es zum Kontakt mit Personen kommt. Weiterhin wird ein Geschwindigkeitsgrenzwert definiert, der nicht überschritten werden darf, wenn Personen im Fahrweg erkannt werden.

2.2.6 Sensoren und Aktoren

Für die Umsetzung von Sicherheitsmaßnahmen, die verschiedene Kollaborationsarten zwischen Menschen und Robotern ermöglichen, werden physikalische Größen und Umwelteinflüsse sicherheitsbezogen gemessen. Je nach Art der Kollaboration werden unterschiedliche Sensoren eingesetzt [76], [77]. Beispielsweise werden Drehgeber zur Winkelmessung eines Gelenk-Motors verwendet, um anschließend die Geschwindigkeit oder Beschleunigung zu bestimmen. Mit Stromsensoren kann das Drehmoment eines Motors bestimmt werden und mit Light Detection and Ranging (LiDAR)-Sensoren können Hindernisse oder Menschen sicher erkannt werden, bevor sie in den Arbeitsbereich des Roboters gelangen [78], [79].

Für gekapselte sicherheitsbezogene Subsysteme gibt der Hersteller sicherheitstechnische Kenngrößen wie PL oder SIL, die durchschnittliche Wahrscheinlichkeit eines gefährlichen Ausfalls je Stunde (engl.: Probability of a Dangerous Failure per Hour, PFH_D) und eine Kategorie vor, während der genaue innere Aufbau für den Anwender nicht bekannt sein muss. Diese Kennwerte erfordern die Einhaltung der vom Hersteller angegebenen Betriebsbedingungen, die z. B. externe Diagnosetests umfassen können [37], [80]. Sicherheitsbezogene Sensoren und Aktoren als gekapselte Subsysteme können über sicherheitsbezogene Kommunikationsprotokolle an die Sicherheits-SPS angebunden werden. Dies verringert zwar den Verdrahtungsaufwand, jedoch wird die Komplexität der Sensoren und Aktoren erhöht. Diese Komplexität entsteht dadurch, dass in die Sensoren und Aktoren

zusätzliche zertifizierte Logikeinheiten mit Diagnose integriert werden, um die sicherheitsbezogenen Protokolle verarbeiten zu können. Nachfolgend werden zwei Beispiele zur Veranschaulichung dieser Problematik gezeigt.

Die bereits vorgestellte Stoppfunktion STO wird in der Regel so implementiert, dass verhindert wird, dass die Leistungshalbleiter des Umrichters mit Impulsmustern angesteuert werden. Dadurch kann im Umrichter kein Drehfeld und somit auch kein Drehmoment im Motor erzeugt werden. Diese Sicherheitsfunktion kann auf unterschiedliche Weise bei Antriebsystemen implementiert werden. Beim ersten Ansatz wird die STO-Funktion über zwei digitale 24 V Eingänge realisiert, während Querschlüsse durch eine externe Sicherheits-SPS überwacht werden. Dieser Ansatz verursacht im Antrieb kaum zusätzliche Kosten. Der zweite Ansatz für STO ist die Ansteuerung über ein sicherheitsbezogenes Protokoll wie FSoE oder PROFIsafe. Für Frequenzumrichter bedeutet dies, dass in der Regel eine zusätzliche Sicherheitskarte mit einer zertifizierten Sicherheitslogik integriert wird, um das sicherheitsbezogene Protokoll und die sicherheitsbezogene Impulssperre zu implementieren [71].

Für die sicherheitsbezogene Bewegungsüberwachung gibt es auf dem Markt Motorfeedback-Systeme mit digitalen sicherheitsbezogenen Schnittstellen, die auf dem Gray-Channel-Prinzip basieren [27]. EnDat 3 [81], [82], HIPERFACE DSL [83] und SCS open link [84] sind im Markt verbreitete digitale Schnittstellen für Drehgeber. Die vorgestellten sicherheitsbezogenen digitalen Schnittstellen ähneln den gängigen sicherheitsbezogenen Feldbus-Protokollen wie PROFINET/PROFIsafe und EtherCAT/FSoE, sind aber nicht identisch. Sicherheitsbezogene und nicht sicherheitsbezogene Daten teilen sich eine gemeinsame Schnittstelle. Es werden zwei unabhängige Positionen von den Drehgebern bereitgestellt, die jedoch eine zusätzliche externe Diagnose, wie beispielsweise einen Kreuzvergleich, benötigen, da sie keine zertifizierte Logik enthalten. Die dadurch entstehenden Zusatzkosten für die funktionale Sicherheit sind vergleichsweise gering. Im Gegensatz dazu gibt es auch sicherheitsbezogene Geber mit sicherheitsbezogenen Protokollen wie FSoE und PROFIsafe. Diese sind im Vergleich deutlich teurer, da diese eine zertifizierte Sicherheitslogik integriert haben, um eine sicherheitsbezogene Position zur Verfügung zu stellen. Die Diagnose und der Kreuzvergleich werden hierbei direkt im Gebersystem implementiert.

Für den Einsatz von Sensoren und Aktoren im funktional sicheren Kontext ist es ebenso gängige Praxis, zwei diversitäre und nach ISO 9001 qualitätsgesicherte Standardkomponenten zu verwenden. Beispielsweise können zwei Standard-Drehgeber von zwei unterschiedlichen Herstellern oder mit unterschiedlicher Technologie verwendet werden, um

eine sicherheitsbezogene Position zu erhalten [37], [45], [85], [86]. In [87] wird ein Ansatz vorgestellt, bei dem ein sicherheitsbezogenes Hinderniserkennungssystem (engl.: Obstacle Detection System) verschiedene nicht sicherheitszertifizierte Sensoren verwendet. Diversitäre Sensoren bieten im Vergleich zu sicherheitszertifizierten Sensoren eine größere Freiheit in Bezug auf 2nd Source, ohne das Sicherheitskonzept zu beeinträchtigen. Die externe Diagnose für die diversitären Standardsensoren wird bei diesem Ansatz von einer übergeordneten sicherheitsbezogenen Logik (Sicherheits-SPS oder Antriebssteuerung mit zusätzlicher Sicherheitskarte) ausgeführt.

2.2.7 Software

Normen und Definitionen

Da Software in der Regel komplex ist, gibt es zunehmend mehr Software- als Hardwarefehler [37]. Nach [88] hat gute Software etwa zwei bis drei Fehler pro 1000 Programmzeilen. Das bedeutet, dass in jeglicher Software Fehler vorhanden sind, die unter bestimmten Bedingungen die Funktion beeinträchtigen.

Die IEC 61508-3, die IEC 62061 und die ISO 13849-1 beschreiben auch Anforderungen an die Entwicklung sicherheitsbezogener Software. Grundsätzlich ist für diese Arbeit die Umsetzung der Anforderungen aus der Norm ISO 13849-1 relevant, da diese hauptsächlich im Maschinenbau angewendet wird. Die IEC 62061 als Sektornorm der IEC 61508 beschreibt ähnliche Anforderungen und Vorgehensweisen wie die ISO 13849-1 in Bezug auf sicherheitsbezogene Anwendungssoftware. Die Gremien beider Normen haben inzwischen auch die Gleichwertigkeit der Anforderungen untersucht und in einem gemeinsamen Report ISO/TR 23849:2010 dokumentiert. Das IFA behandelt in [89] und [90] die praxisgerechte Umsetzung der Anforderungen für sicherheitsbezogene Anwendungs- und Embedded-Software nach den geltenden Normen. Für die Entwicklung sicherheitsbezogener Software wird zwischen zwei Sprachtypen unterschieden:

- Programmiersprache mit nicht eingeschränktem Sprachumfang (engl.: Full Variability Language, FVL). Beispiele sind C, C++ und Assembler. Systeme, die FVL verwenden, sind meist Embedded-Systeme.
- Programmiersprache mit eingeschränktem Sprachumfang (engl.: Limited Variability Language, LVL) ermöglicht vordefinierte, anwendungsspezifische Bibliotheksfunktionen zu kombinieren, um die Anforderungen der Sicherheitsfunktionen zu erfüllen. Beispiele für LVL sind LD und Function Block Diagram (FBD). Systeme, die LVL verwenden, sind z. B. SPSen.

Bei den beiden Arten von Software unterscheidet man:

- Sicherheitsbezogene Embedded-Software (engl.: Safety-Related Embedded Software, SRESW): SRESW ist eine Software, die als Teil des Systems durch den Steuerungshersteller bereitgestellt wird. Ein Beispiel ist eine STL. Diese Software hat die Eigenschaft, dass sie üblicherweise in FVL geschrieben wird und nicht vom Anwender geändert werden kann.
- Sicherheitsbezogene Anwendungssoftware (engl.: Safety-Related Application Software, SRASW): SRASW ist eine Software, die für Anwendungen in den Maschinensteuerungen implementiert wird. Ein typisches Beispiel von SRASW ist das Anwendungsprogramm einer Sicherheits-SPS. Üblicherweise wird SRASW in LVL programmiert.

Der Typ der Softwaresprache (LVL oder FVL) und die Softwareart (SRASW oder SRESW) sind entscheidend dafür, welche Normen angewendet und damit welche Anforderungen an die sicherheitsbezogene Softwareentwicklung erfüllt werden müssen. Wenn SRASW in FVL programmiert wird, müssen jedoch nach den geltenden Normen die Anforderungen für SRESW angewendet werden. Nach den geltenden Normen sind die Anforderungen an SRESW/FVL höher als an SRASW/LVL.

PLCopen Safety

Das PLCopen Safety Komitee richtet sich an Hersteller und Anwender von Sicherheits-SPSen und hat zum Ziel, zum allgemeinen Verständnis von Sicherheitsaspekten sowie zur Zertifizierung und Zulassung durch unabhängige, sicherheitsbezogene Organisationen beizutragen [91]. Es werden Richtlinien, Styleguides und grundlegende Spezifikationen von FBs für die Implementierung in sicherheitsbezogenen Anwendungen definiert. Darüber hinaus ermöglichen die PLCopen-Publikationen die Konformität mit den relevanten softwarebezogenen Anforderungen, wie sie in den aufgeführten Normen definiert sind. Die Publikationen der PLCopen sind eine Grundlage für die Spezifikation der Software Sicherheitsanforderungen für sicherheitsbezogene FBs.

Ein wichtiger Aspekt der PLCopen ist die Definition von drei User Levels. Im Basic Level besteht das Sicherheitsprogramm grundsätzlich aus zertifizierten FBs, die grafisch miteinander verbunden werden können. Das Sicherheitsprogramm hat eine klare Struktur und ist leicht zu lesen und zu verstehen. Durch die Einfachheit des Programms lassen sich erste Fehler vermeiden. Gleichzeitig wird die Entwicklungszeit erheblich verkürzt, da die FBs bereits vorzertifiziert sind. Bei Projekten, für die der aktuelle Stand an zertifizierten FBs nicht ausreicht, kann der Anwender das Programm oder einzelne FBs im Extended Level

erstellen. Im Extended Level steht ein erweiterter Befehlssatz zur Verfügung. Die Validierung der FBs und der Programme im Extended Level kann jedoch erheblich komplexer sein, da der gesamte Verifikationsprozess durchlaufen werden muss. Sind FBs im Extended Level programmiert und bereits zertifiziert, lassen sie sich im Basic Level verwenden. Im Extended Level steht die IEC 61131-3 Programmiersprache Structured Text (ST) zur Verfügung. Eine Programmierung in Assembler sowie in Hochsprachen wie C, C++ steht nur im System Level zur Verfügung, welcher jedoch nicht Teil der PLCopen Spezifikation ist.

Sicherheitsbezogene Funktionsbausteine nach PLCopen

Das PLCopen Safety Komitee definiert in [91] eine Bibliothek an sicherheitsbezogenen FBs, die den genannten Richtlinien und Styleguides entsprechen. Diese FBs basieren auf einem einheitlichen Zustandsautomaten, der je nach Sicherheitsfunktion angepasst wird. Zudem werden generelle Regeln zum Start- und Zeitverhalten der FBs definiert. Ebenso werden einheitliche Diagnosecodes sowie Fehlermeldungen für die PLCopen FBs beschrieben.

Da mit den gängigen Sicherheits-SPSen bisher keine hohe Rechenleistung zur Verfügung stand, behandeln die von der PLCopen definierten sicherheitsbezogenen FBs in der Regel nur binäre Signale wie das Überwachen von Sensoren sowie das Steuern und Überwachen von Zuständen von Aktoren. Von der PLCopen definierte sicherheitsbezogene FBs sind z. B. die Überwachung eines Not-Halts oder die Überwachung einer Zweihand-Bedienung. Komplexe Algorithmen wie die Berechnung von Bewegungen von Achsen oder Robotern mit Gleitkomma-Arithmetik sind bisher nicht vorgesehen. Die Sicherheitsfunktionen in Bezug auf elektrische Antriebssysteme, die von der IEC 61800-5-2 definiert sind, werden bisher ebenfalls nicht von der PLCopen berücksichtigt.

2.3 Virtualisierung

2.3.1 Grundlegendes

Entwickler von IoT- und Edge-Geräten werden mit steigenden Anforderungen konfrontiert, da von vernetzten Geräten zunehmend erwartet wird, dass sie eine Vielzahl von Hardwareressourcen, Betriebssystemen, Softwaretools und Anwendungen unterstützen. Die zunehmende Digitalisierung und Automatisierung in der Industrie erfordert eine hohe Flexibilität und Skalierbarkeit der IT-Infrastruktur. Die Integration von verschiedenen Anwendungen auf einer gemeinsamen Hardwareplattform kann jedoch eine Herausforderung

darstellen. Insbesondere dann, wenn Anwendungen mit unterschiedlicher Kritikalität gefordert werden. Darüber hinaus müssen Sicherheitsrisiken minimiert und Kosteneinsparungen erzielt werden. In der IT wird die Virtualisierung üblicherweise in großen Datenzentren und Rechenclustern eingesetzt. In der Vergangenheit war der Einsatz von Virtualisierung in der Automatisierung nicht weit verbreitet, da die eingesetzten Systeme auf eine Anwendung spezialisiert und ausgerichtet waren [2]. Mit den steigenden Anforderungen und der zunehmenden Nachfrage nach Flexibilität und Modularität kann die Virtualisierung auch in der Automatisierung zunehmend Einzug halten. Im Folgenden werden verschiedene Virtualisierungstechnologien vorgestellt und verglichen.

2.3.2 Hypervisor

Bei einem Hypervisor, auch Virtual-Machine-Monitor (VMM) genannt, handelt es sich um Software, die es ermöglicht, mehrere Betriebssysteme parallel auf derselben Hardware auszuführen, indem die Rechnerressourcen virtualisiert werden. Die Betriebssysteme werden innerhalb von Virtuellen Maschinen (VM), auch Guests genannt, ausgeführt, die eine ganze Computer-Hardware-Umgebung in Software abbilden. Der Hypervisor, auch Host genannt, hat die Aufgabe, diese parallel laufenden VMs zu verwalten. Er trennt die VMs logisch voneinander, indem er ihnen die zugrunde liegenden Ressourcen wie Prozessorkerne, Random-Access Memory (RAM), nichtflüchtigem Speicher, wie Solid-State Drives (SSD) und Secure Digital (SD)-Karten, und Peripherie zuweist. Alle Hypervisoren benötigen für ihre Aufgaben bestimmte Komponenten wie Speicherverwaltung, Scheduler, Gerätetreiber, Security Manager und Netzwerkstack.

Vor der Einführung des ersten Hypervisors konnte auf einer Hardware lediglich nur ein Betriebssystem gleichzeitig ausgeführt werden. Der Nachteil dabei war die nicht optimale Auslastung, da über das Betriebssystem nicht immer die gesamte Rechenleistung und die Ressourcen des Computers genutzt wurden. Im Gegensatz dazu, können mithilfe der Virtualisierung mehrere Anwendungen, jede isoliert in einer eigenen VM mit eigenem Betriebssystem, auf einem einzigen physischen Computer ausgeführt werden. Ein Nachteil der Virtualisierung von Ressourcen ist der entstehende Overhead, der z. B. durch die Emulation und Zuweisung der Ressourcen entsteht. In [92] werden zwei Typen von Hypervisoren unterschieden. Diese Unterscheidung ist auch heute noch gültig und gliedert Hypervisoren in Typ 1 und Typ 2 (siehe Abbildung 10).

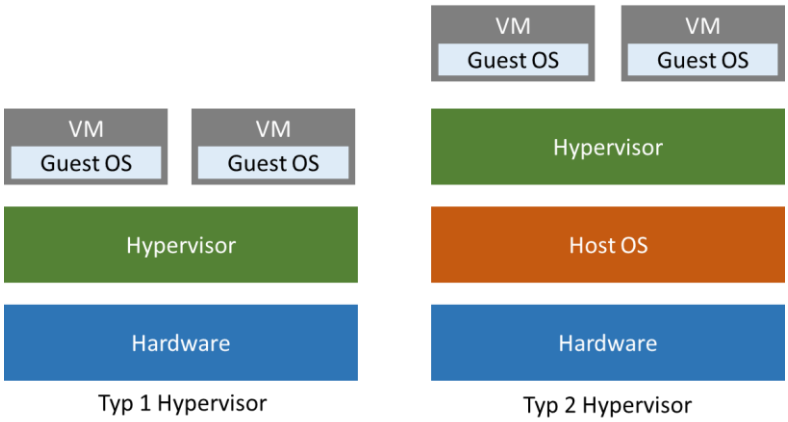


Abbildung 10: Vergleich von Hypervisoren vom Typ 1 und Typ 2

Der Hypervisor vom Typ 1, auch als Bare-Metal Hypervisor bezeichnet, wird direkt auf der physischen Hardware des Computersystems ausgeführt und interagiert direkt mit dieser. Eine Betriebssysteminstallation wird nicht benötigt. Dies setzt jedoch voraus, dass die Hardware des Host-Computers durch entsprechende Treiber unterstützt wird.

Für die Verwaltung von verschiedenen parallel ausgeführten VMs unterstützen einige Typ 1 Hypervisoren einen Sharing Mode. Service VMs ermöglichen die gleichzeitige Nutzung von Ressourcen und sind für deren Emulation zuständig. Diese Konfiguration bringt jedoch aufgrund der Service VM einen Overhead mit sich, da auch diese Ressourcen (CPU, Speicher etc.) benötigt. Verzichtet man auf die Verwendung einer Service VM, können Hardwareressourcen nicht mehr von mehreren VMs gemeinsam genutzt werden und müssen eindeutig einer VM zugewiesen werden. Dieser sogenannte Partition Mode bietet Unabhängigkeit und Isolation für die User VMs. Diese Partitionierung ist insbesondere für echtzeit- und sicherheitskritische Anwendungen von hoher Relevanz. Alle VMs führen hierbei native Gerätetreiber aus und greifen direkt auf ihre konfigurierten Hardwareressourcen zu.

Der Hypervisor vom Typ 2, auch Hosted Hypervisor genannt, läuft im Gegensatz zum Bare-Metal Hypervisor nicht direkt auf der zugrunde liegenden Hardware, sondern als Anwendung innerhalb eines Betriebssystems. Dieses Betriebssystem wird als Host-Betriebssystem oder Host-OS bezeichnet. Um auf die Ressourcen des Rechners zuzugreifen, nutzt der Hypervisor die Gerätetreiber des Host-Betriebssystems. Generell sind Typ 2 Hypervisoren auf allen Rechnern lauffähig, auf denen vom Hypervisor unterstützte Host-Betriebssysteme laufen. Hypervisoren vom Typ 2 ermöglichen einen schnellen und einfachen Zugriff auf ein alternatives Guest-Betriebssystem neben dem primären Host-Betriebssystem. Da Typ 2 Hypervisoren auf Rechen-, Speicher- und Netzwerkressourcen über das Host-Betriebssystem zugreifen, das primären Zugriff auf die physische Hardware

hat, führt dies zu zusätzlichen Latenzen, die die Leistung beeinträchtigen. In dem Aspekt sind Typ 1 Hypervisoren deutlich effizienter, da sie der Anwendung direkten Zugriff auf die Hardware erlauben. Bei Typ 2 Hypervisoren kann es ebenso zu potenziellen Sicherheitsrisiken führen, wenn ein Angreifer das Host-Betriebssystem kompromittiert, da er dann auch jedes Guest-Betriebssystem manipulieren kann.

Zugriff auf Hardware

Ein wichtiger Teil bei der Nutzung eines Hypervisors ist die Virtualisierung von Ressourcen. Dabei gibt es drei gängige Methoden der Gerätevirtualisierung: Emulation, Paravirtualisierung und Device Passthrough.

Bei der Emulation emuliert der Hypervisor Geräte bzw. Hardwarekomponenten, die er der VM unabhängig von der tatsächlichen physischen Hardware zur Verfügung stellt. Ein Vorteil der Emulation ist, dass ein Gerät oder eine bestimmte Hardwareressource von mehreren VMs aus gleichzeitig verwendet werden kann. Ebenso ist die VM-Portabilität ein Vorteil, da jede VM auf jeder Hardware ausgeführt werden kann. Die VM sieht nur die emulierte Hardware, nicht aber die tatsächliche physische Hardware. Nachteile der Emulation sind der hohe Entwicklungsaufwand und der hohe Overhead.

Bei der Paravirtualisierung wird der Guest-VM eine modifizierte Version der physischen Hardwareschnittstelle zur Verfügung gestellt. Diese Art der Gerätevirtualisierung ist wesentlich leistungsfähiger als die Emulation. Der Nachteil ist die Portabilität, da das Gerät, der Hypervisor und das Guest-OS wissen müssen, welche Hardware paravirtualisiert werden soll. In der Praxis bedeutet dies, dass nur bestimmte Hardware in bestimmten Konfigurationen unterstützt wird.

Beim Device Passthrough wird einer VM ein physisches Gerät so zugewiesen, dass die VM ohne Hypervisor-Beteiligung direkt auf dieses zugreifen kann. Diese Art der Gerätevirtualisierung bietet native Performance [93]. Ein Nachteil ist jedoch die exklusive Nutzung von Ressourcen von einer einzelnen VM. Die Unterschiede zwischen Device Passthrough und Emulation werden in der Abbildung 11 dargestellt.

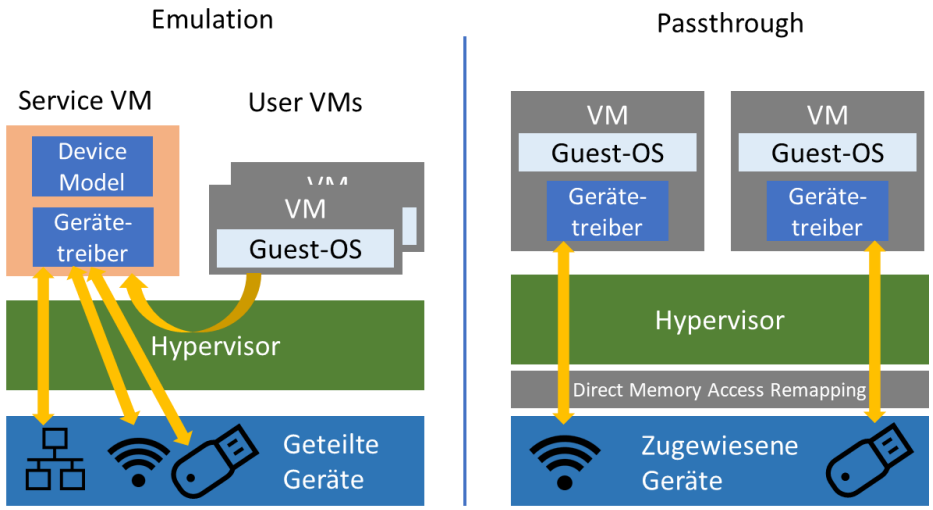


Abbildung 11: Vergleich von Emulation und Passthrough.

Der Zugriffspfad bei der Emulation ist wesentlich länger im Vergleich zu Passthrough. Dies führt bei der Emulation demzufolge zu einer schlechteren Performance. Passthrough bietet eine nahezu native Performance, da keine Context-Switches zwischen der VM und dem Hypervisor verursacht werden [94].

Inter-VM Kommunikation

Moderne Hypervisoren bieten verschiedene Möglichkeiten, wie VMs miteinander kommunizieren können. Gängige Beispiel sind:

- Inter-VM virtueller Universal Asynchronous Receiver Transmitter (vUART)
- Inter-VM Netzwerk-Kommunikation
- Inter-VM Shared Memory

Typ 1 Hypervisoren für die funktionale Sicherheit

Der Bedarf an Systemen mit Anwendungen unterschiedlicher Kritikalität steigt. Es gibt erste sicherheitsbezogene Hypervisoren, die Anwendungen unterschiedlicher Kritikalität ausführen können und für die funktionale Sicherheit zertifiziert sind bzw. werden. Ein Beispiel ist der ACRN Hypervisor als Open-Source-Projekt [94] für Intels x86-Prozessoren, der in der Version 1.4 vom TÜV Süd Rail zertifiziert wurde. Auf der Embedded World Messe 2023 hat Real-Time Systems GmbH ihren RTS Safe Hypervisor vorgestellt. Der RTS Safe Hypervisor unterstützt ebenfalls ausschließlich Prozessoren von Intel und soll bis SIL3 und PL e zertifiziert werden [95].

2.3.3 Mikrokernel

Mikrokernels sind Betriebssystemkernels, die nur grundlegende Funktionen enthalten und auf einer minimalistischen Architektur basieren. Im Gegensatz zu monolithischen Betriebssystemkernels, die alle Systemdienste und -funktionen in einem einzigen Kernel zusammenfassen, teilen Mikrokernels diese Dienste in separate Prozesse auf, die über eine definierte Schnittstelle kommunizieren. Der Ansatz erlaubt es, das Betriebssystem flexibler und modularer zu gestalten, da Dienste nach Bedarf hinzugefügt und entfernt werden können.

Einige Mikrokernels bieten grundlegende Funktionen, die dazu verwendet werden können, die Systemressourcen (Speicher, E/A-Geräte und CPU-Zeit) in separate Partitionen aufzuteilen [96]. Diese Art von Mikrokernel wird auch Separationskernel genannt und integriert eine Virtualisierungsschicht in den Kernel, die es ermöglicht, mehrere Partitionen (vergleichbar mit VMs bei einem Hypervisor) auf einer Hardwareplattform auszuführen und Hardwareressourcen räumlich und zeitlich zwischen den Partitionen zu verwalten [97]. Da die Partitionen auf getrennten Ressourcen ausgeführt werden, sind sie vollständig isoliert.

Mikrokernels und Hypervisoren unterscheiden sich jedoch in einigen Eigenschaften [98], [99], [100]. Mikrokernels kennen alle Threads und Prozesse eines Betriebssystems. Der Hypervisor verwaltet im Gegensatz dazu nur die Guest-Betriebssysteme. Ein Mikrokernel verwaltet alle Gerätetreiber als separate Mikrokernelprozesse. In [98] wird gezeigt, dass die Auswirkungen der Mikrokernel-Virtualisierung auf die Leistung des nativen Betriebssystems größer sind als bei einem Hypervisor.

Separationskernels, die gleichzeitig die Ausführung von Anwendungen unterschiedlicher Kritikalität erlauben, sind PikeOS der Firma SYSGO und der QNX Hypervisor for Safety von BlackBerry QNX. Der QNX Hypervisor nennt sich zwar Hypervisor, basiert jedoch auf einem Mikrokernel. Beide Mikrokernels, QNX und PikeOS, sind für die funktionale Sicherheit zertifiziert, unterstützen neben Intel x86-Prozessoren auch ARM Prozessoren und bieten auch Mechanismen wie Shared Memory für einen schnellen Datenaustausch zwischen den VMs [101], [102].

2.3.4 Prozessor-Unterstützung für die Virtualisierung

Je nach Kontext werden die Prozessoren bei der Ausführung von Code in unterschiedlichen Privileg-Ebenen betrieben. Das Betriebssystem, das auf einem physischen Rechner betrieben wird, wird in der Regel im Kernel-Mode bzw. Supervisor-Mode ausgeführt. Das bedeutet, dass das Betriebssystem uneingeschränkten Zugriff auf die CPU-Register, den

vollständigen Befehlssatz der CPU, den gesamten Adressraum aller angebundener Speicher und die Peripheriesteuerregister usw. hat. Im Gegensatz dazu haben Benutzeranwendungen, die in der Regel im User-Mode ausgeführt werden, nur eingeschränkten Zugriff auf die Systemressourcen.

Moderne Prozessoren bieten Hardwareunterstützung für die Umsetzung von Virtualisierungstechnologien. Die x86-Architektur hat vier Ausführungsmodi, aufgeteilt in vier Ringe, wobei der Kernel-Mode des Betriebssystems in Ring 0 und der User-Mode in Ring 3 ausgeführt wird (siehe Abbildung 12). Bei konventionellen Betriebssystemen wie Linux und Windows sind die Ringe 1 und 2 unbesetzt. Wird das Betriebssystem innerhalb einer VM ausgeführt, kommt es zur folgenden Problematik: Der Kernel darf keinen direkten Zugriff auf die Hardwareressourcen erhalten, da dies die Aufgabe des Hypervisors ist. Gleichzeitig darf der Kernel jedoch nicht erkennen, dass er nicht direkt auf der physischen Hardware aufsetzt und von einem Hypervisor (oder einem Mikrokernel) ausgeführt wird. Intel adressiert diese Problematik durch die Einführung der Intel Virtualization Technology (Intel VT) [103]. Für die Virtualisierung von Prozessorhardware bietet Intel VT die Intel Virtual-Machine Extensions (Intel VMX). Intel VMX ermöglicht die Prozessorunterstützung für die Virtualisierung in Form eines Prozessorbetriebs, der als VMX-Betrieb bezeichnet wird. Dabei gibt es zwei Arten von VMX-Betriebsarten: VMX-Root-Betrieb und VMX-Non-Root-Betrieb. Übergänge zwischen den beiden Betriebsarten werden VMX-Übergänge (VM-Exit, VM-Entry) genannt.

Beide Betriebsarten (Root- und Non-Root-Betrieb) haben vier Ringe. Der Hypervisor wird im Ring 0 vom VMX-Root-Mode ausgeführt und lässt die Ringe 1-3 ungenutzt. Eine VM wird im VMX-Non-Root-Betrieb ausgeführt und hat ihre eigenen vollständigen Ringe 0-3. Der Guest-Kernel wird in Ring 0 ausgeführt, während die Benutzeranwendungen in Ring 3 ausgeführt werden. Ein Hypervisoraufruf (engl.: Hypervisor Call, Hypercall) ist für einen Hypervisor das, was ein Systemaufruf (engl.: System Call, Syscall) für einen Kernel ist. Ein Hypercall ist ein Aufruf von einer VM zum Hypervisor, so wie ein Syscall ein Aufruf von einer Anwendung zum Kernel ist. Hypercalls werden verwendet, um vom Betriebssystem oder der Benutzeranwendung Dienste auszuführen, auf die nur der Hypervisor Zugriff hat. Dabei gibt die VM die Kontrolle über den CPU-Kern an den Hypervisor ab und wird so lange unterbrochen, bis die Anfrage vollständig bearbeitet ist.

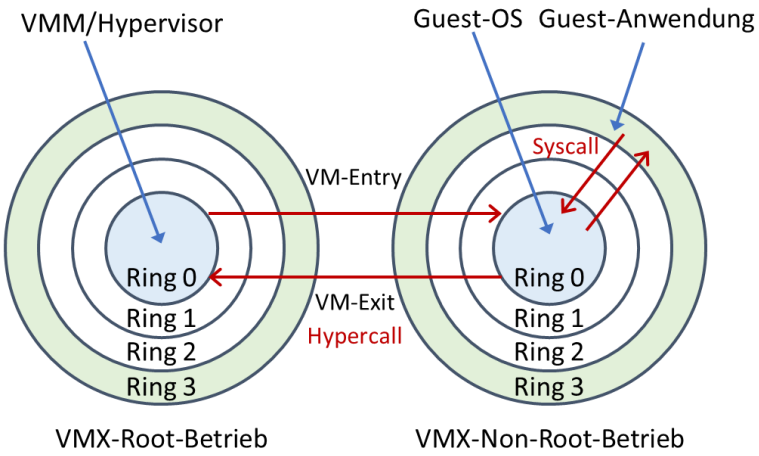


Abbildung 12: Schema der Ringe beim x86-Prozessor mit Virtualisierung.

Bei einem VM-Exit muss der Prozessor einen Snapshot der VM abspeichern. Bei einem Snapshot wird der Zustand eines Systems erfasst und gespeichert, um diesen Zustand zu einem späteren Zeitpunkt wiederherstellen zu können. VM-Exits und damit einhergehend auch VM-Entries sind die Hauptursache für Performanceverluste in virtualisierten Systemen.

Die ARM Virtualization Erweiterung basiert auf einem ähnlichen Ansatz, unterscheidet sich jedoch in einigen Eigenschaften von Intel VT [104]. Sie konzentriert sich auch auf eine neue CPU-Privilegstufe (engl.: Exception Level, EL). Der EL2, in dem der Hypervisor ausgeführt wird, wird zu den bestehenden Stufen für den User-Mode (EL0) und für den Kernel-Mode (EL1) hinzugefügt.

2.3.5 Container

Im Gegensatz zu einer VM, die eine vollständige Betriebssystem-Instanz mit eigenem Kernel und eigenen Ressourcen wie CPU, Arbeitsspeicher und nichtflüchtigem Speicher emulieren, teilen sich Container den Host-Kernel und nutzen die zugrunde liegende Infrastruktur des Betriebssystems. Es wird sichergestellt, dass ein Container nur eine begrenzte Menge an Ressourcen wie CPU, RAM usw. enthält und diese nicht mit anderen Containern oder dem Host-Betriebssystem teilt. Dadurch benötigen Container in der Regel weniger Ressourcen als VMs. Die Containertechnologie befasst sich mit der Isolation und Kapselung von Softwareanwendungen und den notwendigen Bibliotheken in verschiedenen Containern. Das Ziel der Technologie besteht darin, leichtgewichtige Container zu erstellen, die unabhängig von ihrer Umgebung lauffähig sind und zwischen verschiedenen Systemen ausgetauscht werden können.

2.3.6 Vergleich der Virtualisierungstechnologien

Aus sicherheitstechnischer Sicht bieten Architekturen, die auf einem Mikrokernel oder Hypervisor basieren, in der Regel eine bessere Isolation als Architekturen, die auf der Containertechnologie basieren. Die Containerisierung bietet lediglich eine begrenzte Isolation. Beispielsweise kann im ungünstigsten Fall eine schädliche Anwendung bis zum Host-Betriebssystem vordringen und so die Integrität und Sicherheit des Systems gefährden. Durch den Einsatz eines Mikrokernels oder Hypervisors kann dies sicher verhindert werden. Diese Technologien erlauben es, die Anwendungen und Ressourcen auf einer niedrigeren Ebene zu isolieren.

Hypervisoren weisen aufgrund ihrer Vielschichtigkeit eine höhere Komplexität auf und benötigen daher mehr Ressourcen. Mikrokernel hingegen basieren auf einem modularen Ansatz, indem nur grundlegende Dienste vom Kernel bereitgestellt werden. Alle weiteren Dienste werden außerhalb des Mikrokernels ausgeführt, was jedoch zu Performanceeinbußen führen kann. Container bieten hingegen eine einfache Instanziierung, dynamisches Deployment und eine hohe Skalierbarkeit. Sie können auf verschiedenen Host-Betriebssystemen ausgeführt werden, solange diese die Container-Technologie unterstützen.

3 Sicherheitskonzept mit zentraler Sicherheitssteuerung

Dieses Kapitel stellt ein Konzept vor, bei dem die Ausführung bewegungsüberwachender Sicherheitsfunktionen und die damit verbundene Diagnose sicherheitsbezogener Komponenten in einer übergeordneten Sicherheits-SPS erfolgt. Der im vorherigen Kapitel beschriebene Stand der Technik ist, dass die Sicherheitsfunktionen, die eine Bewegung überwachen, dezentral im Antriebssystem implementiert werden. Damit kann lediglich die Bewegung einer einzelnen Achse, nicht aber eine ganze Maschine sicherheitsbezogen überwacht werden. Mithilfe von zentralen Sicherheitsfunktionen soll eine Kollaboration von Menschen und Maschinen einfacher realisiert werden können, da die gesamte Maschine sicherheitstechnisch überwacht werden kann. Zusätzlich müssen die Reaktionszeiten beachtet werden, die für eine Kollaboration gefordert werden. Darüber hinaus soll der vorgestellte Ansatz den Einsatz von nicht sicherheitszertifizierten Sensoren und Aktoren für funktional sichere Anwendungen vereinfachen, da die Diagnose von der übergeordneten Sicherheits-SPS durchgeführt wird. Ebenso wird beschrieben, wie der Zertifizierungs- und Inbetriebnahmeprozess durch die Verwendung von grafischer Programmierung in LVL IEC 61131-3 Programmiersprachen in der Sicherheits-SPS vereinfacht wird.

3.1 Anforderungen an zentrale Sicherheitsfunktionen

Um Sicherheitsmaßnahmen, die komplexe Bewegungen ganzer Maschinen überwachen, zentral in einer übergeordneten Sicherheits-SPS auszuführen, werden unterschiedliche Anforderungen an das System gestellt. Einige Sicherheitsmaßnahmen können mit geringem Aufwand integriert werden, insbesondere wenn keine hohe Rechenleistung oder keine kurzen Reaktionszeiten erforderlich sind. Dagegen erfordern Sicherheitsmaßnahmen, die eine Kollaboration von Menschen und Robotern ermöglichen sollen, deutlich höhere Anforderungen, da komplexe mathematische Algorithmen in möglichst kurzen Zykluszeiten berechnet werden müssen. Nachfolgend wird die Komplexität und der Aufwand von Einzelachs- und Mehrachs-Sicherheitsfunktionen miteinander verglichen. Ebenso wird anhand eines praxisnahen Beispiels beschrieben, welchen Einfluss eine hohe Rechenleistung, kurze Zykluszeiten und eine schnelle Kommunikation auf die funktionale Sicherheit haben.

3.1.1 Vergleich von Einzelachs- und Mehrachs-Sicherheitsfunktionen

Um die Kollaboration von Menschen mit autonomen Maschinen zu ermöglichen, soll das Gesamtverhalten der Maschine sicherheitstechnisch überwacht werden. Für die Realisierung der Kollaborationsart SSM wird jede gefährliche Bewegung eines Roboters überwacht. Die Sicherheitsfunktionen SLS überwacht als Einzelachs-Sicherheitsfunktion nach der IEC 61800-5-2 die Drehzahl bzw. die Geschwindigkeit eines einzelnen Motors. Dazu wird mithilfe eines Drehgebers der Winkel des Motors $\varphi(t)$ zyklisch abgetastet und als zeit- und wertdiskrete Folge φ_k der Sicherheits-SPS bereitgestellt (siehe Abbildung 13).

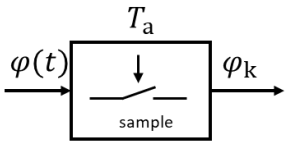


Abbildung 13: Abtastung des Motorwinkels.

Um die Drehzahl (Winkelgeschwindigkeit) zu bestimmen, wird der abgetastete Winkel numerisch differenziert:

$$\omega_\mathrm{k} = \frac{\varphi_\mathrm{k} - \varphi_{\mathrm{k}-1}}{t_\mathrm{k} - t_{\mathrm{k}-1}} \tag{1}$$

Anschließend vergleicht die Sicherheitsfunktion SLS die berechnete Geschwindigkeit mit einem zuvor festgelegten Geschwindigkeitsgrenzwert. Bei Überschreitung des Grenzwerts wird eine geeignete Reaktionsfunktion ausgeführt, um die Achse in den sicheren Zustand zu überführen.

Bezieht sich die Geschwindigkeitsüberwachung auf einen Roboter, definiert die ISO 10218 geeignete Sicherheitsmaßnahmen, wie z. B. die sicherheitsbewertete reduzierte Geschwindigkeit des TCPs, welche im Folgenden ebenfalls als SLS bezeichnet wird. Bei einem Mehrachs-System ist die Komplexität für eine Überwachung der Bewegung deutlich erhöht. Mithilfe der direkten kinematischen Transformation (Vorwärtskinematik) kann z. B. aus den Gelenkwinkeln der Armelemente eines Roboters die Position und Orientierung aller relevanten beweglichen Teile ermittelt werden. Das sind z. B. der TCP und die Ellenbogen, die als Point of Interests (POI) bezeichnet werden. Bei seriellen Robotern, wie Knickarm-Robotern oder Selective Compliance Assembly Robot Arm (SCARA)-Robotern, kann die direkte Kinematik durch eine Matrizenmultiplikation der Denavit-Hartenberg-Matrizen berechnet werden [105]. Bei Robotern mit parallelkinematischer Struktur, die nicht durch die Denavit-Hartenberg-Transformation beschrieben werden können, ist eine analytische Lösung der direkten Kinematik nicht möglich. In [106] und [107] wird die direkte kinematische Transformation eines Delta-Parallel-Roboters, auch als Tripod-

Roboter bezeichnet, durch eine geometrische Herangehensweise beschrieben. Auch bei FTFs ist eine direkte kinematische Transformation notwendig, um die Geschwindigkeit und Richtung des Fahrzeugs zu bestimmen. Hierbei können die Fahrzeuge mit unterschiedlichen kinematischen Antriebsarten, wie einem Differentialantrieb, Drehschemelantrieb oder Doppeldrehschemelantrieb realisiert werden. Mit der Geschwindigkeit der einzelnen Räder, der verwendeten Antriebsart und ggf. mithilfe des Lenkwinkels der Antriebe (entfällt beim Differentialantrieb) kann durch die direkte kinematische Transformation der Geschwindigkeitsvektor des Fahrzeugs berechnet werden. In der Abbildung 14 ist das Blockschaltbild der Sicherheitsfunktion SLS für ein Mehrachs-System zu sehen. Insbesondere die kinematische Struktur von Robotern und Fahrzeugen trägt zur Komplexität der direkten kinematischen Transformation bei.

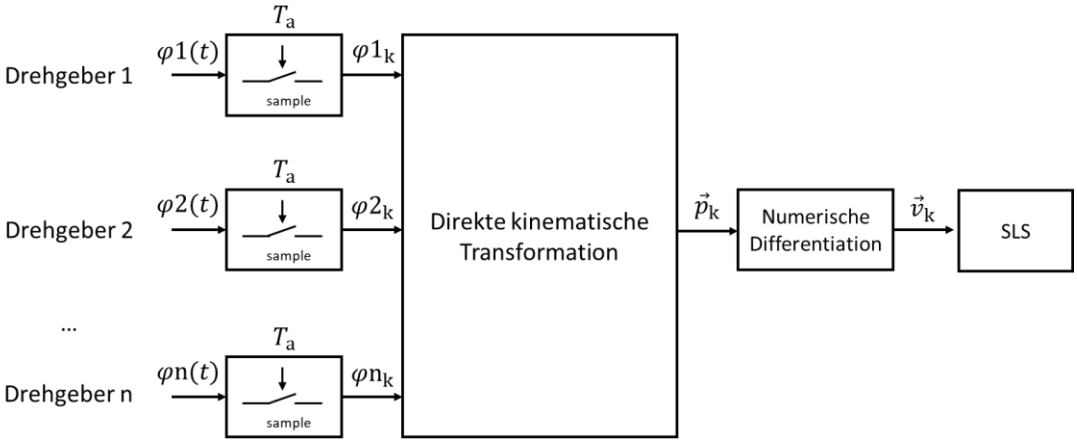


Abbildung 14: Blockschaltbild für die Umsetzung einer Geschwindigkeitsüberwachung für ein Mehrachs-System.

Bei der Überwachung von Maschinen mit mehreren Achsen ist zu beachten, dass ein dreidimensionaler Vektor gegen einen Grenzwert verglichen wird. Zusätzlich ist zu berücksichtigen, dass bei der sicherheitsbezogenen Berechnung von kinematischen Transformationen Gleitkomma-Variablen und trigonometrische Funktionen wie Sinus und Cosinus verwendet werden, weshalb eine zertifizierte FPU vorteilhaft ist.

3.1.2 Bestimmung von Reaktionszeiten

Im Folgenden soll die Zeit bestimmt werden, die ein sicherheitsbezogenes System im Fehlerfall bzw. bei Überschreitung vorgegebener Grenzen benötigt, um eine Reaktionsfunktion im Antrieb anzusteuern. Betrachtet wird eine Geschwindigkeitsüberwachung, die von der bereits vorgestellten Sicherheitsfunktion SLS durchgeführt wird. Zunächst wird die Reaktionszeit für ein Einzelachs-System mit integrierter Sicherheitslogik in der Antriebssteuerung bestimmt, anschließend wird das Problem auf ein Mehrachs-System mit zentraler Sicherheits-SPS übertragen.

Einzelachs-System mit Geschwindigkeitsüberwachung

Abbildung 15 beschreibt den Verlauf eines abgetasteten Motorwinkels. Die Geschwindigkeit ω_k zum Zeitpunkt t_k wird durch numerische Differentiation basierend auf den Messungen zu den Zeitpunkten t_k und $t_{\mathrm{k}-1}$ approximiert. Da die Abtastung nur zu diskreten Zeitpunkten erfolgt, gibt es eine Unschärfe in der Bestimmung der tatsächlichen Geschwindigkeit zum Zeitpunkt t_k. Die Steigung zwischen den Zeitpunkten t_k und $t_{\mathrm{k}-1}$ repräsentiert die Geschwindigkeit zwischen diesen beiden Zeitpunkten. Dadurch ergibt sich näherungsweise eine Totzeit für die Geschwindigkeit von:

$$T_\mathrm{t} = \frac{T_\mathrm{a}}{2} \tag{2}$$

Die tatsächliche Geschwindigkeit des Motors kann bereits einen höheren Wert als die berechnete Geschwindigkeit angenommen haben. Dies ist daran zu erkennen, dass in Abbildung 15 die Steigung der roten Tangente im Punkt t_k größer ist als die Steigung des grünen Steigungsdreiecks für t_k und $t_{\mathrm{k}-1}$. Je kleiner die Abtastzeit T_a ist, umso genauer ist die Bestimmung der Geschwindigkeit und umso kleiner ist die Totzeit.

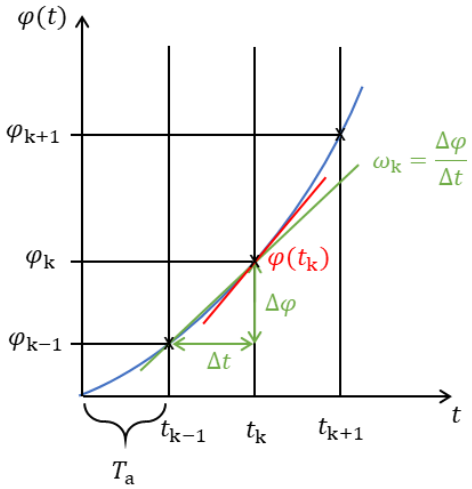


Abbildung 15: Zyklische Abtastung eines Winkels und Geschwindigkeitsbestimmung.

Zur Bestimmung der tatsächlichen Motordrehzahl werden neben der Totzeit T_t durch die numerische Differentiation weitere Verzögerungszeiten berücksichtigt. In [82] wird das Zeitverhalten für die Datenübertragung von Positionssignalen von einem Drehgeber zur Antriebssteuerung beschrieben. Die Zeit für die Datenübertragung vom Drehgeber zur sicheren Logik (T_Geber), die Verarbeitung der Sicherheitsfunktion in der sicheren Logik (T_Logik) und die Ausführung der Reaktionsfunktion (T_STO), wie beispielsweise das Ansteuern der Abschaltpfade für STO, müssen ebenso für die Reaktionszeit berücksichtigt werden. Damit ergibt die minimale Reaktionszeit:

$$T_\mathrm{Reaktion,min} = \frac{T_\mathrm{a}}{2} + T_\mathrm{Geber} + T_\mathrm{Logik} + T_\mathrm{STO} \tag{3}$$

Nun kann der Fall eintreten, dass der Motor beschleunigt und die aktuelle Geschwindigkeit $\omega(t_\mathrm{k})$ den festgelegten Geschwindigkeitsgrenzwert v_SLS bereits überschritten hat, jedoch die berechnete Geschwindigkeit v_k aufgrund von Verzögerung und Abtastung weiterhin unter dem Geschwindigkeitsgrenzwert v_SLS liegt:

$$\omega_\mathrm{k} < \omega_\mathrm{SLS} \quad \text{obwohl} \quad \omega(t_\mathrm{k}) > \omega_\mathrm{SLS} \tag{4}$$

In diesem Fall kann eine zu hohe Geschwindigkeit von der Sicherheitslogik erst zum Zeitpunkt $t_{\mathrm{k}+1}$ erkannt werden. Daraus ergibt sich für ein System mit integrierter Sicherheitslogik in der Antriebssteuerung eine maximale Reaktionszeit von:

$$T_\mathrm{Reaktion,max} = \frac{3T_\mathrm{a}}{2} + T_\mathrm{Geber} + T_\mathrm{Logik} + T_\mathrm{STO} \tag{5}$$

Zusätzlich ist zu beachten, dass der Motor während der Verarbeitung weiter beschleunigen kann, was zu einer höheren Geschwindigkeit und damit zu einem längeren Bremsweg und einer längeren Anhaltezeit führt.

Mehrachs-System mit Geschwindigkeitsüberwachung

Wird die Sicherheitsfunktion nicht dezentral in der Antriebssteuerung, sondern zentral von einer übergeordneten Sicherheits-SPS ausgeführt, sind bei der Fehlerreaktionszeit weitere Faktoren zu berücksichtigen. Der sicherheitsbezogene Winkel wird dabei zunächst über ein sicherheitsbezogenes Protokoll zur Sicherheits-SPS übertragen. Die Synchronisation der Antriebs-PWM mit dem Feldbuszyklus (Distributed-Clock bei EtherCAT) ermöglicht eine vollständig synchrone Abtastung und Aktualisierung der an die SPS angeschlossenen Peripherie [26]. Bei der Geschwindigkeitsüberwachung ist die Zykluszeit der Sicherheitsanwendung (T_SSPS) zu berücksichtigen. Diese entspricht der Abtastzeit für das sicherheitsbezogene Positionssignal:

$$T_{\mathrm{SSPS}} = T_{\mathrm{a}} \tag{6}$$

Bei Verwendung einer Compound-SPS ist es erforderlich, dass die Sicherheits-SPS synchron mit der Standard-SPS und dem Feldbus arbeitet, damit beim Austausch der sicherheitsbezogenen Daten kein Zyklus verloren geht. Wird das Signal zur Ansteuerung der Reaktionsfunktion im Antrieb (z. B. STO) über ein sicherheitsbezogenes Protokoll übertragen, ist auch dessen Zykluszeit zu berücksichtigen. FSoE nutzt ein bidirektionales Producer-Consumer-Modell und benötigt in der Standardkonfiguration mindestens die doppelte Zykluszeit des verwendeten Feldbusses [108], [109]. Es wird eine Nachricht vom Master an die Slaves gesendet, während die Antworten der Slaves an den Master in der Regel im nächsten Zyklus übertragen werden. Wird eine Fehlerreaktion ausgelöst und das entsprechende Signal per FSoE zur Antriebssteuerung gesendet, kann es sein, dass es erst mit dem nächsten Buszyklus gesendet wird:

$$T_{\mathrm{SPDU}} = T_{\mathrm{SSPS}} = T_{\mathrm{a}} \tag{7}$$

Die Abbildung 16 stellt (nicht maßstabgetreu) die Übertragung der Winkel zur übergeordneten Sicherheits-SPS, die Verarbeitung der Sicherheitsfunktion SLS in der Sicherheits-SPS und das Auslösen der Reaktionsfunktion STO dar. Die Kommunikation basiert auf EtherCAT und dem zugehörigen Sicherheitsprotokoll FSoE. Wenn das alleinige Abschalten des Drehmoments in bestimmten Anwendungen nicht sinnvoll ist, kann zusätzlich auch die Reaktionsfunktion sichere Bremsenansteuerung (engl.: Safe Brake Control, SBC) implementiert werden. Die Pfeile in orange zeigen die Übertragung der synchron zur Distributed-Clock abgetasteten Winkel. Die Abtastung der Winkel wird um einen Vorhaltewert vor der Distributed-Clock gestartet, damit das Positionslatching des Gesamtsystems synchron erfolgt, was bei der Berücksichtigung der Reaktionszeit jedoch keine Rolle spielt [82], [110]. Die Winkel werden zur Sicherheitsanwendung in der Sicherheits-SPS übertragen, die synchron mit der Standardanwendung in der Standard-SPS ausgeführt wird. Die Sicherheitsfunktion SLS erkennt die Überschreitung der Geschwindigkeit und sendet ein Signal zur Ansteuerung von Reaktionsfunktionen (STO/SBC) zum Antrieb. In der Abbildung 16 überträgt das sicherheitsbezogene Protokoll das STO/SBC-Signal im gleichen Zyklus zum Antrieb, was jedoch im ungünstigsten Fall einen Zyklus verzögert erfolgen kann. Die Antriebssteuerung verarbeitet die empfangenen Daten so, dass Sollwerte taktsynchron mit dem nächsten Distributed-Clock-Signal übernommen werden. Das Ansteuern der Abschaltpfade für die Reaktionsfunktionen wird durch den Block T_{STO} rechts unten in der Abbildung dargestellt. Bei der Reaktionsfunktion ist eine Synchronisation mit dem Distributed-Clock-Signal nicht erforderlich.

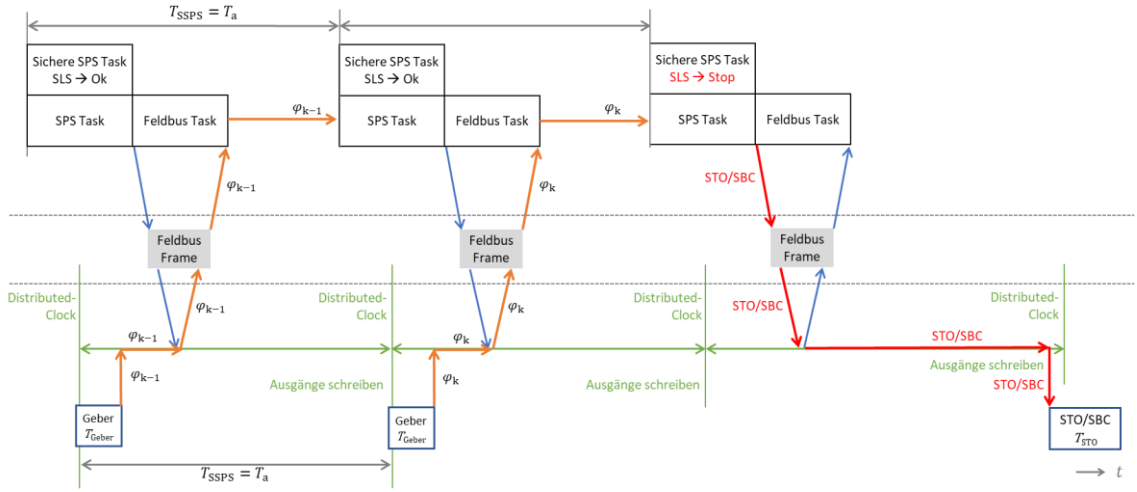


Abbildung 16: Reaktionszeit mit übergeordneter Sicherheits-SPS (nicht maßstabgetreu).

Zur Bestimmung der Reaktionszeit wird die Totzeit der Geschwindigkeit und der Fall, dass die Geschwindigkeitsüberschreitung durch die Mittelwertbildung erst einen Zyklus später erkannt wird, berücksichtigt. Daraus ergibt sich in der beschriebenen Konfiguration eine maximale Reaktionszeit mit einer übergeordneten Sicherheits-SPS von:

$$T_{\text{Reaktion,max}} = T_t + 2\,T_{SSPS} + T_{SPDU} + T_a + T_{STO} = \frac{9\,T_a}{2} + T_{STO} \tag{8}$$

Beispielrechnung mit Industrieroboter

Nachfolgend soll an einem Beispiel die Relevanz der Zykluszeit einer übergeordneten Sicherheits-SPS und der sicherheitsbezogenen Kommunikation veranschaulicht werden. Ein Knickarm-Industrieroboter soll mit reduzierter Geschwindigkeit betrieben werden. Er beschleunigt mit maximaler Beschleunigung und beim Überschreiten des Geschwindigkeitsgrenzwerts soll eine geeignete Reaktionsfunktion aufgerufen werden. Die maximale Beschleunigung des TCPs des Knickarm-Industrieroboters beträgt in diesem Beispiel:

$$a_{\text{max,TCP}} = 80\ \frac{\text{m}}{\text{s}^2} \tag{9}$$

Die ISO 10218 definiert für den Betrieb mit reduzierter Geschwindigkeit einen Grenzwert von 0,25 m/s [73]. Es sollte jedoch auch möglich sein, geringere Geschwindigkeiten als Grenzwert zu definieren. Die sicher begrenzte Geschwindigkeit für die Kollaboration sei hier:

$$v_{\text{SLS,TCP}} = 0{,}1\ \frac{\text{m}}{\text{s}} \tag{10}$$

Die Zeit für die Verarbeitung der Reaktionsfunktion STO kann je nach Implementierung unterschiedlich ausfallen. Im Folgenden wird

$$T_{\mathrm{STO}} = 0{,}2\ \mathrm{ms} \tag{11}$$

angenommen.

Für die Bewegung des TCPs des Roboters gelten die allgemeinen Bewegungsgesetze. Somit gilt nach dem Zeit-Geschwindigkeit-Gesetz:

$$\vec{v}(t) = \vec{v}_0 + \vec{a}t \tag{12}$$

Und nach dem Zeit-Weg-Gesetz

$$\vec{s}(t) = \vec{s}_0 + \vec{v}_0 t + \frac{1}{2}\vec{a}t^2 \tag{13}$$

Der Einfluss der Zykluszeit der übergeordneten Sicherheits-SPS für die Bestimmung der Reaktionszeit wird ersichtlich, wenn zwei verschiedene Sicherheits-SPSen mit unterschiedlichen Zykluszeiten betrachtet werden:

$$T_{\mathrm{SSPS1}} = T_{\mathrm{a}} = 10\ \mathrm{ms} \tag{14}$$

$$T_{\mathrm{SSPS2}} = T_{\mathrm{a}} = 1\ \mathrm{ms} \tag{15}$$

Im Folgenden werden die Richtungskomponenten der physikalischen Größen wie Geschwindigkeit, Beschleunigung und Strecke nicht berücksichtigt, sondern nur deren Betrag. Der Industrieroboter, der aus dem Stillstand mit maximaler Beschleunigung beschleunigt, erreicht die vorgegebene maximale Geschwindigkeit für SLS nach:

$$t = \frac{v}{a} = \frac{0{,}1\ \frac{\mathrm{m}}{\mathrm{s}}}{80\ \frac{\mathrm{m}}{\mathrm{s}^2}} = 1{,}25\ \mathrm{ms} \tag{16}$$

Durch Einsetzen der Zykluszeiten in (8) ergeben sich die Reaktionszeiten

$$T_{\mathrm{Reaktion,max,SSPS1}} = 45{,}2\ \mathrm{ms} \tag{17}$$

und

$$T_{\mathrm{Reaktion,max,SSPS2}} = 4{,}7\ \mathrm{ms} \tag{18}$$

Durch Einsetzen der Reaktionszeiten in (12) und (13) kann die Geschwindigkeit und die zurückgelegte Strecke nach Auslösen der Reaktionsfunktion ermittelt werden.

Die erreichte Geschwindigkeit berechnet sich durch:

$$v(T_{\mathrm{Reaktion,max}}) = v_{\mathrm{SLS,TCP}} + a_{\mathrm{max,TCP}} \cdot T_{\mathrm{Reaktion,max}} \tag{19}$$

Diese ergibt für die beiden SPSen mit unterschiedlicher Reaktionszeit:

$$v(T_{\mathrm{Reaktion,max,SSPS1}}) = 0{,}1\ \frac{\mathrm{m}}{\mathrm{s}} + 80\ \frac{\mathrm{m}}{\mathrm{s}^2} \cdot 0{,}0452\ \mathrm{s} = 3{,}716\ \frac{\mathrm{m}}{\mathrm{s}} \tag{20}$$

$$v(T_{\mathrm{Reaktion,SSPS2}}) = 0{,}1\ \frac{\mathrm{m}}{\mathrm{s}} + 80\ \frac{\mathrm{m}}{\mathrm{s}^2} \cdot 0{,}0047\ \mathrm{s} = 0{,}476\ \frac{\mathrm{m}}{\mathrm{s}} \tag{21}$$

Die Strecke, die aufgrund der Reaktionszeit zurückgelegt wird, berechnet sich wie folgt:

$$s_{\text{Reaktion}} = v_{\text{SLS,TCP}} \cdot T_{\text{Reaktion}} + \frac{1}{2} a_{\text{max,TCP}} \cdot {T_{\text{Reaktion}}}^2 \tag{22}$$

Durch Einsetzen der Reaktionszeiten der entsprechenden SPSen ergibt die jeweilige zurückgelegte zusätzliche Strecke:

$$s_{\text{Reaktion,SSPS1}} = 0{,}1 \frac{\text{m}}{\text{s}} \cdot 0{,}0452\ \text{s} + 40 \frac{\text{m}}{\text{s}^2} \cdot (0{,}0452\ \text{s})^2 = 86{,}2\ \text{mm} \tag{23}$$

und

$$s_{\text{Reaktion,SSPS2}} = 0{,}1 \frac{\text{m}}{\text{s}} \cdot 0{,}0047\ \text{s} + 40 \frac{\text{m}}{\text{s}^2} \cdot (0{,}0047\ \text{s})^2 = 1{,}35\ \text{mm} \tag{24}$$

Das vorgestellte Beispiel mit einem Industrieroboter zeigt, dass für die Kollaboration von Menschen und Maschinen eine hohe Rechenleistung und kurze Zykluszeiten von großer Bedeutung sind. Die Gleichungen (20) und (23) zeigen, dass Sicherheits-SPSen mit hohen Reaktionszeiten für kollaborative Anwendungen nicht geeignet sind, da die Sicherheit von Personen gefährdet werden kann. Bei einer vergleichsweise hohen Zykluszeit erreicht der Roboter nach der Reaktionszeit mehr als das 37-fache seiner erlaubten Geschwindigkeit und legt über 8,6 cm an zusätzlicher Strecke zurück. Dabei ist anzumerken, dass der Roboter zu diesem Zeitpunkt noch nicht den Stillstand erreicht hat. Um den genauen Anhalteweg und die genaue Anhaltezeit ab Auslösen des Stoppsignals zu ermitteln, wird auf die ISO 10218 [73] verwiesen. Dabei ist zu berücksichtigen, dass die deutlich höhere Geschwindigkeit zu einem längeren Anhalteweg und zu einer längeren Anhaltezeit führt. Es wird auf die ISO/TS 15066 [74] verwiesen, um einen geeigneten Sicherheitsabstand für die Kollaboration zwischen Roboter und Personen zu definieren. Der Sicherheitsabstand ist von einigen Faktoren, unter anderem von der Reaktionszeit und vom Anhalteweg des Robotersystems abhängig. Erst die hohe Übertragungsgeschwindigkeit der sicherheitsbezogenen Daten und die schnelle Verarbeitung dieser Daten in einer übergeordneten Sicherheits-SPS mit kurzen Zykluszeiten ermöglichen kurze Reaktionszeiten und damit geringe Sicherheitsabstände. Die kurzen Sicherheitsabstände sind Voraussetzung für eine wirkungsvolle MRK.

3.2 Sicherheitsbezogene Sensoren und Aktoren

Mit den heutigen schnellen ethernetbasierten Feldbussen kann die zentrale Maschinensteuerung zunehmend Aufgaben übernehmen, die bisher im Antrieb ausgeführt wurden. Die Verarbeitung von Motion-Control-Informationen in einer überlagerten Sicherheits-SPS ist hierbei der nächste Schritt [111]. Denn nur mit einem zentralisierten Ansatz ist es möglich, das Gesamtverhalten einer Maschine sicherheitsbezogen zu berechnen. Dies ist insbesondere für die Kollaboration von Maschinen und Menschen wichtig, da erst das Zusammenwirken einzelner Komponenten zur Gefahr für den Menschen werden kann.

Für die Realisierung einer Automatisierungslösung mit funktionaler Sicherheit ist die Wahl der verwendeten Komponenten von Bedeutung, da diese maßgeblich den Verdrahtungsaufwand, die Komplexität und die Kosten bestimmen. Für die Ausführung von Sicherheitsfunktionen in der zentralen Sicherheits-SPS werden sicherheitsbezogene Messwerte benötigt. Abbildung 17 zeigt zwei gängige Methoden zur Übertragung von sicherheitsbezogenen Winkeln an die übergeordnete Sicherheits-SPS, um eine Bewegung sicherheitsbezogen zu überwachen. Auf der linken Seite (a) ist ein Aufbau mit zwei Drehgebern dargestellt. Ein Drehgeber überträgt einen hochauflösenden Winkel an das Antriebssystem. Ein weiterer sicherheitsbezogener Drehgeber auf Basis eines Feldbus-Sicherheitsprotokolls wie PROFINET/PROFIsafe oder EtherCAT/FSoE wird an der Last des Motors montiert. Dieser misst den sicherheitsbezogenen Winkel der Motorlast und stellt diesen der übergeordneten Sicherheits-SPS bereit. Diese Implementierung ist mit einem hohen Verdrahtungsaufwand verbunden, da zwei Drehgeber verwendet werden. Außerdem sind Drehgeber als gekapselte Subsysteme mit zertifizierter Logik zur Bereitstellung einer sicherheitsbezogenen Position über ein sicherheitsbezogenes Protokoll mit erheblichen Kosten verbunden.

Die Abbildung 17 (b) zeigt einen Aufbau mit einem Drehgeber, der auf einer sicherheitsbezogenen digitalen Schnittstelle [27] basiert und sowohl einen sicherheitsbezogenen als auch einen hochauflösenden Winkel für die Regelalgorithmen in der Antriebssteuerung zur Verfügung stellt. Für diese Art von Drehgeber als gekapseltes Subsystem wird vom Hersteller in der Regel eine externe Diagnose gefordert, die meist von der Sicherheitslogik in der Antriebssteuerung ausgeführt wird. Diese zweikanalige Sicherheitslogik in der Antriebssteuerung in Form einer zusätzlichen Sicherheitskarte implementiert die SCLs der digitalen Protokolle der Drehgeberschnittstelle (z. B. EnDat 3, SCS open link, HIPERFACE DSL) und führt neben den Diagnosetests auch Einzelachs-Sicherheitsfunktionen aus. Die Weiterleitung an die übergeordnete Sicherheits-SPS kann mit dem verwendeten Feldbus und dem zugehörigen Sicherheitsprotokoll realisiert werden. Dieser Ansatz bietet

ebenfalls keine optimale Lösung, da er auf einer zweikanaligen Logik in der Antriebssteuerung basiert, die aus Kostengründen oft nicht gewünscht ist. Ebenso entsteht ein Overhead für die Übertragung der sicheren Position zur Sicherheits-SPS, da heutige digitale Drehgeberschnittstellen oft schon Maßnahmen zur sicherheitsbezogenen Datenübertragung nach dem Gray-Channel-Prinzip beinhalten [27]. In diesem Fall wird der sicherheitsbezogene Positionswert zunächst in der sicheren Logik in der Antriebssteuerung dekodiert und anschließend für das Sicherheitsprotokoll wieder als SPDU kodiert.

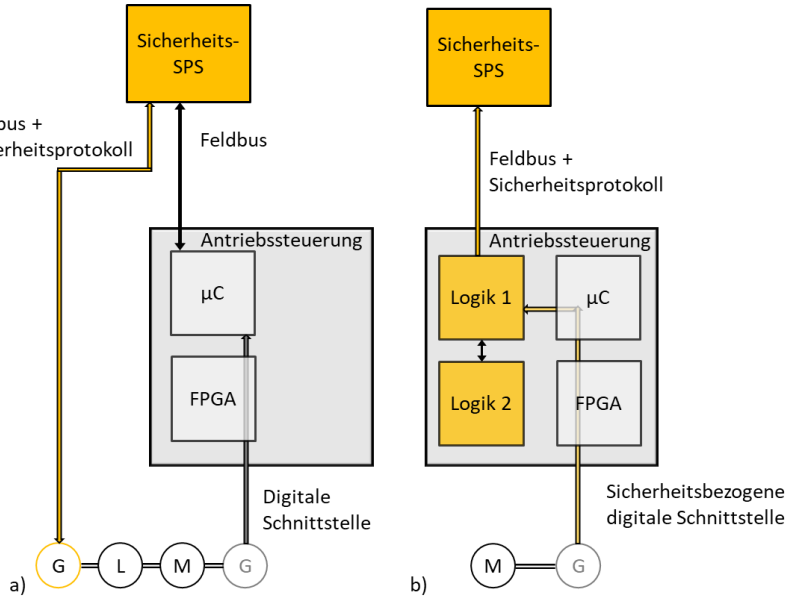


Abbildung 17: Gängige Methoden einen sicherheitsbezogenen Winkel an die übergeordnete Sicherheits-SPS zu übertragen.

Insbesondere dann, wenn Sicherheitsfunktionen und Diagnosetests von einer übergeordneten Sicherheits-SPS ausgeführt werden, kann die Komplexität von Sensoren und Antrieben reduziert werden. Es ist lediglich sicherzustellen, dass die Übertragung der Messwerte der sicherheitsbezogenen Sensoren und die Ansteuerung der Aktoren über eine sicherheitsbezogene Kommunikation gemäß IEC 61784-3 erfolgt.

In [110] wird ein Ansatz für feldbusbasierte Sicherheitsfunktionen beschrieben, bei dem kostengünstige FPGA- und Mikrocontroller-Standardkomponenten unter Verwendung von diversitärer Redundanz in einer Antriebssteuerung eingesetzt werden. Die Abbildung 18 beschreibt die Architektur einer Antriebssteuerung mit zweikanaliger Sicherheitslogik (a) und den neuen Ansatz der kombinierten Lösung, indem Teile der Standardkomponenten für die feldbusbasierten Sicherheitsfunktionen genutzt werden und vollständige redun-

dante Sicherheitslogiken in den Antrieben nicht mehr notwendig sind (b). Sicherheitsfunktionen wie SLS werden bei diesem Ansatz nicht mehr in den Antriebssteuerungen, sondern in einer übergeordneten Sicherheits-SPS ausgeführt.

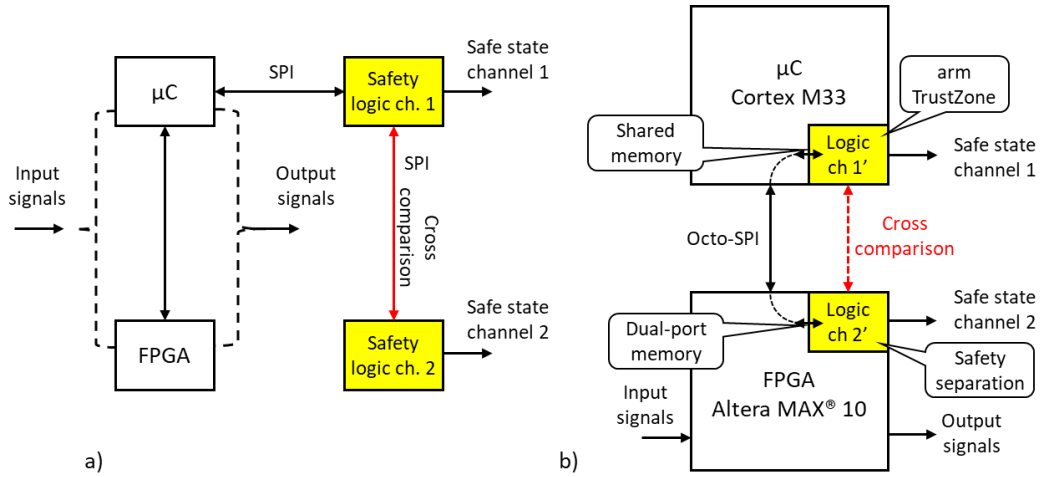


Abbildung 18: Sicherheitsarchitekturen von Antriebssteuerungen [110].

Die vorgestellte Sicherheitsarchitektur einer Antriebssteuerung nach Abbildung 18 (b) zeigt, dass sich die Komplexität von Antriebssystemen reduzieren lässt, wenn die übergeordnete zentrale Sicherheits-SPS die entsprechenden Sicherheitsfunktionen und Diagnosetests ausführt.

3.3 Topologie für gemischt-kritische Anwendungen

Sicherheitsfunktionen sind heute integraler Bestandteil von Automatisierungslösungen. Um die Komplexität, den Aufwand und die Kosten zu reduzieren, wird im Folgenden eine Sicherheitslösung mit zentraler Diagnose auf Basis einer Compound-SPS und einer effizienten Anbindung von Sensoren und Aktoren vorgestellt. Insbesondere gemischt-kritische Systeme sollen den Grundstein für zukünftige Automatisierungslösungen bilden, um die Umsetzung von funktional sicheren Anwendungen zu vereinfachen und die Komplexität zu reduzieren. Abbildung 19 veranschaulicht auf der linken Seite den klassischen Ansatz mit einer Sicherheits-SPS und einer Standard-SPS, einem Antrieb mit zusätzlicher Sicherheitskarte und zwei Gebern für die sicherheitsbezogene und nicht sicherheitsbezogene Position. Dieser klassische Ansatz, der mit einem hohen Verdrahtungsaufwand, hohen Kosten und einer erhöhten Komplexität verbunden ist, widerspricht dem Grundprinzip der funktionalen Sicherheit, bei dem eine geringe Komplexität typischerweise weniger Fehler aufweist. Auf der rechten Seite wird die neue Topologie für gemischt-kritische Anwendungen gezeigt. Eine Compound-SPS führt sowohl die sicherheitsbezogene als

auch die nicht sicherheitsbezogene Applikation aus. Die Antriebssteuerung kann mit kostengünstigen Standardkomponenten realisiert werden und besitzt keine zertifizierte Sicherheitslogik [110]. Bei diesem Ansatz bieten sich Drehgeber mit sicherheitsbezogenen digitalen Schnittstellen [27], [82] und gemäß ISO 9001 qualitätsgesicherte Standard-Sensoren an, da eine zentrale externe Diagnose von der übergeordneten Sicherheits-SPS ausgeführt wird.

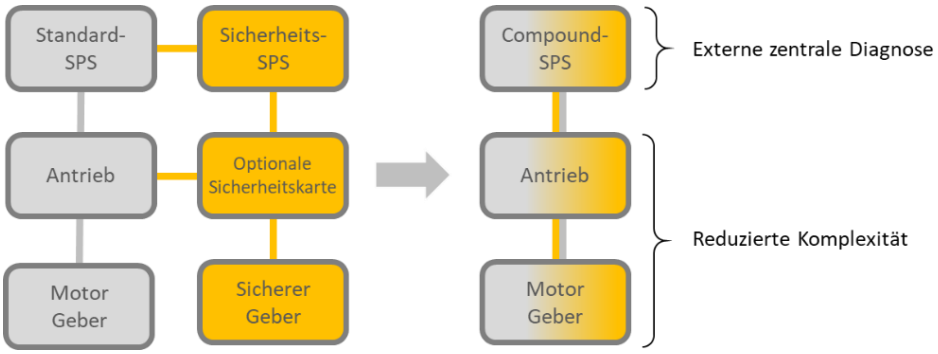


Abbildung 19: Topologie für gemischt-kritische Anwendungen.

3.4 Mehrachs-System mit zentraler Sicherheitssteuerung

Die Sicherheitslösung mit zentralen Sicherheitsfunktionen und Diagnosetests basiert auf einer rückwirkungsfreien Compound-SPS, bei der die Sicherheits- und die Standard-SPS sicher voneinander isoliert sind. Werden zur Realisierung der Compound-SPS zwei Steuerungen in einem Gehäuse implementiert, kann die Kommunikation zwischen den Steuerungen auf Basis von unterschiedlichen Schnittstellen wie dem Serial Peripheral Interface (SPI) oder Inter-Integrated-Circuit (I^2C)-Bus erfolgen. Basiert die Compound-SPS auf einem Multi-Core SoC, können Shared Memory Bereiche für den Datenaustausch zwischen den zwei Steuerungen genutzt werden. Nachfolgend wird ein für Roboter, Cobots oder FTFs geeignetes Mehrachs-System vorgestellt, das von einer übergeordneten Compound-SPS gesteuert, sicher überwacht und im Fehlerfall durch eine Reaktionsfunktion sicher abgeschaltet wird.

3.4.1 Ansteuerung von Reaktionsfunktionen im Antrieb

Die Ansteuerung der Reaktionsfunktionen des Mehrachs-Systems, wie z. B. die Motorbremse mit SBC oder die Abschaltpfade im Antrieb mit STO, erfolgt über ein sicherheitsbezogenes Protokoll wie EtherCAT/FSoE oder PROFINET/PROFIsafe. Die Sicherheits-SPS ist der Master der sicherheitsbezogenen Protokolle und die Antriebe bzw. Encoder

sind dementsprechend Slaves. Da EtherCAT häufig in der Antriebstechnik für die Ansteuerung von Antrieben eingesetzt wird, wird im vorgestellten Sicherheitskonzept das zugehörige sicherheitsbezogene Protokoll FSoE zur Ansteuerung der Reaktionsfunktionen SBC und STO verwendet. Für die redundante Ansteuerung der beiden Fehlerreaktionsfunktionen werden je zwei Bits im Sicherheitsprotokoll verwendet. Bei STO werden z. B. die High-Side-Transistoren gemeinsam und die Low-Side-Transistoren gemeinsam durch jeweils ein Bit über FSoE abgeschaltet. Für die Übertragung von wenigen Daten (1 oder 2 Byte) eignet sich FSoE für eine sicherheitsbezogene Kommunikation. Der Overhead des Protokolls steigt, wenn größere Datenmengen versendet werden sollen. Für je zwei Byte sicherheitsbezogene Daten sind zwei zusätzliche Byte für die zyklische Redundanzprüfung (engl.: Cyclic Redundancy Check, CRC) erforderlich [109].

3.4.2 Sicherheitsbezogene Position in der Sicherheitssteuerung

Für Sicherheitsfunktionen, die die Position, Geschwindigkeit oder Beschleunigung beweglicher Maschinenteile überwachen, werden nach dem hier vorgestellten Konzept die sicherheitsbezogenen Winkel der einzelnen Motoren zyklisch an die Sicherheits-SPS übertragen. Die zwei bereits gezeigten Ansätze, sicherheitsbezogene Winkel an die übergeordnete Sicherheits-SPS zu übertragen (siehe Abbildung 17) bieten keine geeigneten Lösungen für eine MRK.

In [112] wird das Konzept der Verkettung von SPDUs mit einer statischen Konfiguration der Prozessdaten beschrieben, um unnötigen Overhead bei der sicherheitsbezogenen Datenübertragung zu vermeiden. Der Ansatz dabei ist, die Drehgeber-SPDUs immer direkt an die Feldbus-Slave-SPDUs (FSoE-SPDU) anzuhängen und in der übergeordneten Sicherheits-SPS auszuwerten. Das bedeutet, dass der SCL von der dezentralen sicheren Logik im Antrieb in die zentrale übergeordnete Sicherheits-SPS verlagert wird. Gleichzeitig bedeutet dies auch, dass das Drehgeber-Protokoll der sicherheitsbezogenen digitalen Schnittstelle nun auch über den ethernetbasierten Standard-Feldbus als Teil des Gray-Channels übertragen wird. In der Abbildung 20 ist die Gray-Channel-Kommunikation für die Übertragung der sicherheitsbezogenen Drehgeber-Daten zu sehen.

Der Ansatz, die sicherheitsbezogene Kommunikation auf Basis der sicherheitsbezogenen digitalen Drehgeberschnittstelle in der übergeordneten Sicherheits-SPS auszuwerten, erfüllt alle Anforderungen der IEC 61784-3, da Fehler während der Datenübertragung durch die Sicherheitsmaßnahmen im SCL erkannt werden.

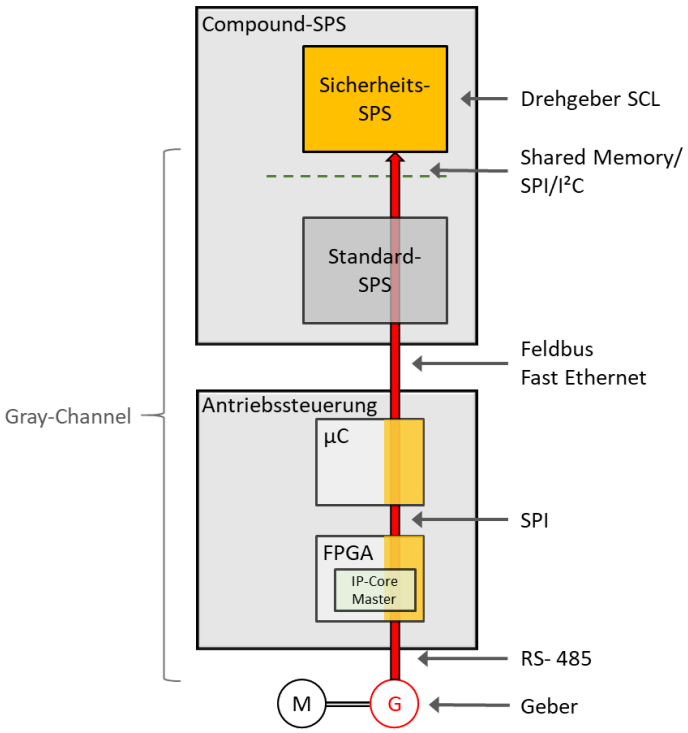


Abbildung 20: Übertragung der Drehgeber-SPDUs zur Sicherheits-SPS.

3.4.3 Sicherheitsbezogene Ströme in der Sicherheitssteuerung

Nach [113] sollen Robotersysteme für die Kollaborationsart PFL auch Sicherheitsfunktionen für eine sicherheitsbezogene Drehmoment-Überwachung verfügen. Die Sicherheitsfunktion sicher begrenztes Drehmoment (engl.: Safely-Limited Torque, SLT) verhindert, dass ein Motor ein festgelegtes Drehmoment überschreitet. Zur Umsetzung der Sicherheitsfunktion SLT in der übergeordneten Sicherheits-SPS können sicherheitsbezogene Stromwerte ausgewertet werden, um das sicherheitsbezogene Drehmoment von Permanent Magnet Alternating Current (PMAC)-Synchronmotoren zu bestimmen:

$$m(t) = K_\mathrm{T}\, i_\mathrm{q}(t) \tag{25}$$

Die drehmomentbildende Stromkomponente i_q kann mithilfe der Clarke/Park-Transformation aus den Strömen i_u, i_v, i_w und dem Kommutierungswinkel φ_e bestimmt werden. Der Kommutierungswinkel berechnet sich bei PMAC-Motoren aus dem Rotorwinkel, der Motorpolpaarzahl und dem Kommutierungsoffset. Bei der Drehmomentkonstanten K_T des PMAC-Motors sind eventuelle Fertigungstoleranzen, eine eventuelle Abhängigkeit vom Rotorwinkel und die Temperaturabhängigkeit zu berücksichtigen [114]. Für eine Kollaboration nach PFL werden die einzelnen Motoren auf ein Maximal-Drehmoment überwacht.

Der Ansatz, das Geber-Protokoll zentral in der sicherheitsbezogenen Steuerung statt dezentral in der Antriebssteuerung auszuwerten, erfordert kaum zusätzlichen Aufwand. Dabei wird lediglich der Gray-Channel um den Antrieb, den verwendeten Standard-Feldbus und den nicht sicherheitsbezogenen Teil der Compound-SPS erweitert. Zunächst wird bei der Strommessung sichergestellt, dass die Messdaten sicher erfasst werden [114]. Zur Messung der Motorströme bietet sich wie üblich in der funktionalen Sicherheit die Redundanz an. In [112] und [114] werden Ansätze beschrieben, die Motorströme mit Stromwandlern (engl.: Current Transducer, CT) und Dezimierungsfiltern ($sinc^k$ in der Abbildung 21) sicherheitsbezogen zu erfassen und diese zur Sicherheits-SPS zu übertragen. Das Prinzip der Strommessung ist in der Abbildung 21 dargestellt.

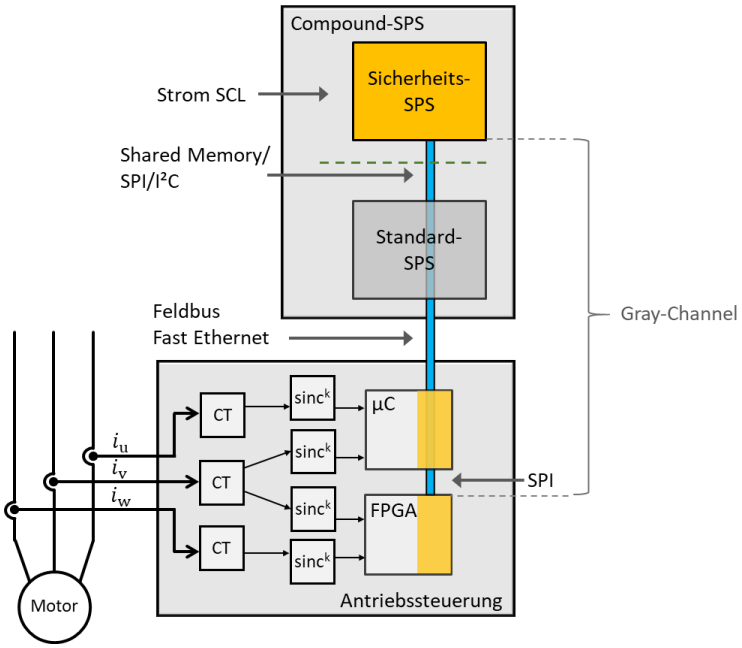


Abbildung 21: Sicherheitsbezogene Strommessung und Übertragung der Strom-SPDUs zur Sicherheits-SPS.

Da FSoE als Sicherheitsprotokoll für EtherCAT nur für die Übertragung von geringen Datenmengen geeignet ist, wird in [112] der Ansatz beschrieben, die Stromdaten an die FSoE-SPDU zu verketten und gemeinsam an die Sicherheits-SPS zu senden, anstatt die Stromdaten als Teil der FSoE-SPDU zu übertragen. Dabei werden CRCs über die Stromdaten und über den FSoE-CRC gebildet, sodass die Stromwerte sicher einer Achse zugewiesen werden können. Durch die Verkettung der Strom-SPDUs an die FSoE-SPDUs werden die Sicherheitsmaßnahmen für die Absicherung des Datentransports nicht nochmals explizit für die Ströme definiert, da das FSoE-Protokoll diese Maßnahmen als sicherheitsbezogenes Protokoll nach IEC 61784-3 bereits implementiert.

3.4.4 Allgemeine Architektur

Die Abbildung 22 zeigt die allgemeine Architektur für die Sicherheitslösung mit zentraler Sicherheits-SPS. Die übergeordnete Compound-SPS, bestehend aus einer Sicherheits-SPS und einer Standard-SPS, steuert, überwacht und vernetzt ein Mehrachs-System. Die logische Vernetzung der OT-Komponenten mit der Compound-SPS wird über verschiedene Pfeile dargestellt. Die schwarze Linie zwischen den Antrieben und der Standard-SPS zeigt den verwendeten Feldbus. Das verwendete sicherheitsbezogene Black-Channel-Protokoll wird durch die gelben Pfeile dargestellt. Die blauen und roten Pfeile zeigen die Gray-Channel-Kommunikation zur Übertragung der sicherheitsbezogenen Strom- und Winkel-Messwerte an die übergeordnete Sicherheits-SPS.

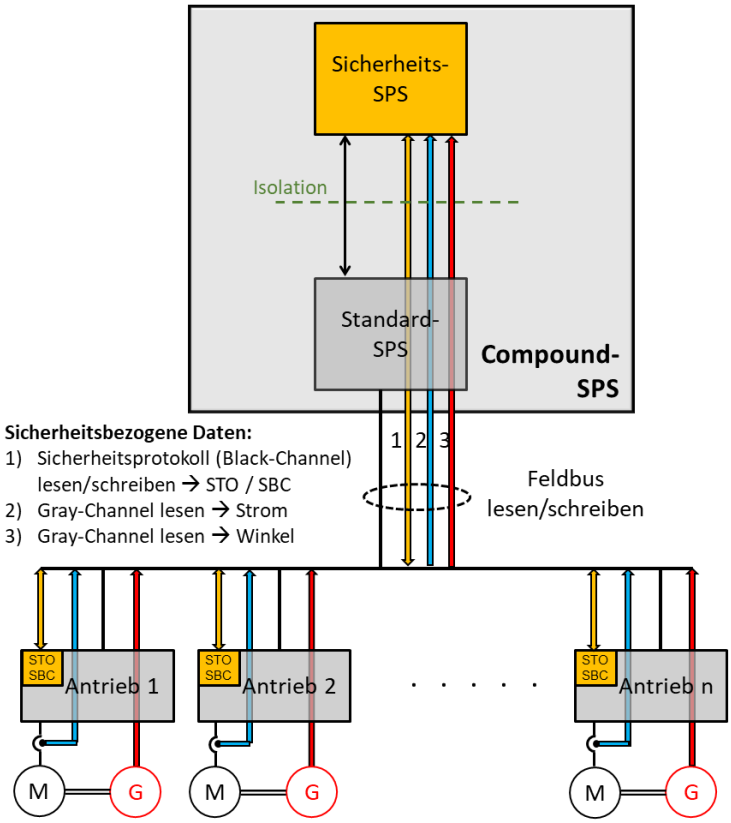


Abbildung 22: Blockschaltbild für ein gemischt-kritisches Mehrachs-System mit zentraler Diagnose.

3.5 Sicherheitsbezogenes Blockschaltbild

Nach ISO 13849-1 Kategorie 3 erfolgt die Ausführung der Sicherheitsfunktionen in zwei Kanälen der Sicherheits-SPS. Im Fehlerfall oder bei Überschreiten der Grenzwerte (Position, Geschwindigkeit etc.) wird eine Reaktionsfunktion in Form von Stoppfunktionen (STO, SS1, SS2 etc.) ausgeführt. In [37] wird die Anwendung der ISO

13849-1 beschrieben und dabei wird auch auf die Kombination und Verschaltung von SRP/CS eingegangen. In der Regel ist eine Verschaltung mehrerer Subsysteme erforderlich, von denen jedes einen Teil der Sicherheitsfunktion ausführt. Diese SRP/CS als Subsysteme können unterschiedliche Technologien und auch unterschiedliche Kategorien oder PL umfassen. Jedes dieser Subsysteme kann in der Regel auf die vorgesehenen Architekturen abgebildet werden und besteht demnach aus den logischen Blöcken Eingang, Logik und Ausgang.

Im Folgenden werden Sicherheitsfunktionen betrachtet, die eine Bewegung überwachen (SLP, SLS, SLA etc.). Diese Sicherheitsfunktionen werden bei dem vorgestellten Ansatz mit der zentralen Sicherheits-SPS von drei verschiedenen SRP/CS Subsystemen ausgeführt: Geber, Sicherheits-SPS und Antrieb. Die Norm IEC 61800-5-2 definiert, dass die Sicherheitsfunktionen, die eine Bewegung überwachen, für ihre Ausführung sicherheitsbezogene Winkel benötigen. Bussysteme können dabei als Verbindungsmittel für die SRP/CS eingesetzt werden. Bussysteme bzw. sicherheitsbezogene Protokolle, die den Anforderungen der Norm IEC 61784-3 genügen, lassen sich ohne jeglichen Zusatz mit der Norm ISO 13849-1 verwenden [37]. Die Auswertung und Dekodierung der SPDUs erfolgt im SCL in der Sicherheits-SPS. Es werden die sicherheitsbezogenen Daten aus der SPDU extrahiert und dem Sicherheitsprogramm bereitgestellt. Sollten beim Dekodieren und Extrahieren der SPDU Fehler aufgedeckt werden, wird eine geeignete Fehlerreaktion initiiert. In Abbildung 23 ist die Verschaltung der drei jeweils redundanten Teilsysteme, die gemeinsam die Sicherheitsfunktion realisieren und im Fehlerfall die Energiezufuhr im Antrieb abschalten, zu sehen.

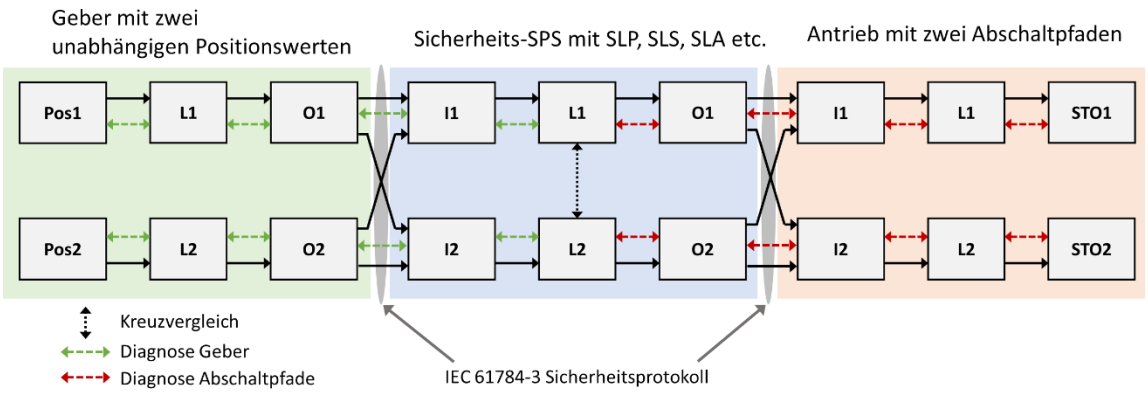


Abbildung 23: Verschaltung von Teilsystemen für die zentrale Ausführung von Sicherheitsfunktionen.

Die sicherheitsbezogene Winkelerfassung für die Bewegungs-Sicherheitsfunktionen erfolgt beim vorgestellten Ansatz mit Drehgebern als gekapselte Subsysteme mit externer Diagnose [27]. Diese sind meist zweikanalig ohne Sicherheitslogik zertifiziert und die

benötigte externe Diagnose wird in der Regel mit einem Kreuzvergleich realisiert. Die fehlende vollständige Sicherheitslogik im Drehgeber ist in der Abbildung 23 an der fehlenden Verbindung für den Kreuzvergleich zwischen L1 und L2 zu erkennen. Die zentrale Sicherheits-SPS hat die Aufgabe, die sicherheitsbezogenen Daten des Gebers auszuwerten und im Fehlerfall, z. B. bei einem fehlerhaften Kreuzvergleich, das System in einen sicheren Zustand zu bringen, da der Geber als Subsystem dies nicht leisten kann und aus Kostengründen auch nicht leisten soll. Alternativ können auch zwei diversitäre, nicht sicherheitsbezogene Geber eingesetzt werden, die ebenfalls von der übergeordneten Sicherheits-SPS mittels Kreuzvergleich geprüft werden.

Um die Komplexität der Antriebe zu reduzieren, fehlt analog zum Geber auch beim Antrieb die Verbindung für den Kreuzvergleich. Dies ist eine vereinfachte Darstellung des Konzepts aus [110]. Der Fokus liegt bei diesem Ansatz auf der Implementierung von feldbusbasierten Sicherheitsfunktionen. Demzufolge werden in der Antriebssteuerung keine komplexen Sicherheitsfunktionen (wie SLS, SLT etc.) ausgeführt. Die Sicherheits-SPS versetzt den Antrieb bei Überschreiten definierter Grenzwerte über entsprechende Bits im Sicherheitsprotokoll in den sicheren Zustand. Aufgrund der fehlenden Sicherheitslogik im Geber und im Antrieb werden die gestrichelten Pfeile (grün und rot), die die Diagnose darstellen, von beiden Systemen bis zur Sicherheits-SPS durchgezogen.

Neben der detaillierten Ansicht von Sicherheitsfunktionen, die in Teilen von mehreren Subsystemen ausgeführt werden, wird das vereinfachte zusammengefasste Blockschaltbild in Abbildung 24 vorgestellt. Es zeigt die zweikanaligen Teilsysteme: Geber, Sicherheits-SPS und die Abschaltpfade im Antrieb.

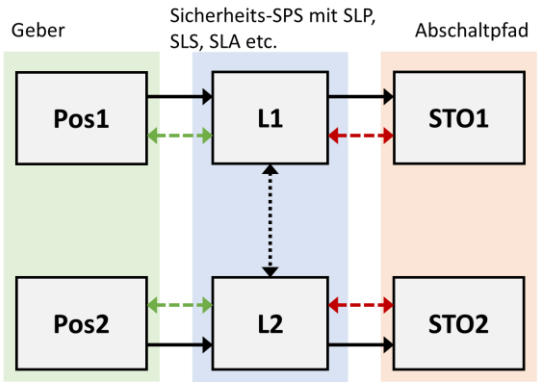


Abbildung 24: Vereinfachtes Kategorie 3 Blockschaltbild für die zentrale Ausführung von Sicherheitsfunktionen.

Wird der vorgestellte Ansatz auf ein mehrachsiges System, z. B. auf einen Roboter, ausgedehnt, so ergibt sich das in Abbildung 25 dargestellte Blockschaltbild. Mit dieser Architektur können alle Sicherheitsfunktionen durch Konfiguration der zentralen Sicherheits-SPS realisiert werden [115].

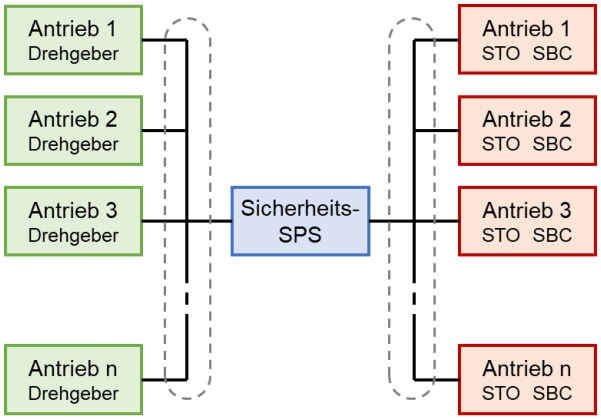


Abbildung 25: Blockschaltbild für ein sicherheitsbezogenes Mehrachs-System.

3.6 Zentrale Testung von Sicherheitsfunktionen

Durch den Einsatz von gekapselten Subsystemen ohne zertifizierte Sicherheitslogik, wie z. B. Drehgeber nach [27], Antriebe für feldbusbasierte Sicherheitsfunktionen nach [110] und diversitäre Standard-Sensoren oder Aktoren, ist eine externe Diagnose und damit eine zentrale Testung der Sicherheitsfunktionen bzw. sicherheitsbezogener Komponenten erforderlich. In [69] wird der Einsatz von Antriebssteuerungen behandelt, die Sicherheitsfunktionen umsetzen. Es wird die Anwendung der von der Norm IEC 61800-5-2 definierten Sicherheitsfunktionen und deren Umsetzung beschrieben. Beispielhaft sollen im Folgenden die in der IEC 61800-5-2 definierten Sicherheitsfunktionen STO und SBC als Fehlerreaktionsfunktionen betrachtet werden.

Tritt während des Betriebs ein Fehler auf, kann die Ausführung der Reaktionsfunktionen, d. h. die Ansteuerung der Impulssperre oder der Haltebremse, versagen. Wenn keine Maßnahmen zur Testung der Abschaltpfade oder der Haltebremse getroffen werden, kann die Fehleraufdeckung nur bei Anforderung der Sicherheitsfunktionen erfolgen. Dies reicht jedoch in den meisten Fällen nicht aus, um den geforderten PL zu erreichen.

In [41] und [69] werden geeignete Maßnahmen in Form von Diagnosetests und deren Anforderungen beschrieben, um Fehler in den Sicherheitsfunktionen und Bauteilen zu erkennen. Eine dieser Maßnahmen ist das regelmäßige Testen der beteiligten Komponenten in den Antrieben oder Sensoren mit Testsignalen. Dabei ist es nicht von Relevanz, ob die

Diagnosetests von einer Sicherheitslogik im Antrieb, im Sensor selbst oder von einer übergeordneten Sicherheits-SPS ausgeführt werden. Es ist jedoch zu beachten, dass bei der Implementierung der Diagnosetests in der übergeordneten Sicherheits-SPS eine sicherheitsbezogene Übertragung der Testsignale und der Testergebnisse erforderlich ist. Für die Übertragung der redundanten Bits als Testsignale für die Sicherheitsfunktionen STO und SBC bietet sich ein sicherheitsbezogenes Protokoll für die Kommunikation zwischen Antrieb und der übergeordneten Sicherheits-SPS an [112].

Beim Sicherheitskonzept in der vorliegenden Arbeit werden in regelmäßigen Abständen die redundanten Kanäle der Impulssperre (STO) und der Bremse (SBC) getestet, indem das Auslösen der Sicherheitsfunktionen überprüft wird. Für eine konsequente Trennung der beiden Kanäle im Antrieb werden auch die SPDUs zweikanalig übertragen. Die SPDUs beider Kanäle übertragen nach der Ausführung des Tests das Ergebnis zum Master. Die Sicherheits-SPS wertet anschließend die Testergebnisse aus und initiiert im Fehlerfall eine geeignete Fehlerreaktionsfunktion. Das logische Blockschaltbild für ein Kategorie 3 System mit externer zentraler Diagnose von Sicherheitsfunktionen im Antrieb ist in der Abbildung 26 dargestellt. Dieser Ansatz kann ebenso auf ein Sensorsystem übertragen werden.

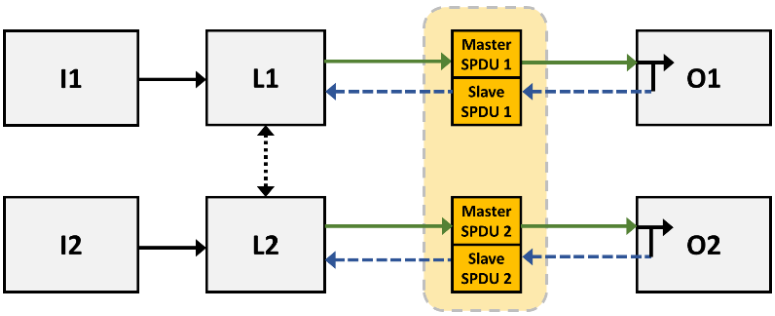


Abbildung 26: Ein zweikanaliges System mit zentraler externer Diagnose von dezentralen Ausgängen über ein sicherheitsbezogenes Protokoll.

Neben den Sicherheitsfunktionen STO und SBC können auch weitere relevante Diagnosemaßnahmen zentral durchgeführt werden. Auch die angeschlossenen sicherheitsbezogenen Drehgeber müssen laut Hersteller während des Betriebs in regelmäßigen Abständen getestet werden. Einige zertifizierte Drehgeber als gekapselte Subsysteme haben die Anforderung in regelmäßigen Abständen Zwangsdynamisierungstests durchzuführen [81]. Die sicherheitsbezogene Strommessung kann analog zu diesen sicherheitsbezogenen Drehgebern getestet werden. In einem Fehlerfall kann ein Stromsensor defekt sein und einen konstanten Stromwert anzeigen. Die Datenübertragung funktioniert sicher und fehlerfrei, der defekte Sensor kann jedoch ohne eine zentrale Diagnose oder Testung nicht

von der Sicherheits-SPS erkannt werden. In diesem Fall wird über ein sicherheitsbezogenes Protokoll ein Testbit an den Antrieb gesendet, das dafür sorgt, dass das Strommesssignal dynamisiert wird, indem sicherheitsbezogene Komponenten (wie z. B. die Dezimierungsfilter) für die sicherheitsbezogene Strommessung kurzzeitig nacheinander auf ein Referenzsignal geschaltet werden [114]. Durch das Rücklesen der Stromwerte und die Auswertung in der übergeordneten Sicherheits-SPS wird die korrekte Funktion der sicherheitsbezogenen Strommessung gewährleistet, während die Stromregelung im Antriebssystem nicht beeinflusst wird. Das erste Kirchhoffsche Gesetz, die Summe der drei Phasenströme ist Null, kann für eine Plausibilisierung der Stromwerte in der Sicherheits-SPS hinzugezogen werden.

3.7 Degradierter Betrieb

In der Fertigungs- und Maschinenautomatisierung sind Sicherheits-SPSen in der Regel so ausgelegt, dass sie bei der Erkennung von Fehlern in den sicherheitsbezogenen Komponenten den sicheren Zustand herbeiführen. Der sichere Zustand wird in fast allen Fällen durch Abschalten der Energiezufuhr erreicht, wie z. B. bei Antrieben durch die Fehlerreaktionsfunktion STO. Dieser Maschinenstopp kann bei komplexen Fertigungssystemen größere Auswirkungen haben, wenn dadurch der gesamte Fertigungsablauf gestoppt wird. Ebenso wird die Sicherheitstechnik an Maschinen häufig als eine potenzielle Ursache für ungewollte Maschinenstillstände betrachtet. Dies kann den Anreiz erhöhen, die Sicherheitstechnik zu manipulieren, damit die Maschine weiter betrieben werden kann.

3.7.1 Sicherheitsfunktionen

Der Betrieb im degradierten Zustand wurde von der ZVEI vorgestellt und beschreibt eine Alternative zur klassischen Energietrennung [116], [117]. Ziel bei dem neuen Konzept ist es, bei bestimmten Fehlern ausgewählte Maschinenfunktion mit ausreichender Sicherheit in definierten Grenzen zuzulassen. Der Begriff Degradierung beschreibt die temporäre oder dauerhafte Verringerung des Beitrags einer Sicherheitsfunktion. Als Folge des Herabsetzens des Beitrags der Sicherheitsfunktion resultiert eine Erhöhung des Restrisikos.

Da bisher bei jedem Fehler die übliche Fehlerreaktion in Form der Energietrennung durchgeführt wird, sind neue Verfahren und Methoden erforderlich, die einen sicheren Betrieb ermöglichen, obwohl ein Fehler in einer Sicherheitsfunktion erkannt wurde [118]. Nicht bei jedem Fehler ist ein Weiterbetrieb der Maschine möglich, so dass eine Fehlerbeurteilung und eine Fehlerbewertung vorzunehmen sind. Dabei wird zwischen nicht tolerierba-

ren Fehlern und tolerierbaren Fehlern unterschieden. Systematische Fehler und CCF gehören zu den nicht tolerierbaren Fehlern. Nur zufällige Fehler erlauben den degradierten Betrieb und sind tolerierbare Fehler [116]. Dieses neue Konzept mit einem qualifizierten Entscheider, der beurteilt, ob ein degradierter Betrieb zulässig ist, ist in der Abbildung 27 b) dargestellt.

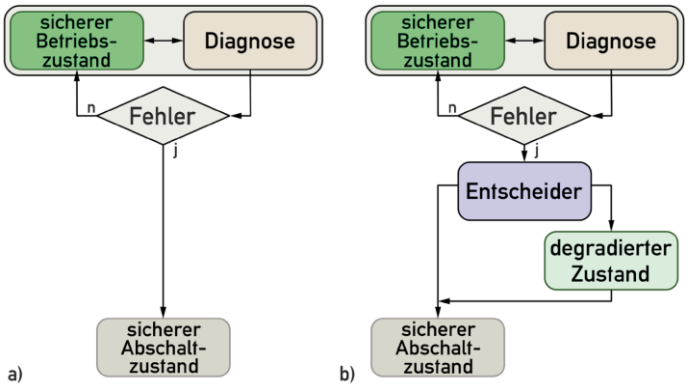


Abbildung 27: a) Klassisches Sicherheitskonzept. b) Degradierter Betrieb [116].

Die Aufgabe des Entscheiders ist es, abhängig vom erkannten Fehler, das System in den sicheren Abschaltzustand oder in den degradierten Zustand zu versetzen. Dafür ist eine qualifizierte Diagnose mit Fehlererkennung und -beurteilung erforderlich. Es gelten dabei höhere Ansprüche an die Diagnose im Vergleich zur Realisierung der Anforderungen an Kategorie 2, 3 oder 4 der ISO 13849-1, da nicht nur Fehler erkannt, sondern zusätzlich beurteilt werden müssen. Die Diagnose muss dabei berücksichtigen, dass nicht nur zufällige Hardware- oder Bauteilfehler, sondern auch systematische Fehler oder CCFs vorliegen können, die einen degradierten Betrieb nicht zulassen. Bei Betrachtung von zweikanaligen Systemen (Kategorie 3 und 4), kann bei Verlust der Sicherheitsfunktion in einem Kanal ein Strukturwechsel stattfinden: Aus Kategorie 3 oder 4 wird Kategorie 2.

3.7.2 Qualifizierte Teilsysteme

Die qualifizierte Diagnose bewertet, welcher Fehler vorliegt, wo dieser vorliegt, ob die Sicherheitsfunktion weiter ausgeführt werden kann und ein degradierter Betrieb möglich ist. Zu Beginn dieses Kapitels wurde beschrieben, dass SRP/CS als Subsysteme hintereinandergeschaltet werden und in Teilen eine Sicherheitsfunktion ausführen können. Eine Sicherheitsfunktion wird in der Regel in Teilen von einem zweikanaligen Sensor, einer zweikanaligen Sicherheits-SPS und einem zweikanaligen Leistungsantrieb ausgeführt. Nach [117] führen qualifizierte Teilsysteme ihre Diagnose innerhalb des jeweiligen Teilsystems selbst aus. Betrachtet man die Verschaltung eines Kategorie 3 Leistungsantriebs

als qualifiziertes Teilsystem mit einer Kategorie 3 Sicherheits-SPS, ist von der Darstellung in Abbildung 28 auszugehen:

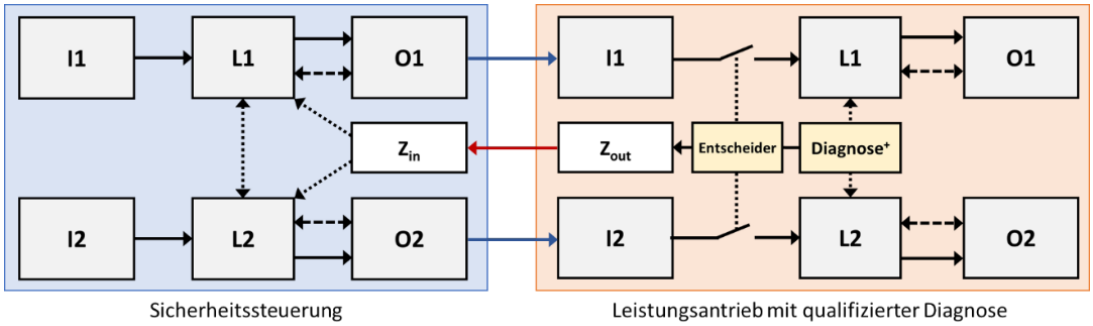


Abbildung 28: Degradierter Betrieb mit qualifizierter Diagnose im Antriebssystem.

Die Antriebssteuerung mit qualifizierter Diagnose erkennt Fehler und Zustände, die zu einer Gefahr führen können. Der Entscheider in der zweikanaligen Sicherheitslogik der Antriebssteuerung ist in der Lage den Fehler klar zuzuordnen und kann diesen als tolerierbar oder nicht tolerierbar beurteilen. Der Status, dass eine Sicherheitsfunktion im Antrieb im degradierten Betrieb ist, wird über einen logischen Meldeausgang (Z_{out}) an die Sicherheits-SPS übertragen. Liegt ein zeitlich begrenzter degradierter Betrieb vor, wird der Leistungsteil des Antriebs nach einer definierten Zeit abgeschaltet. Die Kommunikation zwischen Sicherheits-SPS und Antrieb findet in der Regel über ein sicherheitsbezogenes Protokoll statt. Jede Sicherheitsfunktion kann ein Status-Bit im sicherheitsbezogenen Protokoll enthalten, welches den degradierten Betrieb anzeigt.

3.7.3 Sicherheitssteuerung als zentraler Entscheider

Zwar zeigt die ZVEI in ihren Publikationen, dass ein degradierter Betrieb von Sicherheitsfunktionen im Einklang mit den definierten Zielen der Maschinenrichtlinie steht und mit den harmonisierenden Normen ISO 13849-1 und IEC 62061 vereinbar ist, jedoch bedarf es viel Aufwand die geforderte qualifizierte Diagnose in sicherheitsbezogene Aktoren und Sensoren zu implementieren. Die vorliegende Arbeit unterstreicht den Ansatz, Standard-Sensoren und -Aktoren auch für die funktionale Sicherheit einzusetzen und verzichtet auf eine vollwertige zweikanalige Sicherheitslogik in den entsprechenden Komponenten. Standardkomponenten können keine qualifizierte Diagnose beinhalten, da eine Zertifizierung erforderlich ist. Der Ansatz, dass jedes Teilsystem eine qualifizierte Diagnose enthält, ist dabei kaum wirtschaftlich realisierbar. Bestehende sicherheitsbezogene Sensoren und Aktoren müssen um die qualifizierte Diagnose erweitert werden. Alternativ müssten neue Sensoren und Aktoren entwickelt werden, die sowohl für die funktionale Sicherheit

zertifiziert sind, als auch den degradierten Betrieb unterstützen und eine qualifizierte zertifizierte Diagnose beinhalten. Es wird nochmals betont, dass die Anforderungen an die qualifizierte Diagnose höher sind als an die übliche Diagnose von Systemen der Kategorie 2, 3 oder 4.

Der vorgestellte Ansatz der zentralen Diagnose wird durch die in dieser Arbeit eingeführte zentrale Sicherheits-SPS für gemischt-kritische Anwendungen ermöglicht. Die heutigen schnellen sicherheitsbezogenen Protokolle wie FSoE und die heutige Prozessorleistung, die auch für Sicherheits-SPSen zur Verfügung steht, ermöglichen die Reduzierung der Komplexität von Sensoren und Aktoren, indem die übergeordnete Sicherheits-SPS Diagnoseaufgaben übernimmt.

Wenn die übergeordnete Sicherheits-SPS die Vorverarbeitung der sicherheitsbezogenen Sensorsignale oder die Ansteuerung der sicherheitsbezogenen Aktoren selbst durchführt, kann eine qualifizierte Diagnose zentral in der übergeordneten Sicherheits-SPS implementiert werden. Der Entscheider ist dann nicht in den Sensoren und Aktoren, sondern in der Sicherheits-SPS integriert. Dies wird zu einer deutlich höheren Akzeptanz des degradierten Betriebs führen, da in den Sensoren und Aktoren auf eine vollwertige zweikanalige zertifizierte Logik verzichtet werden kann. Ebenso ermöglicht dieser Ansatz den degradierten Betrieb von nicht sicherheitsbezogenen Sensoren und Aktoren, die zweikanalig realisiert sind. Die Erweiterung der Sensoren und Aktoren in Form von Hardware kann durch Software in der übergeordneten Sicherheits-SPS ersetzt werden. Eine Sicherheits-SPS mit zentralem Entscheider ist in der Abbildung 29 dargestellt. Das Fehlen der vollwertigen Sicherheitslogik in der Antriebssteuerung ist durch das Fehlen der Linie für den Kreuzvergleich zwischen L1 und L2 dargestellt.

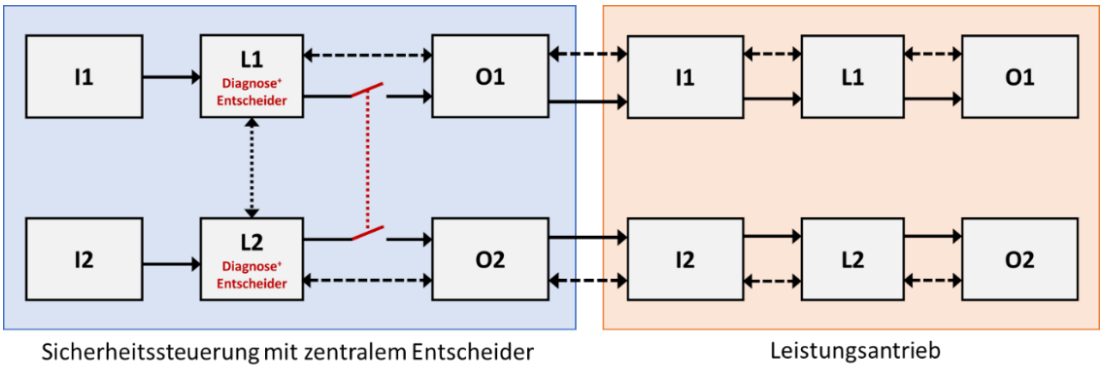


Abbildung 29: Degradierter Betrieb mit qualifizierter Diagnose als Software-Erweiterung in der Sicherheits-SPS.

3.8 Reduzierung der Komplexität in Software und Hardware

Durch die zentrale Sicherheits-SPS als Fortführung der IPC-basierten Automatisierungsphilosophie können Hardwarekomponenten in Antrieben und in Sensoren durch Softwaremodule im Sicherheitsprogramm der übergeordneten Sicherheits-SPS ersetzt werden. Eine Programmierumgebung für sicherheitsbezogene Steuerungen beinhaltet alle gängigen Compiler für die IEC 61131-3 Programmiersprachen und unterstützt damit den Aspekt der PLCopen, Standardverfahren auch im Bereich der funktionalen Sicherheit zu nutzen. Neben grafischen Programmiersprachen wie FBD wird auch die Programmierung von Sicherheitsfunktionen im PLCopen Extended Level (üblich mit der Programmiersprache ST) unterstützt. Bei der Entwicklung und Programmierung von Sicherheitsfunktionen in ST gelten die Anforderungen an SRESW und FVL. Unter Berücksichtigung der Styleguides und Anforderungen der PLCopen für die Programmierung von Sicherheitsfunktionen als FB ist davon auszugehen, dass der Verifikationsprozess durch Zertifizierungsstellen vereinfacht und beschleunigt wird. Für das Sicherheitsprogramm in FBD, das die Sicherheitsfunktionen als FBs enthält, gelten die Anforderungen für SRASW und LVL, da das Sicherheitsprogramm gut lesbar und leicht verständlich ist. Abbildung 30 zeigt die Programmierung von Sicherheitsfunktionen als FBs und die anschließende grafische Verschaltung der Sicherheitsfunktionen im Sicherheitsprogramm.

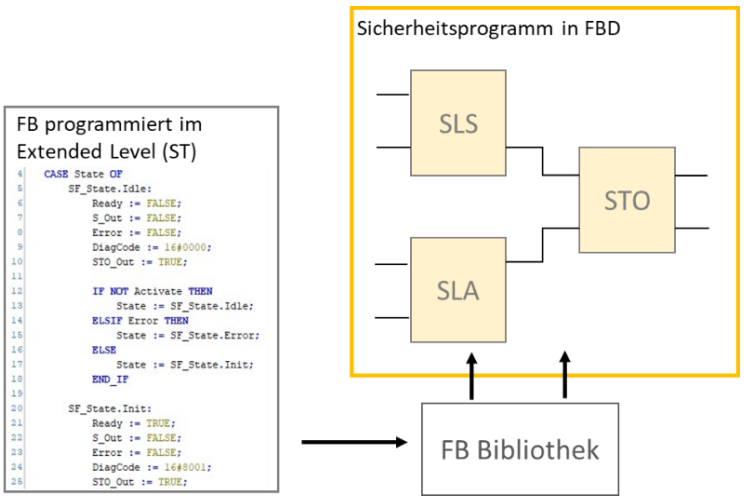


Abbildung 30: Programmierung und Aufruf einer Sicherheitsfunktion.

Bei der Implementierung von Sicherheitsfunktionen, die die Bewegung einer Achse überwachen, ist es bisher üblich, diese dezentral in der Antriebssteuerung zu implementieren. Die Systemsoftware von der Sicherheitslogik einer Antriebssteuerung, die meist aus zwei Mikrocontrollern (Kategorie 3) besteht, wird in der Regel in der Programmiersprache C

programmiert. Für die funktionale Sicherheit ist C eine Programmiersprache, die den Anforderungen von SRESW und FVL unterliegt und von der PLCopen (System Level) nicht berücksichtigt wird. Daher ist der zugrunde liegende Ansatz, das Sicherheitsprogramm in grafischen LVL-Programmiersprachen zu programmieren und die in FVL geschriebenen und validierten FBs grafisch zu implementieren, für Antriebssteuerungen nicht möglich. Neben der erhöhten Komplexität in der Software haben Sicherheitsfunktionen in der Antriebssteuerung einen weiteren Nachteil. Die Sicherheitsfunktionen müssen in der Regel auf einer entsprechenden zertifizierten zusätzlichen Hardware (Kategorie 3 Sicherheitskarte) ausgeführt werden, was erhebliche Zusatzkosten verursacht.

Die zentrale Diagnose bietet die Möglichkeit Sicherheitsfunktionen per Software frei zu konfigurieren [119]. Insbesondere die Akzeptanz des degradierten Betriebs von Sicherheitsfunktionen kann von dem softwarebasierten Ansatz profitieren. Mithilfe einer Entwicklungsumgebung für SPSen können Sicherheitsfunktionen um den degradierten Betrieb erweitert werden, anstatt ihn fest in Sensor- oder Aktor-Teilsystemen zu integrieren. Dies bietet Anwendern große Freiheiten ihr System zu konfigurieren und an ihre Bedürfnisse anzupassen. Ein zentraler Entscheider in einer übergeordneten Sicherheits-SPS, der einen degradierten Betrieb für eine Vielzahl von Sicherheitsfunktionen ermöglicht, erlaubt auch eine qualifizierte Diagnose für das Gesamtsystem. Mithilfe einer detaillierten Analyse des Restrisikos bei Ausfall eines Kanals einer Sicherheitsfunktion kann für das Gesamtsystem entschieden werden, ob ein degradierter Betrieb einzelner Sicherheitsfunktionen unter Berücksichtigung des Restrisikos des Gesamtsystems möglich ist.

4 IT-Kommunikation in der Automatisierung

Zukünftige Automatisierungssysteme erfordern einen hohen Vernetzungsgrad, um die Anforderungen neuer Paradigmen wie Industrie 4.0 und 5.0 umzusetzen. Unabhängig vom Automatisierungsgrad können Komponenten miteinander kommunizieren [120], [121], [122]. Dies hat zur Folge, dass die Grenzen zwischen den klassischen Ebenen der Automatisierungspyramide zunehmend an Bedeutung verlieren und einfache, standardisierte Kommunikationsmethoden zwischen den verschiedenen Ebenen der Pyramide gefordert werden.

In diesem Kapitel werden bewährte Kommunikationstechnologien aus dem Bereich der IT vorgestellt. Es wird zudem diskutiert, wie diese in der Automatisierung eingesetzt werden können und wie eine effiziente Vernetzung der verschiedenen Automatisierungskomponenten realisiert werden kann. Darüber hinaus wird gezeigt, wie eine einfache Synchronisation von zwei oder mehr SPSen ohne großen Aufwand und zusätzliche Kosten realisiert werden kann. Abschließend wird eine sicherheitsbezogene Steuerungsarchitektur für mobile Assistenzsysteme gezeigt.

4.1 Publish-Subscribe Methoden

Betrachtet man die C2C-Kommunikation oberhalb einer Modulsteuerung in der Automatisierungspyramide, gibt es derzeit keine eindeutige Wahl bei der Vernetzungstechnologie. Hier werden auch heute noch die klassischen Ethernet-Feldbusse wie PROFINET und EtherNet/IP eingesetzt. Alternativ können Prozessdaten auch mit bewährten und standardisierten Methoden aus der IT übertragen werden. MQTT z. B. ist ein offenes Netzwerkprotokoll für die Machine-to-Machine (M2M)-Kommunikation. MQTT, entwickelt für IoT, basiert auf dem TCP/IP-Protokoll. Die Variante MQTT-SN ermöglicht weitere Übertragungswege wie dem User Datagram Protocol (UDP). Der MQTT-Server als Broker hält die gesamte Datenlage seiner Kommunikationspartner bereit und wird so zu einer globalen Zustandsdatenbank. Damit ist es möglich, auch kleine, weniger leistungsfähige MQTT-Clients ereignisgesteuert mit einem MQTT-Broker zu verbinden. Die Clients senden jeweils ihre Prozessdaten an den Broker und so entsteht ein vollständiges, konsistentes Prozessabbild auf dem MQTT-Broker. Der Broker kann ein Server in der Cloud sein, aber auch ein sogenannter Edge Controller in der Nähe der Maschine.

Um den Echtzeitbetrieb zu erleichtern, hat die OPC Foundation das PubSub-Modell in den OPC UA Standard Part 14 aufgenommen [123]. Dieses Modell unterstützt brokerbasierte Protokolle wie MQTT und Advanced Message Queuing Protocol (AMQP). Neben dem Datenaustausch über die genannten Protokolle, bietet OPC UA auch Dienste wie Datenmodellierung, Adressraum, Alarm- und Ereignismanagement und Zugriffskontrolle [124].

DDS, auf dem das ROS 2 Framework basiert, verfolgt ähnliche Ansätze [31]. Das von DDS verwendete Kommunikationsmodell ist ein unidirektionaler many-to-many-Datenaustausch. In eProsimas Open-Source-DDS-Implementierung [125], die von ROS 2 unterstützt wird, können die Kommunikationsmechanismen zwischen den Prozessen auch über Shared Memory genutzt werden. Dies ermöglicht einen sehr schnellen Datenaustausch zwischen verschiedenen Softwarekomponenten auf demselben Host unter Verwendung von DDS-Methoden ohne Netzwerkkommunikation. Für die Netzwerkkommunikation bietet eProsimas Fast DDS zusätzlich vier IP-basierte Transportoptionen: UDPv4, UDPv6, TCPv4, und TCPv6. Abbildung 31 zeigt eine mögliche Vernetzungsstruktur verschiedener Geräte mit DDS in der Automatisierung. Über den DDS-Datenbus lassen sich SPSen mit ihrer zyklischen Kommunikation aus der Automatisierungstechnik ganz einfach mit einem auf ROS 2 basierenden Roboter verbinden. Nicht-Echtzeit-Kamerasysteme und Systeme mit künstlicher Intelligenz (KI) können über DDS einfach integriert werden und nutzen die Vorteile des ereignisgesteuerten PubSub-Ansatzes.

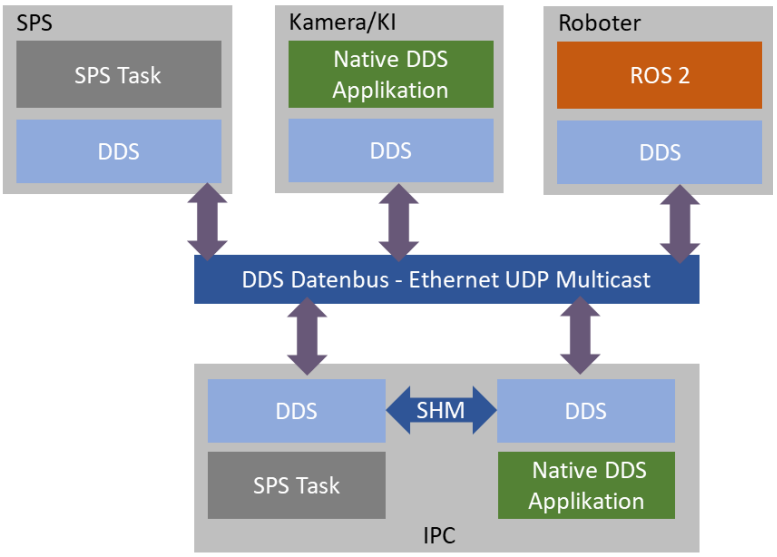


Abbildung 31: Vernetzungsstruktur mit DDS in der Automatisierung.

4.2 Modulsteuerung-Kommunikation

Der Einsatz des PubSub-Modells unterstützt eine effiziente Vernetzung von nicht zyklischen Ereignissen, wie z. B. die Datenübertragung von Bildverarbeitungssystemen oder KI mit mehrachsigen Motion-Control-Systemen wie Robotern, Cobots und FTFs. Roboterkomponenten mit ROS 2 bieten keine bewährten Mechanismen zur taktsynchronen Signalverarbeitung für eine zyklische Abtaststeuerung. Gängige Ansätze zur Synchronisation mit ROS 2 basieren auf dem Network Time Protocol (NTP), Precision Time Protocol (PTP) oder TSN [126], [127]. Aus Komplexitäts- und Kostengründen ist es jedoch sinnvoller, ROS 2 mit bewährter SPS-Technik und synchronen Feldbussen aus der Automatisierungstechnik zu kombinieren. Synchronisationsmechanismen wie NTP erfüllen nicht die hohen Anforderungen und PTP und TSN erfordern kompatible Netzwerkkomponenten (Switches und Router) und komplexe Softwarestacks. Würde man hier statt der bewährten Motion-Control-Feldbusse mit Methoden zur synchronen Abtastung (z. B. Distributed-Clock bei EtherCAT) durch GbE mit PTP/TSN etc. ersetzen, erhöht dies die Kosten ohne zusätzlichen Nutzen für die Anwendung.

Abbildung 32 zeigt die Kommunikation aus Sicht einer Modulsteuerung, die z. B. einen Roboter steuert. Die Modulsteuerung ist hier die Grenze zwischen IT und OT [33]. Von der Modulsteuerung bis hinunter zur Feldebene ist es sinnvoll, klassische OT-Methoden mit zyklischer Abtastung zu verwenden. Speziell für Motion-Control-Systeme ist EtherCAT eine weit verbreitete, effiziente und kostengünstige Lösung zur Vernetzung von Antrieben. Die Distributed-Clock-Funktionalität von EtherCAT erlaubt eine vollständig synchrone Abtastung und Aktualisierung der Peripherie [26]. Bei diesem Ansatz sind die in der Automatisierungstechnik üblichen IEC 61131-3- Programmierverfahren nach wie vor die bevorzugte Lösung. Oberhalb einer Modulsteuerung harmonieren OPC UA PubSub, MQTT und DDS mit dem ereignisgesteuerten Ansatz der SPS-Norm IEC 61499 [32]. Gerade in komplexen Projekten mit einer großen Anzahl vernetzter Teilnehmer ist es hilfreich, dass durch die ereignisgesteuerte Kommunikation in der Regel deutlich weniger Prozessdaten pro Sekunde gesendet werden. Ein weiterer Vorteil ist das einfache Hinzufügen und Entfernen von Geräten auch während des Betriebs.

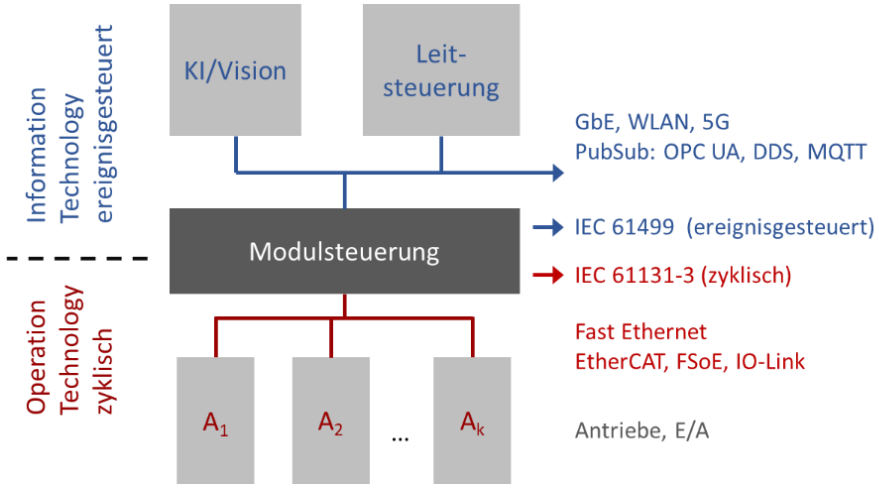


Abbildung 32: Eine Modulsteuerung bildet die Grenze zwischen IT und OT.

Eine sinnvolle Vernetzung einer Modulsteuerung nutzt DDS, da es als Middleware für ROS 2 die Integration von ROS 2 Komponenten und Robotern in die IEC 61131-3 Welt ermöglicht. IPC-basierte Steuerungen bieten in der Regel keine Onboard-E/As. IO-Link als Teil der Antriebssteuerungshardware eignet sich für die kostenoptimierte Anbindung von E/A-Geräten. Werden die 24 V Digital-E/As eines Antriebs mit IO-Link-Transceivern realisiert, kann die Modulsteuerung die ungenutzten Antriebs-E/As frei konfigurieren.

4.3 Synchronisation von Prozessdaten zwischen zwei Steuerungen

Im Folgenden wird ein Praxisbeispiel beschrieben, um das Synchronisationskonzept vorzustellen. Eine Modulsteuerung steuert ein Förderband und verfolgt mit geeigneter Sensorik Objekte, die darauf platziert werden. Eine weitere Modulsteuerung steuert einen Roboter, der die verfolgten Objekte vom laufenden Förderband aufnimmt und in Kisten ablegt. Die Herausforderung hierbei besteht in der Synchronisation der beiden zyklisch arbeitenden SPSen. Die Bewegungsposition des Förderbandes bzw. des verfolgten Objekts wird von einer Steuerung an die andere Steuerung übertragen. Zunächst wird eine einzelne Compound-SPS betrachtet und in Abbildung 33 dargestellt. Eine Modulsteuerung, die auf einer Multi-Core CPU basiert, hat ihre eigene quarzbasierte Takterzeugung mit individuellen Toleranzen. PubSub-Methoden wie OPC UA PubSub und DDS bieten keine Methoden zur synchronen Abtastung der Peripherie, weshalb EtherCAT mit der Distributed-Clock-Funktionalität hier für die Vernetzung der Antriebe eingesetzt wird. Die Motion-Control-Anwendung und der EtherCAT-Master arbeiten synchron mit der gleichen Zykluszeit wie die SPS-Task (1 ms). Die internen Regelkreise der angeschlossenen Antriebe

arbeiten synchron, mit deutlich kürzeren Zykluszeiten als die übergeordnete SPS, typischerweise 16-mal schneller (62,5 µs). Eine gängige C2C-Kommunikation erfordert keine derart schnellen Zykluszeiten, hier sind oft 10 ms ausreichend, um die Prozessdaten, z. B. die Position des Förderbandes, zu übertragen. Allerdings müssen Jitter und Latenz der C2C-Kommunikation berücksichtigt werden.

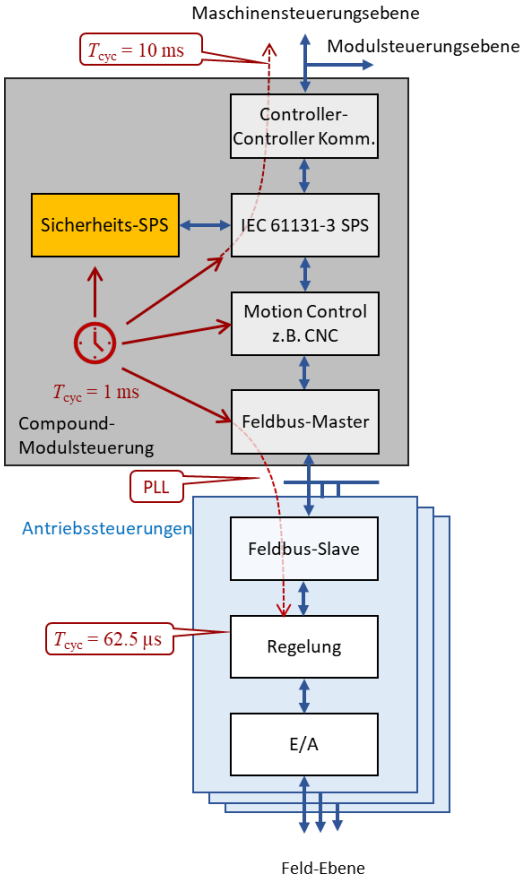


Abbildung 33: Synchronisation einer Compound-SPS.

Die Abbildung 34 zeigt die oben beschriebene Beispielanwendung. Während die Antriebe des Förderbands und des Roboters von der SPS über einen Feldbus (EtherCAT) synchron angesteuert werden, werden für die C2C-Kommunikation oberhalb der SPS PubSub-Methoden aus der IT wie DDS eingesetzt. Hier bildet die Modulsteuerung die Grenze zwischen OT und IT. DDS sowie die SPS-Anwendung sind zwei getrennte Prozesse auf dem IPC. Für den Datenaustausch zwischen beiden Prozessen, wird eine Interprozess-Kommunikation implementiert, die aus Performancegründen mit Shared Memory realisiert wird.

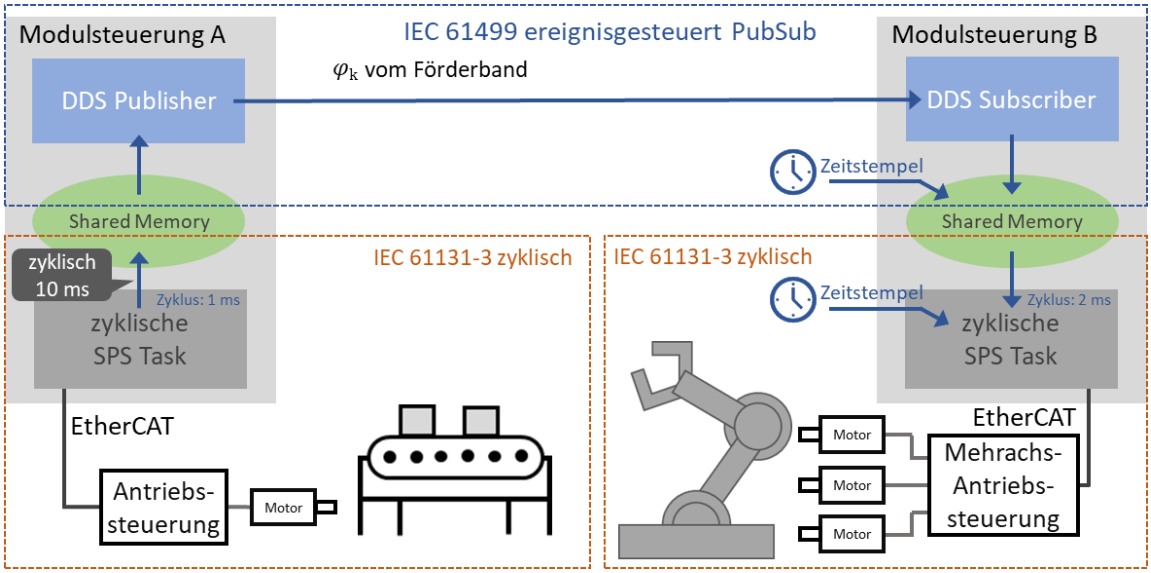
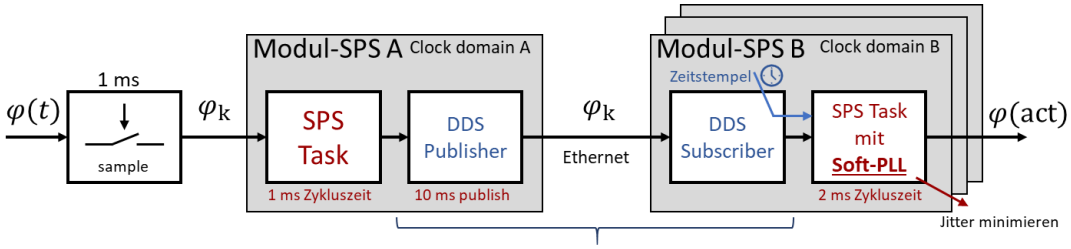


Abbildung 34: Beispielanwendung für die Controller-to-Controller Kommunikation in der Automatisierung.

Die zeitkritische Übertragung von Prozessdaten, sowie die Synchronisation von Steuerungen ist in der Abbildung 35 detailliert dargestellt. Da Synchronisationsmethoden wie TSN und PTP für viele industrielle Anwendungen zu teuer und zu komplex sind, basiert das vorgestellte Synchronisationskonzept auf einer zyklischen Kommunikation mit Zeitstempel und einer Software-PLL (Soft-PLL).

Zunächst wird der Positionswert $\varphi(t)$ innerhalb der Taktdomäne A zyklisch von einem Drehgeber (1 ms) abgetastet und vom Antrieb an die übergeordnete SPS gesendet. Der abgetastete Wert φ_k wird mit einer langsameren Zykluszeit (10 ms) über eine Shared Memory Schnittstelle an einen DDS-Publisher auf demselben Host übertragen. Der DDS-Publisher veröffentlicht den Positionswert mit einer UDP/IP Nachricht über den DDS-Datenbus. Der DDS-Teilnehmer in der Modulsteuerung B hat die Daten von der Modulsteuerung A abonniert. Beim Empfang der Nachricht fügt der DDS-Subscriber den Prozessdaten einen Zeitstempel hinzu und stellt diese dem zyklischen SPS-Prozess innerhalb der Clock-Domäne B bereit. In dieser SPS-Anwendung ist eine Soft-PLL implementiert, die in der IEC 61131-3 Programmiersprache ST programmiert ist, um den Jitter durch die nicht-harte Echtzeitübertragung zu minimieren. Neben dem Jitter muss für eine synchronisierte Kommunikation auch die durchschnittliche Latenzzeit berücksichtigt werden. Bei einer geeigneten Netzwerkkonfiguration mit geringer Buslast und einer minimalen Anzahl von Switches zwischen den Controllern kann jedoch davon ausgegangen werden, dass die Übertragungszeit eine konstante Latenzzeit zuzüglich Jitter beträgt.



$$T_{\text{DDS delay}} = T_{\text{Pub}} + T_{\text{UDP}} + T_{\text{Switch}} + \ldots + T_{\text{Sub}} \approx \text{konst.}$$

Abbildung 35: Blockschaltbild für die Übertragung von Prozessdaten zwischen Modulsteuerungen.

Die Abbildung 36 zeigt das Synchronisationskonzept für die soft-synchronisierte Echtzeitkommunikation zwischen zwei Steuerungen mit unabhängigen Taktdomänen. Es ist zu erkennen, dass die beiden Steuerungen unterschiedliche Zykluszeiten haben können. Die Zeit $T_{\text{DDS delay}}$ ist die Latenz, die durch die Übertragung der Prozessdaten über DDS entsteht. Diese kann beispielsweise durch hin- und zurücksenden der Daten auf einfache Art ermittelt werden. Das gemessene Ergebnis kann dann aufgrund der Symmetrie durch zwei geteilt werden. Wenn die Prozessdaten vom DDS-Teilnehmer empfangen werden, wird den Prozessdaten ein hochpräziser Zeitstempel ($t_{\text{DDS Sub}}$) hinzugefügt. Da die SPS-Task zyklisch ausgeführt wird, sind die verschiedenen empfangenen Daten unterschiedlich alt. Daher wird in der SPS-Task ebenfalls ein Zeitstempel (t_{SPS}) gesetzt. Die Differenz der beiden Zeitstempel, also das Alter des Prozessdatums, ist T_{Sync}. Da in der Empfänger-SPS mehrere Taktzyklen für jedes publizierte Prozessdatum durchlaufen werden, muss in jedem Zyklus die Zeit t_{SPS} aktualisiert werden. Dadurch wird auch das Alter des Prozessdatums (T_{Sync}) aktualisiert. Die Gesamtverzögerungszeit setzt sich somit wie folgt zusammen:

$$T_{\text{Sync}} = t_{\text{SPS}} - t_{\text{DDS Sub}} \tag{26}$$

$$T_{\text{RX delay}} = T_{\text{DDS delay}} + T_{\text{Sync}} \tag{27}$$

Die Verzögerungszeit $T_{\text{RX delay}}$ wird entsprechend verwendet, um den Winkel für die Positionieraufgaben in Modulsteuerung B zu schätzen:

$$\varphi_{\text{k}_{\text{sub}}} = \varphi_{\text{k}_{\text{pub}}} + \omega_{\text{k}}\, T_{\text{RX delay}} \tag{28}$$

mit

$$\omega_{\text{k}} = \frac{\Delta\varphi_{\text{k}_{\text{pub}}}}{T_{\text{cyc\_publish}}} \tag{29}$$

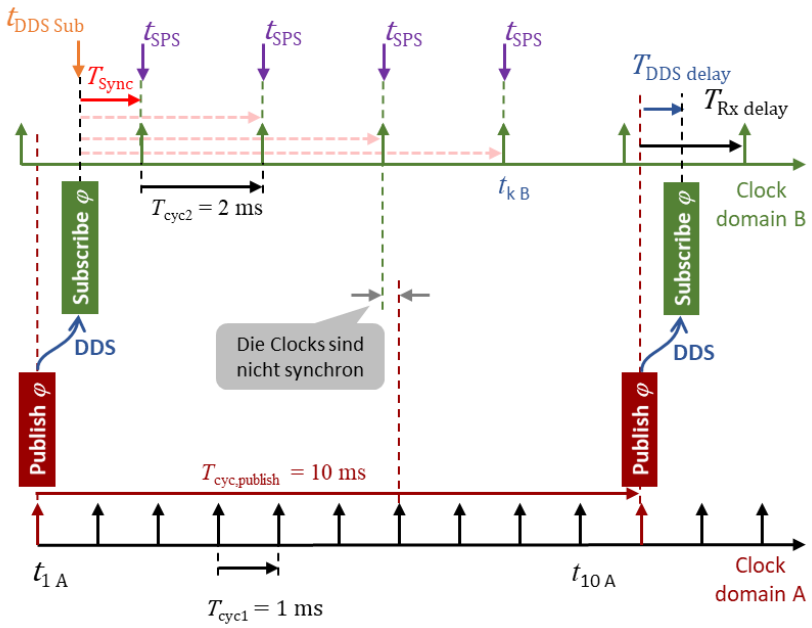


Abbildung 36: Zeitverhalten und Synchronisation für eine DDS-basierte Kommunikation zwischen zwei Modulsteuerungen.

Durch die vorgestellte Methode zur Synchronisation von zwei oder mehreren Steuerungen mit der PubSub-Kommunikation aus der IT ist es in der Regel nicht mehr notwendig einen zyklischen Feldbus für die C2C-Kommunikation zu nutzen. Damit ist es möglich die vorgestellten Vorteile der PubSub-Kommunikation, wie z. B. das Hinzufügen und Entfernen von Teilnehmern im laufenden Betrieb, für die Automatisierung zu nutzen. Außerdem sind bei der vorgestellten Anwendung keine komplexen und teuren Synchronisationsmethoden wie TSN oder PTP erforderlich, um die Steuerungen zu synchronisieren.

4.4 Sicherheitsbezogene IT-Kommunikation

Für die Kommunikation zwischen Steuerungen wird ebenfalls eine sicherheitsbezogene Datenübertragung gefordert. Insbesondere im Kontext von Industrie 4.0 und 5.0 wird die Kommunikation zwischen Steuerungen verschiedener Hersteller immer wichtiger, auch wenn es hier niedrigere Anforderungen an die funktionale Sicherheit gibt und Signale in der Regel mit höheren Zykluszeiten verarbeitet werden.

Mit OPC UA Safety wird eine sicherheitsbezogene Kommunikation nun auch zwischen Controllern ermöglicht [128]. Es handelt sich um eine gemeinsam von der Interessengruppe PROFIBUS & PROFINET International (PI) und der OPC Foundation definierte Spezifikation, die auf dem OPC Standard für die M2M-Kommunikation aufsetzt. OPC

UA Safety kann sowohl in der Client-Server- als auch als PubSub-Architektur implementiert werden. Es sind einige Features vorhanden, die bei den gängigen sicherheitsbezogenen Protokollen in der OT nicht vorgesehen sind. Dazu gehört unter anderem ein dynamischen Verbindungsaufbau mit wechselnden Partnern, was insbesondere für mobile Roboter wie FTFs, AMRs von Bedeutung ist. Hierbei ist es nicht mehr notwendig alle Teilnehmer bereits in der Projektierungsphase miteinander bekannt zu machen. Mit OPC UA Safety ist es möglich einen neuen mobilen Roboter in das System aufzunehmen, ohne alle stationären Maschinen neu zu parametrieren.

4.5 Sicherheitsbezogene Steuerungsarchitektur für mobile Robotersysteme

Für die lokale Anbindung von Sensoren und Aktoren ist es derzeit nicht wirtschaftlich OT-Feldbusse durch die IT-Kommunikation zu ersetzen. Insbesondere EtherCAT, erweitert um das sicherheitsbezogene Protokoll FSoE, ist weit verbreitet und bietet eine hochgenaue Synchronisation der angebundenen Peripherie mit der übergeordneten SPS. Mit schnellen Feldbussen und den damit verbundenen Sicherheitsprotokollen in Verbindung mit schnellen Sicherheits-SPSen mit Gleitkomma-Arithmetik können sicherheitsbezogene Antriebssteuerungen weniger komplex gestaltet werden, ohne auf Rechenleistung zu verzichten. Mit der heutigen verfügbaren hohen Rechenleistung für Sicherheits-SPSen lassen sich auch komplexe Kinematiken sicherheitsbezogen überwachen. Durch die integrierte zentrale Sicherheits-SPS können Hardwarekomponenten durch Softwaremodule ersetzt werden.

Ein externes Bildverarbeitungssystem mit KI kann über PubSub-Methoden wie DDS optional implementiert werden. Ebenso ist es möglich ROS 2 kompatible Geräte über DDS einzubinden. Der Verzicht auf die Integration der KI-Verarbeitung direkt in den IPC ist vorteilhaft, da spezielle KI-Hardware in der Regel kürzere Innovationszyklen als Automatisierungskomponenten aufweist. Demzufolge kann die KI-Hardware veraltet sein, während die Automatisierungshardware noch aktuell ist.

Für eine übergeordnete Leitsteuerung eignet sich ein leistungsfähiger IPC. Da auf dieser Ebene im Bereich der funktionalen Sicherheit in der Regel nur digitale Signale mit vergleichsweise hohen Zykluszeiten verarbeitet werden, sind die Leistungsanforderungen an die Sicherheits-SPS im Vergleich zu einer Modulsteuerung mit Gleitkomma-Arithmetik geringer. Mit Coded Processing kann Standard-IPC-Hardware oder eine virtuelle SPS auch für funktional sichere Anwendungen verwendetet werden. Die geringere Performance und Energieeffizienz von Coded Processing sind bei einer Leitsteuerung mit hoher

IPC-Rechenleistung nicht von großer Bedeutung, da in der Regel nur ein kleiner Teil der Rechenleistung sicherheitsrelevant ist. Die drahtlose Kommunikation zwischen den mobilen Robotersystemen und der Leitsteuerung basiert auf WLAN oder zukünftig auf 5G. Mit dem in der VDA5050 vorgeschlagenen MQTT-Protokoll und der OPC UA Safety Erweiterung lässt sich eine drahtlose und effiziente PubSub-Architektur für die Vernetzung von FTFs und AMRs realisieren. Abbildung 37 zeigt gemäß [61] eine geeignete sicherheitsbezogene Steuerungsarchitektur für autonome mobile Roboter.

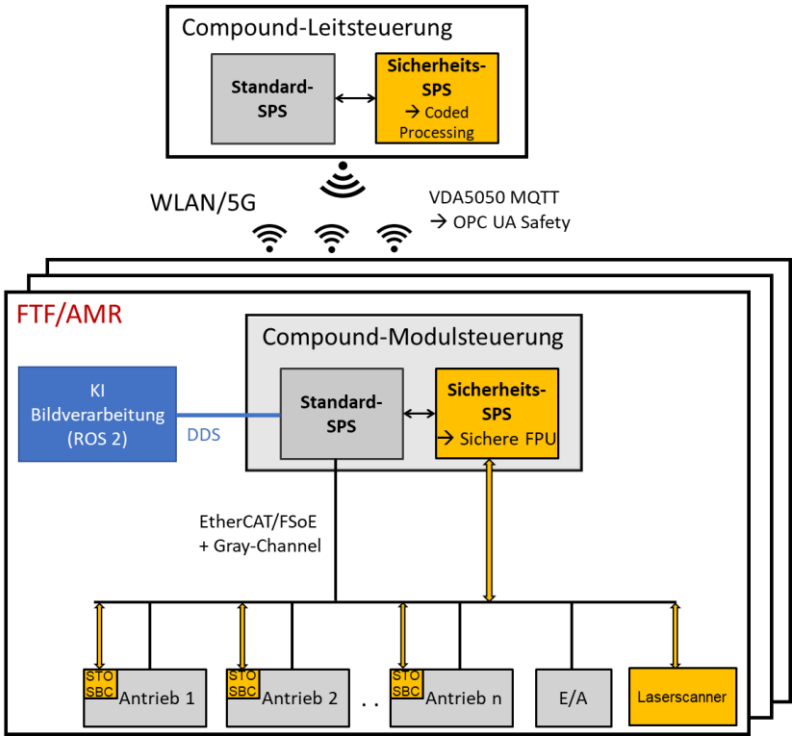


Abbildung 37: Sicherheitsbezogene Steuerungsarchitektur für autonome mobile Roboter.

5 Systemarchitektur für gemischt-kritische Steuerungen

In diesem Kapitel wird eine Systemarchitektur einer Compound-SPS für gemischt-kritische Anwendungen vorgestellt. Die gemischt-kritische Compound-SPS soll die vorgestellten Ansätze der Sicherheitslösung mit zentraler Logikverarbeitung und Diagnose sowie die IT- und OT-Kommunikationsmethoden unterstützen. Die integrierte zentrale Sicherheits-SPS ermöglicht mit einer hohen Rechenleistung und Gleitkomma-Arithmetik die Berechnung von komplexen Sicherheitsfunktionen. Diese Sicherheitsfunktionen erlauben eine sichere Kollaboration zwischen Menschen und Maschinen. Virtualisierungstechnologien ermöglichen eine flexible und skalierbare Automatisierungslösung.

Zunächst werden die Anforderungen an den funktional sicheren Teil der Steuerung genannt. Es werden Lösungsansätze diskutiert, wie die Rückwirkungsfreiheit zwischen dem sicherheitsbezogenen und dem nicht sicherheitsbezogenen Teil realisiert werden kann. Anschließend wird die Softwarearchitektur beschrieben.

5.1 Multi-Core System-on-a-Chip

Die Systemarchitektur der Compound-SPS für gemischt-kritische Anwendungen in der Fertigungs- und Maschinenautomatisierung basiert auf einem nicht sicherheitsbezogenen Teil für die Standardanwendung und einem sicherheitsbezogenen Teil für die Sicherheitsanwendung. Der hier vorgestellte Ansatz der Systemarchitektur verwendet einen Multi-Core SoC. Um die Redundanz zu gewährleisten, wird die Sicherheitsanwendung auf zwei Prozessorkernen ausgeführt. Da beide Kanäle für die Sicherheitsfunktionen auf demselben Prozessor implementiert werden, ist für diesen Ansatz eine externe TE notwendig, die als unabhängiger Silizium-Die in der Lage ist, das System in einen sicheren Zustand zu versetzen [37]. Eine Sicherheitsarchitektur mit einem Multi-Core Prozessor lässt sich ebenso wenig wie die Architektur des Lockstep Prozessors direkt auf eine vorgesehene Architektur der ISO 13849-1 abbilden. Demzufolge sind zusätzliche Maßnahmen zur Fehlererkennung (DC_{avg} von hoch) erforderlich, um eine Kategorie 3, PL d äquivalente Risikoreduzierung zu erreichen. Diese zusätzlichen Maßnahmen in Form von Selbsttests und einem Kreuzvergleich sind bei der Systemarchitektur der Modulsteuerung zu berücksichtigen.

5.2 Architekturansätze

Eine wichtige Anforderung bei der Entwicklung sicherheitsbezogener Steuerungen ist die Rückwirkungsfreiheit. Nicht sicherheitsbezogene Anwendungen dürfen sicherheitsbezogene Anwendungen nicht beeinflussen [37], [91]. Für die Zertifizierung im Maschinensektor bis SIL2 bzw. PL d und Kategorie 3 basiert der sicherheitsbezogene Teil in der Regel auf einer zweikanaligen Architektur. Im Folgenden werden verschiedene Ansätze zur Realisierung der geforderten Redundanz sowie der Rückwirkungsfreiheit zwischen sicherheitsbezogenen und nicht sicherheitsbezogenen Komponenten (Hardware und Software) diskutiert.

5.2.1 Prozess-Isolation mit Echtzeitbetriebssystem

Ein sicherheitsbezogenes RTOS, welches verschiedene Anwendungen in zugewiesenen Prozessen ausführt, unterstützt in der Regel eine zertifizierte Prozess-Isolation auf Basis einer Memory Management Unit (MMU) oder einer Memory Protection Unit (MPU). Die MMU ist eine Hardwarekomponente, die in moderne Prozessoren integriert ist und zur Speicherverwaltung eingesetzt wird. Sie bietet eine Virtualisierungsschicht, die es dem Betriebssystem ermöglicht, für jeden Prozess einen virtuellen Adressraum zu erstellen [129]. Auf diese Weise kann jeder Prozess auf eine isolierte Speicherregion zugreifen, die ihm vom Betriebssystem zugewiesen wird. Die MPU ist eine reduzierte Version der MMU, die den Speicherschutz ohne die Virtualisierung von Adressen unterstützt. Sie wird in der Regel bei Prozessoren mit geringer Leistungsaufnahme eingesetzt, die nicht den vollen Funktionsumfang der MMU benötigen. Die MPU ermöglicht die Definition von Speicherbereichen und die Zuweisung von Speicherzugriffsrechten und -attributen. Bei einer Speicherzugriffsverletzung löst die MPU eine Fehlerausnahme aus.

Da die Sicherheitsarchitektur äquivalent zu Kategorie 3 mit zwei unabhängigen Kanälen realisiert werden soll, werden die Sicherheitsfunktionen bei diesem Ansatz gleichzeitig in zwei Prozessen ausgeführt, die je einem physischen Prozessorkern zugewiesen sind. Eine nicht sicherheitsbezogene SPS-Anwendung kann dabei in einem eigenen Echtzeit-Prozess ausgeführt und vom sicherheitsbezogenen RTOS auf verfügbare Prozessorkerne verteilt werden. Eine mögliche Systemarchitektur ist in der Abbildung 38 zu sehen.

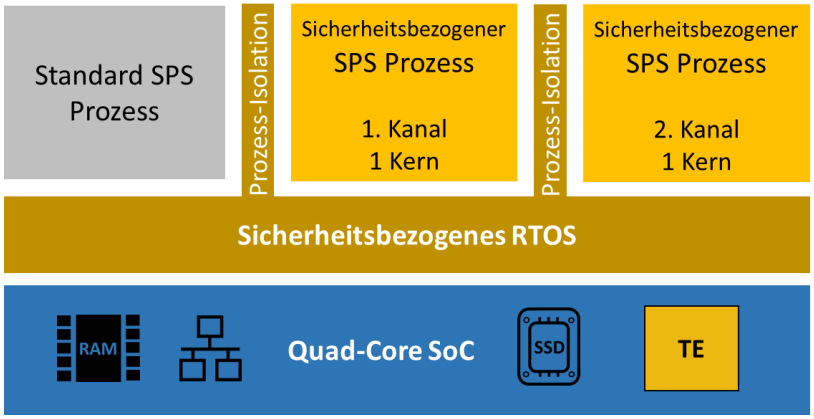


Abbildung 38: Systemarchitektur mit sicherheitsbezogenem RTOS und Prozess-Isolation.

Dieser Ansatz erfüllt die Sicherheitsanforderungen, schränkt aber viele zusätzliche Möglichkeiten ein. Die Implementierung von SPS- und IPC-Innovationen wie Cloud Computing, ROS 2, HMI- und IoT-Anwendungen ist nur eingeschränkt möglich, da das sicherheitsbezogene Betriebssystem die meisten Funktionen blockiert oder zusätzliche Funktionen zertifiziert werden müssen. Automatische Software-Updates sind bei diesem Ansatz nicht ohne weiteres möglich, da nach jedem Update eine erneute Zertifizierung erfolgen muss. Auch die Verwendung von nicht zertifizierten Gerätetreibern ist nicht möglich. Da ein sicherheitsbezogenes Betriebssystem zertifiziert werden muss, werden neue Versionen seltener angeboten. Für die Automatisierung ist dieser Ansatz weniger attraktiv, da die Innovationen von IPC-Systemen kaum genutzt werden können.

5.2.2 VM-Isolation mit Hypervisor

Der zweite Ansatz basiert auf einem zertifizierten, sicherheitsbezogenen Hypervisor, kann jedoch auch auf Betriebssysteme mit Mikrokernel und Virtualisierungsfunktionalität (Separationskernel) übertragen werden. Bei diesem Ansatz können Anwendungen unterschiedlicher Kritikalität in verschiedenen VMs implementiert werden. Sobald Anwendungen mit unterschiedlicher Kritikalität auf gemeinsame Ressourcen zugreifen, muss der Einfluss auf sicherheitsbezogene Anwendungen sicherheitstechnisch bewertet werden. Eine Möglichkeit, die geforderte Rückwirkungsfreiheit zu realisieren, ist die feste Zuweisung von Ressourcen an die entsprechenden VMs. Dazu gehört unter anderem eine feste Zuweisung von CPU-Kernen, Speicher und Gerätetreibern. Ein Hypervisor mit der Partition Mode Konfiguration und Device Passthrough erfüllt die hohen Anforderungen der Isolation. VMs haben in dieser Konfiguration keine Kenntnis von der Existenz anderer VMs und ein Ausfall oder ein Neustarten einer VM hat keinen Einfluss auf andere VMs in dem System. Darüber hinaus bieten Hypervisoren in der Regel verschiedene Inter-VM-

Kommunikationsmethoden an, damit die VMs miteinander kommunizieren können. Virtuelle Netzwerke, vUARTs und geteilte Speicherbereiche wie Shared Memory sind einige Beispiele, wobei sich letzteres insbesondere aus Performancegründen anbietet [97], [98].

Eine nicht sicherheitsbezogene VM hostet bei diesem Ansatz ein RTOS und wird unabhängig vom sicherheitsbezogenen Teil ausgeführt. Bei der Realisierung des sicherheitsbezogenen Teils und der zweikanaligen Architektur gibt es mehrere Konfigurationsmöglichkeiten. Bei der ersten Konfiguration werden zwei physische Prozessorkerne, die für die zweikanalige Verarbeitung notwendig sind, fest einer sicherheitsbezogenen VM zugeordnet. Innerhalb des sicherheitsbezogenen Betriebssystems führen zwei Prozesse parallel das gleiche Sicherheitsprogramm aus, jeweils auf einem eigenen physischen Kern. Der sicherheitsbezogene Hypervisor weist den Gerätetreiber der externen TE der sicherheitsbezogenen VM zu und stellt damit sicher, dass nicht aus der Standard VM auf ihn zugegriffen werden kann. Abbildung 39 zeigt diese Systemarchitektur.

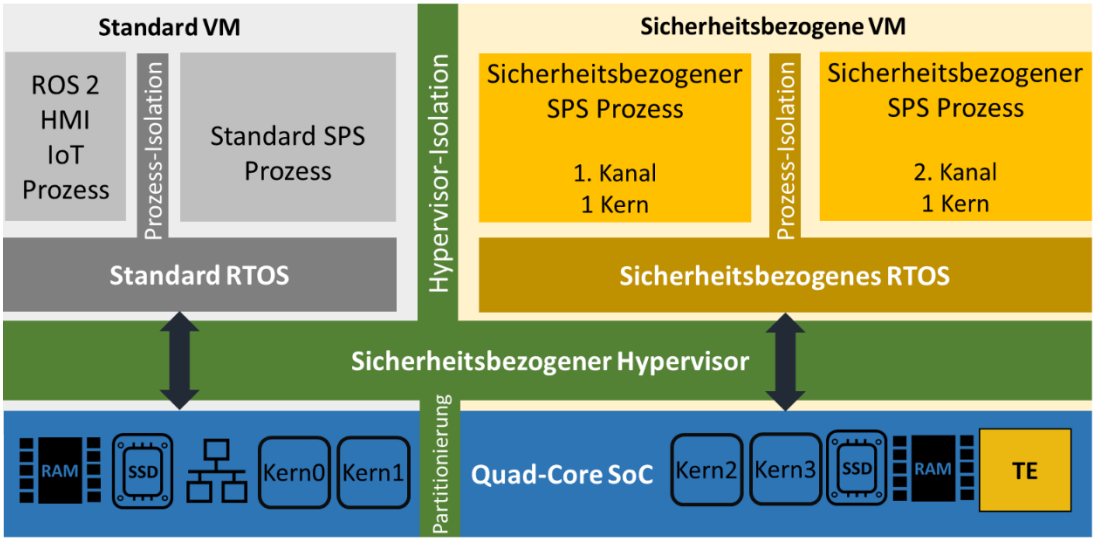


Abbildung 39: Systemarchitektur mit Hypervisor und zwei VMs unterschiedlicher Kritikalität.

Mit zunehmender Komplexität des Betriebssystems steigt der Zertifizierungsaufwand überproportional an. Daher kann ein sicherheitsbezogenes Betriebssystem sehr eingeschränkt sein und eine erweiterbare Bare-Metal-Laufzeitumgebung darstellen. Einige sicherheitsbezogene Betriebssysteme unterstützen z. B. kein Multitasking mit virtuellen Adressen, d. h. sie erlauben nicht die Ausführung paralleler Prozesse mit Speicherschutz.

Eine weitere Konfiguration mit Redundanz im sicherheitsbezogenen Teil basiert auf der Implementierung von zwei getrennten sicherheitsbezogenen VMs mit jeweils einem eigenem physischen Prozessorkern, eigenem Arbeitsspeicher (RAM) und eigenem nichtflüchtigen Speicher. Diese Konfiguration ist in der Abbildung 40 zu sehen. Dieser Ansatz mit

einem Hypervisor im Partition Mode hat die Einschränkung, dass die externe TE nur einer sicherheitsbezogenen VM zugewiesen werden kann. Zusätzlicher Aufwand ist erforderlich, damit beide sicherheitsbezogenen VMs die Ergebnisse ihrer Diagnosemaßnahmen an die TE zur Auswertung übertragen können.

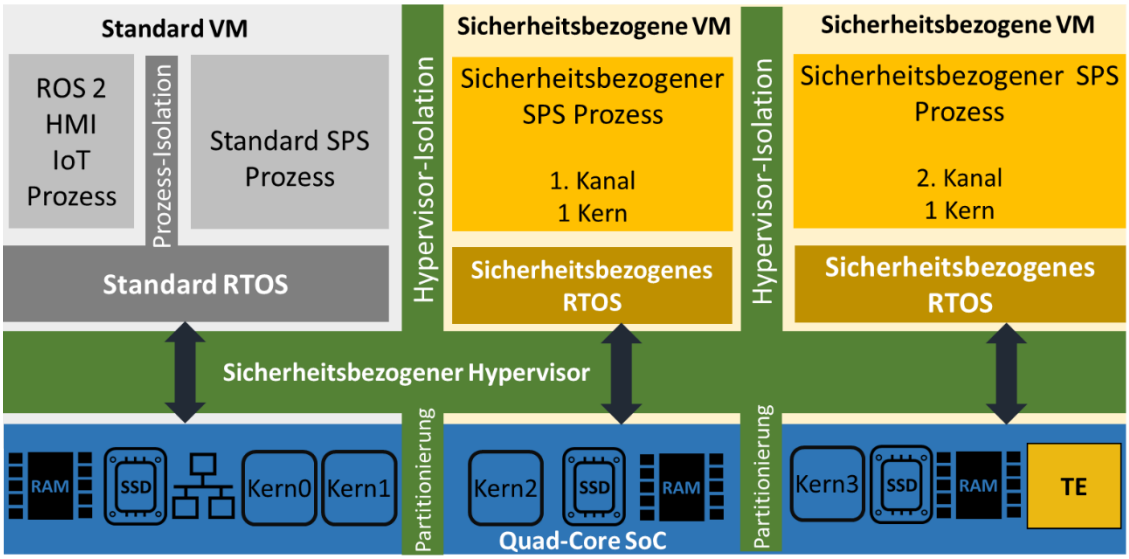


Abbildung 40: Systemarchitektur mit Hypervisor und zwei VMs für den sicherheitsbezogenen Teil.

Die bisher vorgestellten Ansätze basieren auf einem zertifizierten Betriebssystem für den sicherheitsbezogenen Teil, was jedoch mit erheblichen Kosten verbunden ist. SRP/CS werden häufig auch mit Standardkomponenten für den industrielle Anwendungen realisiert, insbesondere wenn diversitäre Redundanz angewendet wird. In [37] und [90] werden Anwendungsfälle beschrieben, bei denen sich diversitäre Redundanz mit unterschiedlicher Embedded-Software, z. B. mit zwei unterschiedlichen Betriebssystemen umsetzen lässt. Auf einen zertifizierten Hypervisor kann nicht verzichtet werden, da der Hypervisor einkanalig ist und somit nicht redundant implementiert werden kann. Die Architektur mit zwei diversitären Betriebssystemen ist in der Abbildung 41 dargestellt.

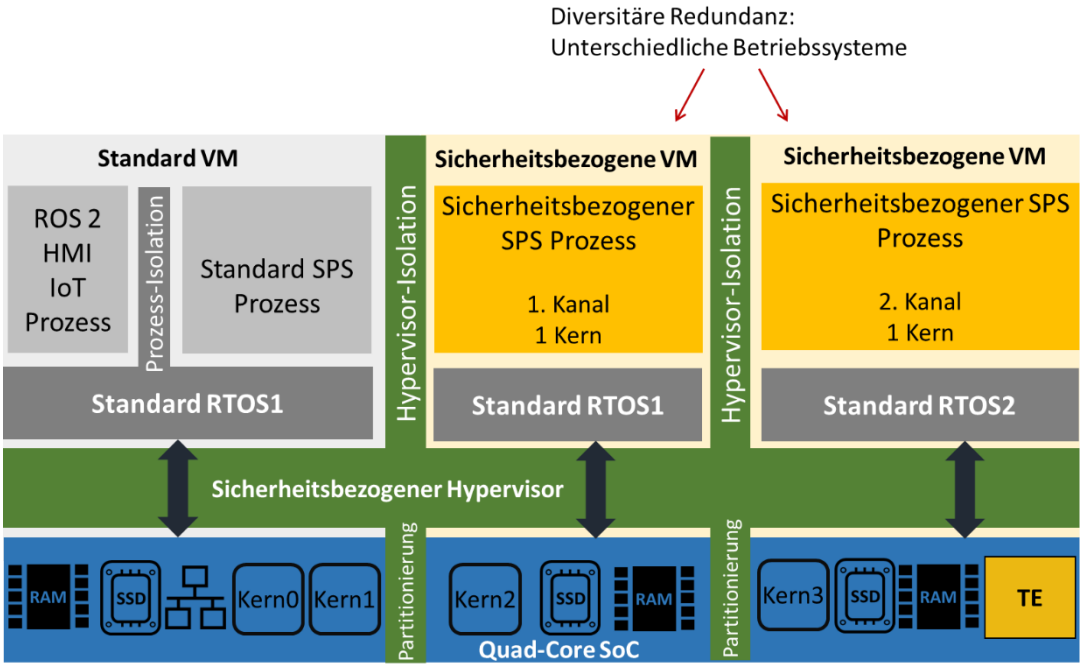


Abbildung 41: Systemarchitektur mit diversitären Betriebssystemen für den sicherheitsbezogenen Teil.

Der Einsatz eines Hypervisors hat den Vorteil, dass verfügbare SPS- und IPC-basierte Erweiterungen und Innovationen nicht durch ein sicherheitsbezogenes Betriebssystem beschränkt werden. Dieser auf einem Hypervisor (oder Mikrokernel) basierende Ansatz ermöglicht nicht nur ein hohes Maß an Sicherheit, sondern kann auch die Zertifizierung erheblich vereinfachen und damit Kosten senken. Insbesondere in den Bereichen funktionale Sicherheit und Security besteht die Gefahr, dass durch die Veränderung bestimmter Systemeigenschaften die Sicherheitsmaßnahmen aufgehoben werden und das System unsicher wird. In der Regel muss in einer solchen Situation das gesamte Gerät sicherheitstechnisch neu bewertet werden, einschließlich der Komponenten, die sich nicht geändert haben. Die Partitionierung der Gerätesoftware unter Verwendung eines zertifizierten Hypervisors oder Mikrokernels ermöglicht die Neuzertifizierung nur der von der Aktualisierung betroffenen VM. Diese strikte Partitionierung gewährleistet, dass die Sicherheitsvoraussetzungen der unveränderten VMs bestehen bleiben.

Dieser vorgestellte Ansatz benötigt mehr Arbeitsspeicher als die oben vorgestellten Ansätze, da mehrere VMs gehostet werden. Außerdem wird die Performance der sicherheitsbezogenen Kerne möglicherweise nicht voll ausgenutzt. Im Vergleich zum Sicherheits-RTOS mit der Prozess-Isolation ist dieser Ansatz bezüglich der Standardapplikation wesentlich flexibler und daher für industrielle Anwendungen geeignet. Die Containertechnologie wird bisher nicht zur Isolation im funktional sicheren Kontext eingesetzt, kann aber

zur einfachen Implementierung von Anwendungen auf dem nicht sicherheitsrelevanten Standardteil verwendet werden.

5.2.3 VM- und Prozess-Isolation mit Hypervisor

Der dritte Ansatz ist eine Kombination der beiden oben beschriebenen Ansätze. Ein zertifizierter Hypervisor hostet ein zertifiziertes RTOS für die zweikanalige sichere Logik und zusätzliche Prozesse für die Standardanwendung. Beispielsweise können Teile des Standard-SPS-Programms in zusätzlichen Prozessen im sicherheitsbezogenen Teil ausgeführt und vom Sicherheits-RTOS verwaltet und isoliert werden. Die Architektur dieses Ansatzes ist in der Abbildung 42 zu sehen.

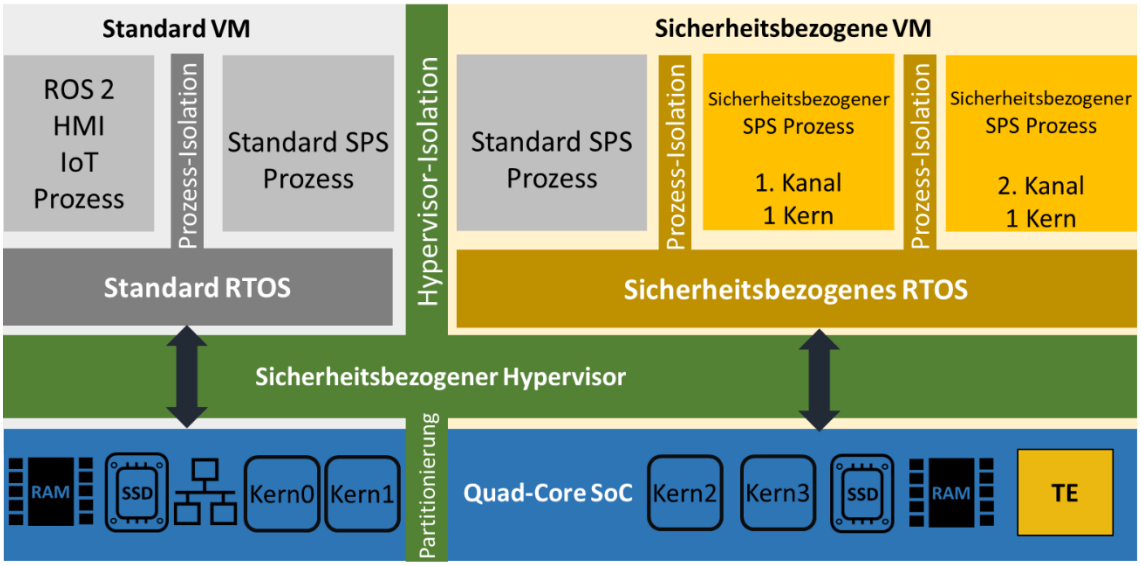


Abbildung 42: Systemarchitektur mit nicht sicherheitsbezogenen Arbeitslasten im sicherheitsbezogenen Teil.

Innerhalb der nicht sicherheitsbezogenen VM gibt es keine Einschränkungen, so dass HMI-, IoT-, ROS 2- und PubSub-Prozesse implementiert werden können. Nahezu alle SPS- und IPC-basierten Erweiterungen und Innovationen mit regelmäßigen Software-Updates sind ohne weitere Zertifizierung verfügbar. Dies macht den Ansatz auch für den industriellen Einsatz attraktiv. Dieser Ansatz eignet sich besonders für leistungsfähige Multi-Core Prozessoren, da die ungenutzte Rechenleistung von zwei schnellen Kernen für die Sicherheitsanwendung für andere, nicht sicherheitsbezogene Aufgaben zur Verfügung steht.

Der vorgestellte Ansatz basiert erneut auf einem sicherheitsbezogenen Betriebssystem, das aus Kosten- und Zertifizierungsgründen oft nicht erwünscht ist. Alternativ kann die

Prozess-Isolation mit Hypervisor um die diversitäre Redundanz erweitert werden. Das Betriebssystem für die sicherheitsbezogene Anwendung unterscheidet sich dabei von dem Betriebssystem für die nicht sicherheitsbezogene echtzeitfähige SPS-Anwendung. Abbildung 43 zeigt die Architektur dieses Ansatzes. Auf der linken Seite kann das Betriebssystem mithilfe der Prozess-Isolation sowohl nicht sicherheitsbezogene Prozesse als auch einen sicherheitsbezogenen Prozess ausführen. Der zweite sichere Kanal, der die identische sicherheitsbezogene Software wie der erste sichere Kanal ausführt, kann isoliert in einer anderen VM mit einem anderen Betriebssystem ausgeführt werden.

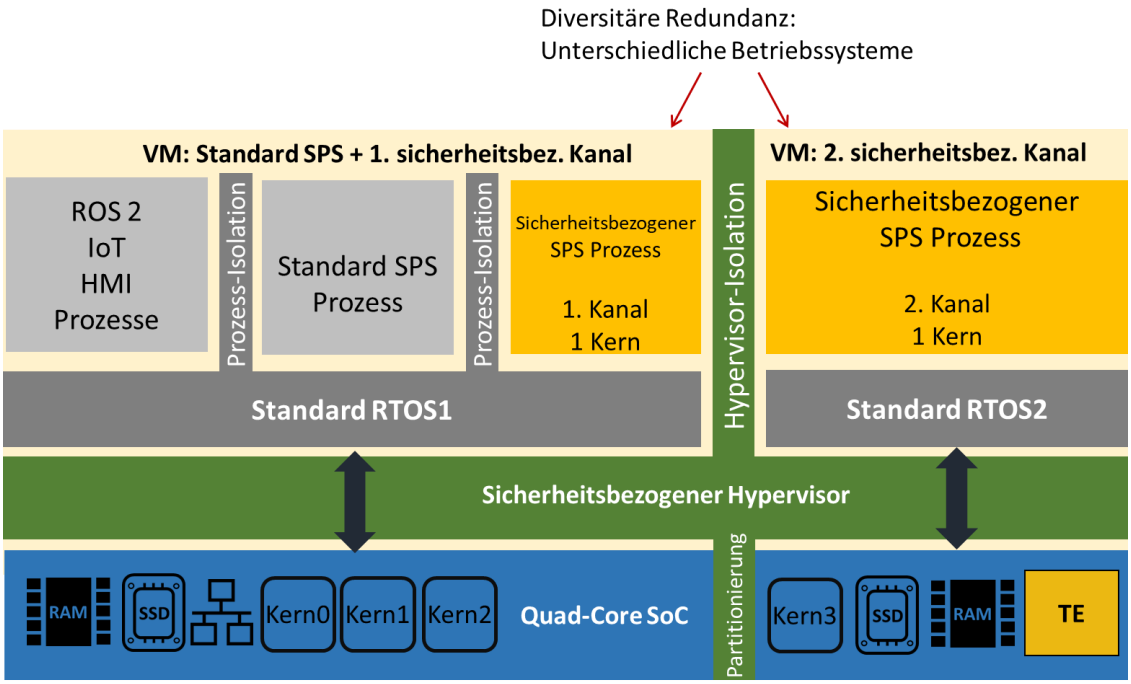


Abbildung 43: Systemarchitektur mit Hypervisor- und Prozess-Isolation auf Basis diversitärer Betriebssysteme.

Dieser Ansatz eignet sich für eine kosteneffiziente Fertigungs- und Maschinenautomatisierung, da er alle oben genannten Vorteile in sich vereint. Gleichzeitig wird der Konfigurations- und Verwaltungsaufwand reduziert, da nur zwei VMs gehosted werden. Diese vorgestellte Systemarchitektur mit Hypervisor- und Prozess-Isolation auf Basis diversitärer Betriebssysteme (Abbildung 43) wird in dieser Arbeit verwendet.

5.3 Softwarearchitektur

5.3.1 Allgemeiner Aufbau

Die Softwarearchitektur wird zunächst in die zwei Komponenten Embedded-Software und Anwendungssoftware unterteilt. Die Embedded-Software ist die Software, die vom Betriebssystem ausgeführt wird. Diese ist in der Regel in Hochsprachen wie C, C++ oder Python programmiert und enthält in einer Sicherheits-SPS sowohl die Laufzeitsystem-Software als auch die Software für die STL und den Kreuzvergleich. Ein sicherheitsbezogenes Laufzeitsystem ermöglicht das Ausführen von sicherheitsbezogenen Anwendungen in der Automatisierung. Für Embedded-Software im funktional sicheren Kontext gelten nach Sicherheitsnormen die Anforderungen für SRESW und FVL. Die sicherheitsbezogene Anwendungssoftware wird in der Regel mit grafischen IEC 61131-3 Programmiersprachen erstellt und unterliegt damit den geringeren Anforderungen von SRASW und LVL. Abbildung 44 zeigt den typischen Aufbau von einem IEC 61131-3 Laufzeitsystem. Der Anwender kann das Programm mit selbst erstellten oder vordefinierten IEC 61131-3 FB-Bibliotheken programmieren. Beim Kompilieren wird Maschinencode erzeugt, der dann in der Regel zyklisch in der SPS ausgeführt wird. Dabei werden die Eingangsdaten aus dem Prozessabbild der Eingänge (PAE) gelesen und die Ausgangsdaten in das Prozessabbild der Ausgänge (PAA) geschrieben. Die Prozessabbilder (PAE und PAA) synchronisieren ihre Daten zyklisch mit den verwendeten Feldbussen.

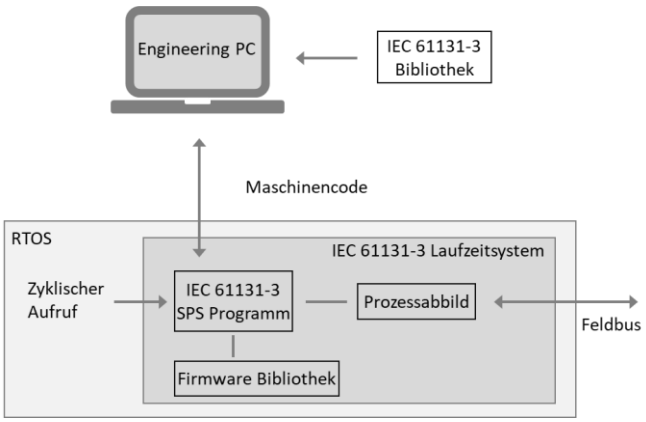


Abbildung 44: Aufbau eines IEC 61131-3 Laufzeitsystems.

Die Systemarchitektur der vorliegenden Arbeit basiert auf der Implementierung einer Standard-SPS und einer redundanten Sicherheits-SPS. Wird die Abbildung 44 um den sicherheitsbezogenen zweikanaligen Teil unter Berücksichtigung der oben vorgestellten Systemarchitektur mit Hypervisor erweitert, so ergibt sich die in Abbildung 45 darge-

stellte Softwarearchitektur. Herausforderungen bei dieser Architektur sind die Synchronisation der zyklisch auszuführenden Applikationen und der Datenaustausch zwischen den Steuerungen, da nur das Prozessabbild des nicht sicherheitsbezogenen Laufzeitsystem mit dem Feldbus synchronisiert ist.

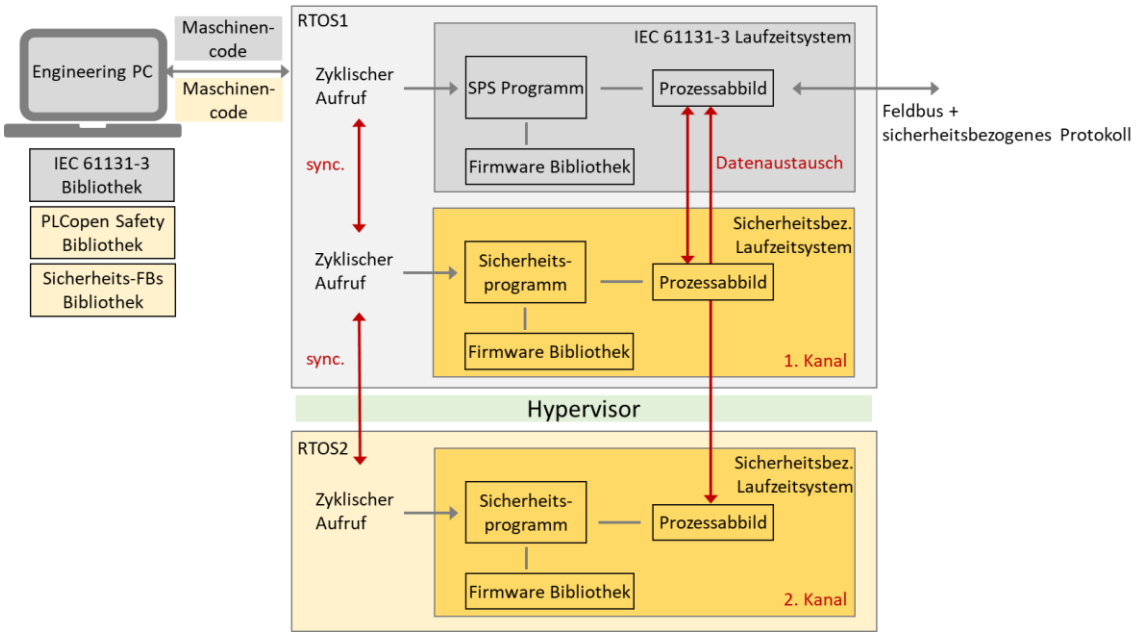


Abbildung 45: Softwarearchitektur für gemischt-kritische Anwendungen.

Weitere Softwarekomponenten sind in der Abbildung 46 dargestellt. Selbsttests und ein Software-Kreuzvergleich jeweils im sicherheitsbezogenen Teil der Steuerung, IoT-Anwendungen und in Containern isolierte Anwendungen im nicht sicherheitsbezogenen Teil komplettieren die Softwarearchitektur.

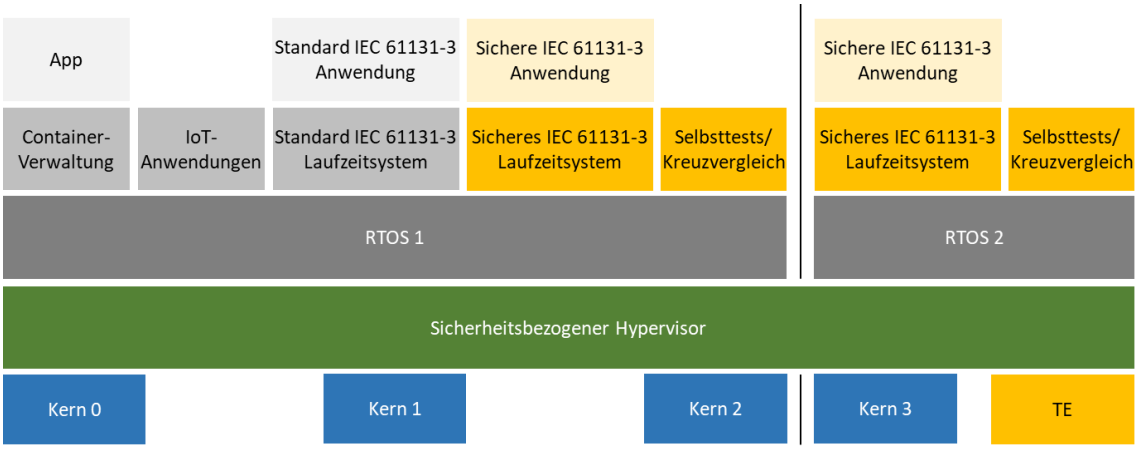


Abbildung 46: Gemischt-Kritische Softwarearchitektur mit Hypervisor ermöglicht die Implementierung von IoT- und in Containern isolierte Anwendungen.

5.3.2 Speicherschutz

In einem sicherheitsbezogenen System, das aus Embedded-Softwarekomponenten besteht, wird typischerweise nicht jede Zeile Code zertifiziert. Es ist zu beachten, dass der nicht zertifizierte Code den sicherheitsbezogenen Code nicht beeinflusst und vor allem nicht auf Ressourcen (Periphere, Speicher etc.) zugreift, die als sicher gekennzeichnet sind. Wie bei der Prozess-Isolation beschrieben, können das Betriebssystem und eine MMU oder MPU den Zugriff auf sicherheitsbezogene Speicherbereiche verwalten. Die Abbildung 47 zeigt den Speicherschutz einer Softwarekomponente, die sowohl zertifizierten als auch nicht zertifizierten Code enthält. Der gelbe Speicherbereich wird als sicher gekennzeichnet und das Betriebssystem kann mithilfe von Scheduling-Objekten wie Threads oder Prozessen den Zugriff auf diesen Speicher mit Zugriffsattributen steuern. Es darf nicht sicherheitsbezogener Code nur lesend auf den sicherheitsbezogenen Speicherbereich zugreifen, während zertifizierter Code keine Einschränkungen beim Speicherzugriff hat.

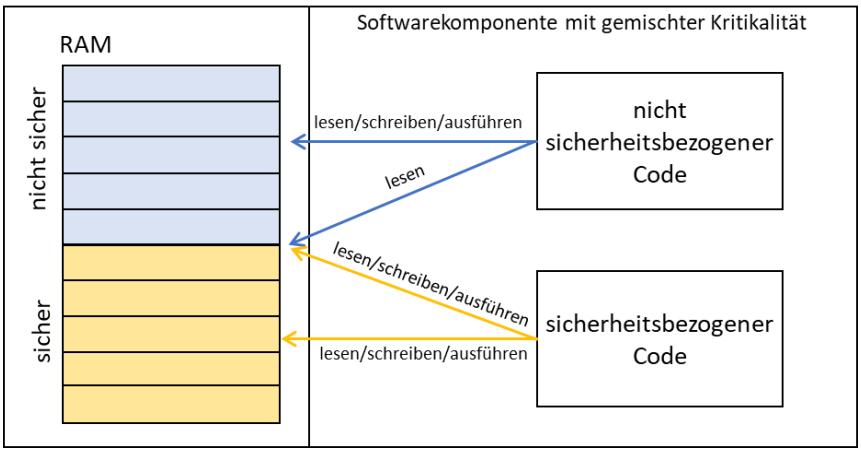


Abbildung 47: Zugriff auf Speicherbereiche von Softwarekomponenten unterschiedlicher Kritikalität.

5.3.3 Sicherheitsbezogene Kommunikation

Eine im Sicherheitsprogramm ausgeführte Sicherheitsfunktion benötigt Zugriff auf sicherheitsbezogene Eingänge (aus dem PAE) und schreibt sicherheitsbezogene Ausgänge (in das PAA). Diese sicherheitsbezogenen Daten werden in der Regel über sicherheitsbezogene Protokolle nach dem Black- oder Gray-Channel-Prinzip an die Sicherheits-SPS übertragen.

In dem vorgestellten Ansatz hat nur eine der beiden VMs Zugriff auf die Gerätetreiber der physischen Netzwerkschnittstellen (Ethernet-Ports, WLAN-Adapter etc.), an denen die Daten der Feldbus- und Sicherheitsprotokolle empfangen und gesendet werden. Dies ist dadurch bedingt, dass diese per Device Passthrough im Partition Mode des Hypervisors

exklusiv einer VM zugewiesen sind. Mithilfe von Shared Memory Bereichen können VMs effizient miteinander kommunizieren. Wenn sicherheitsbezogene Daten mit geeigneten Maßnahmen gegen Übertragungsfehler abgesichert sind, kann der gesamte Kommunikationskanal als Black- bzw. Gray-Channel betrachtet werden. Dies schließt den Shared Memory zwischen den Prozessen und VMs mit ein. In der vorgestellten Systemarchitektur hat die nicht sicherheitsbezogene Soft-SPS, die innerhalb einer VM ausgeführt wird, Zugriff auf die Ethernet-Ports und die Feldbus-PDUs. Sie extrahiert die SPDU und kopiert diese in Shared Memory Bereiche für die beiden sicherheitsbezogenen Kanäle der Sicherheits-SPS. Im vorgestellten Konzept wird ein Kanal der Sicherheits-SPS in derselben VM wie die nicht sicherheitsbezogene Soft-SPS ausgeführt, während der zweite Sicherheitskanal in einer anderen VM implementiert ist. Um mit beiden Kanälen der Sicherheits-SPS kommunizieren zu können, werden zwei verschiedene Shared-Memory Bereiche implementiert:

- Interprozess Shared Memory zur Datenübertragung zwischen Prozessen/Anwendungen innerhalb einer VM.
- Inter-VM Shared Memory zur Datenübertragung zwischen Prozessen/Anwendungen in unterschiedlichen VMs.

Die Abbildung 48 beschreibt die sicherheitsbezogene Kommunikation bei einer Single-Chip-Architektur bei der die beiden sicherheitsbezogenen Kanäle in verschiedenen VMs implementiert sind. Die weißen und gelben Blöcke sind Speicherbereiche im RAM des Single-Chip SoCs. Die weißen Blöcke sind Teil des Black- bzw. Gray-Channels und müssen nicht sicherheitsbezogen betrachtet werden.

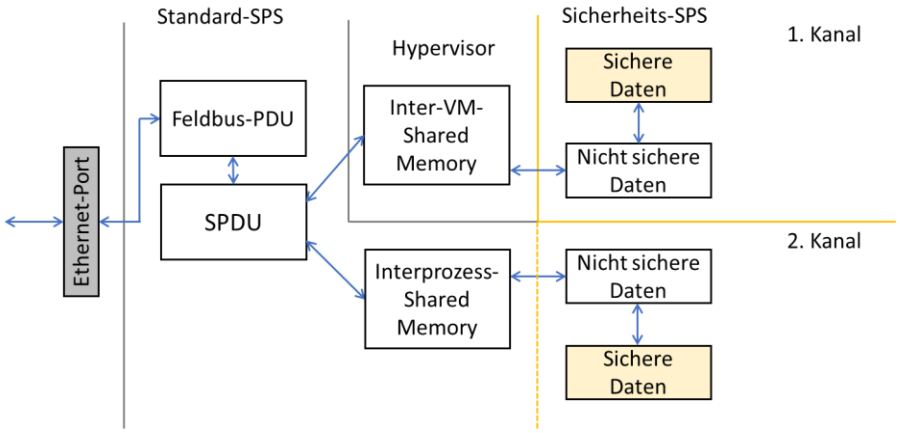


Abbildung 48: Sicherheitsbezogene Kommunikation bei einer Single-Chip-Architektur mit Hypervisor.

Abbildung 49 zeigt den Datenaustausch zwischen dem Standardteil und einem Sicherheitskanal. Der Standardteil und der dargestellte Sicherheitskanal sind in zwei getrennten VMs implementiert. Die weißen Blöcke beschreiben den nicht sicherheitsbezogenen Speicherbereich, der als Teil des Black- bzw. Gray-Channels betrachtet wird. Die Prozessdaten werden als Teil einer SPDU über einen Feldbus empfangen und werden in den Shared Memory im Hypervisor kopiert. SPDUs sind vor der Auswertung durch den SCL als nicht sichere Daten zu betrachten. Der SCL kann nach verschiedenen Ansätzen implementiert werden. Er kann entweder als sicherheitsbezogene Embedded-Komponente in der sicherheitsbezogenen Laufzeitsystem-Software oder alternativ als Komponente in der Sicherheitsanwendung, z. B. als Bibliotheks-FB, implementiert werden. Erst wenn der SCL die SPDU gegen Übertragungsfehler ausgewertet hat, können die sicherheitsbezogenen Daten, die Teil der SPDU sind, in den sicherheitsbezogenen Speicherbereich kopiert werden. Auf diese Daten hat nun nur zertifizierter Code, z. B. eine zertifizierte Sicherheitsfunktion im Sicherheitsprogramm, Zugriff. Diese Sicherheitsfunktion verarbeitet die sicherheitsbezogenen Eingangsdaten und schreibt daraufhin die Ausgangsdaten. Dies kann beispielsweise ein STO-Signal sein, welches zum Antriebssystem über die Master-SPDU gesendet wird.

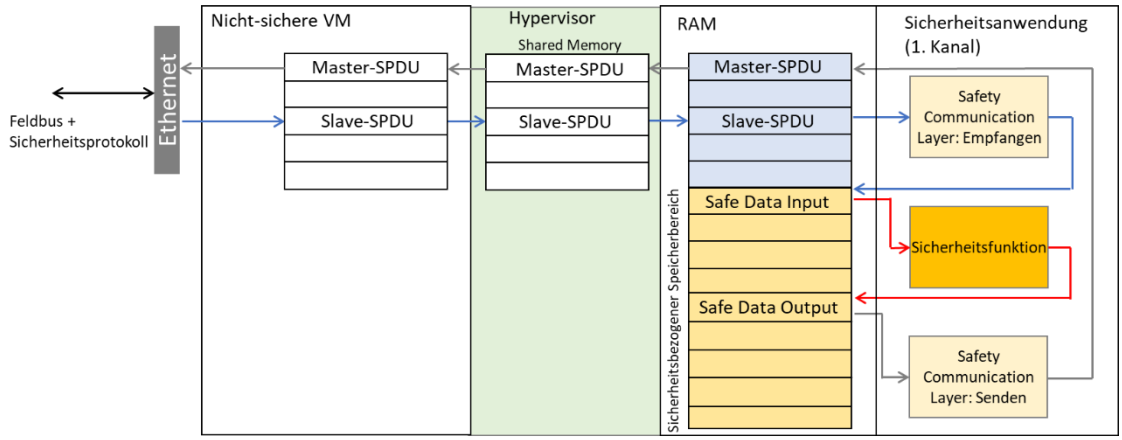


Abbildung 49: Datenaustausch von sicherheitsbezogenen Prozessdaten zwischen zwei VMs.

5.3.4 Diagnosemaßnahmen

Selbsttests

Bei der Implementierung der STL in einer virtualisierten Umgebung gibt es folgende Problematik: Die STL greift auf die zugrunde liegende Hardware des Systems zu, um diese gegen Fehler zu testen. Gleichzeitig kann eine VM nur auf die Ressourcen des Systems zugreifen, die ihr zugewiesen werden, da das Guest-Betriebssystem der VM nicht

weiß, dass es von einem Hypervisor ausgeführt wird. Dies bedeutet, dass die STL, die in einer VM ausgeführt wird, nicht das gesamte System testen kann. Daher wird bei diesem Ansatz die Software der STL in den Hypervisor integriert. Das bedeutet, dass der kompilierte STL-Code in den Hypervisor gelinkt wird. Über einen Hypercall kann eine VM den Hypervisor auffordern, die Selbsttests auszuführen. Dies führt zu den Kontextwechseln VM-Exit und VM-Entry zwischen der VM und dem Hypervisor. Diese Kontextwechsel sind jedoch vorhersehbar und können im Hinblick auf die Echtzeitfähigkeit berücksichtig werden.

Die STL kann als eigene Softwarekomponente, also als eigenständiger Prozess, oder direkt in die Laufzeitsystem-Software integriert werden. Nach der Ausführung der Selbsttests gibt der Hypervisor das Ergebnis an die jeweilige VM zurück. Es ist zu beachten, dass die VMs mit ihrem jeweiligen Betriebssystem keinen Zugriff auf die Rohdaten der Selbsttest-Ergebnisse haben sollen, da diese als einkanalig zu betrachten sind. Einkanalig bedeutet in diesem Fall, dass jede VM unterschiedliche Hardwarekomponenten, wie z. B. unterschiedliche Prozessorkerne, testet. Damit eine Verfälschung der Ergebnisse erkannt werden kann, sind Maßnahmen zur sicherheitsbezogenen Datenübertragung zu implementieren. Mithilfe der Black-Channel-Kommunikation kann sichergestellt werden, dass eine Verfälschung der Selbsttest-Daten von der TE erkannt wird. Die Implementierung der STL in der Konfiguration mit zwei sicherheitsbezogenen VMs ist in der Abbildung 50 dargestellt. Über den Block Trigger STL initiiert die jeweilige VM die Ausführung der Selbsttests im Hypervisor (blaue Pfeile). Anschließend führt der Hypervisor die Selbsttests durch und gibt die Ergebnisse an die jeweilige Anwendung zurück. VM0 hat in dieser Konfiguration keinen Zugriff auf den Gerätetreiber der TE und schreibt die Ergebnisse abgesichert in einen Shared Memory Bereich. Anschließend werden beide Ergebnisse von VM1 an die TE gesendet und dort ausgewertet.

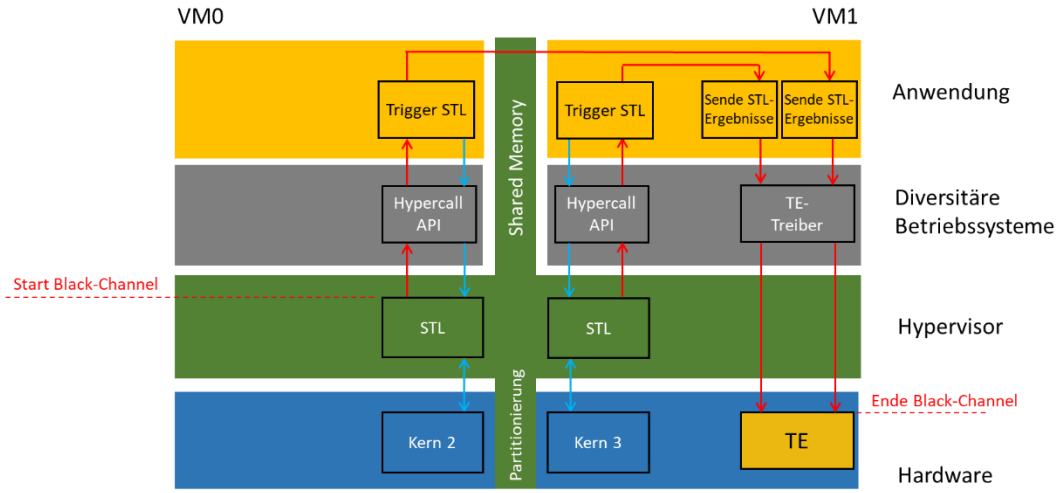


Abbildung 50: Softwarearchitektur der Selbsttests für ein zweikanaliges System mit Hypervisor.

Kreuzvergleich

Die Abbildung 51 zeigt die Softwarearchitektur des Kreuzvergleichs, bei dem beide VMs einen Snapshot über den ihrer jeweiligen VM zugewiesenen sicherheitsbezogenen Speicherbereich bilden (blaue Pfeile). Bei einem Snapshot über einen Speicherbereich handelt es sich um eine Momentaufnahme des Inhalts des Speicherbereichs. Die Snapshots werden von beiden Kanälen an die TE gesendet, die sie miteinander vergleicht (rote Pfeile). Der Software-Kreuzvergleich basiert im Gegensatz zu den Selbsttests auf einer zweikanaligen Architektur.

Hierbei ist ebenfalls zu beachten, dass nur eine VM Zugriff auf den Gerätetreiber der TE hat und damit für das Senden der beiden Snapshots an die TE verantwortlich ist. Der Zugriff auf den Snapshot der anderen VM wird über Shared Memory realisiert. Wie die Selbsttest-Ergebnisse sind auch die Snapshots gegen Übertragungsfehler geschützt.

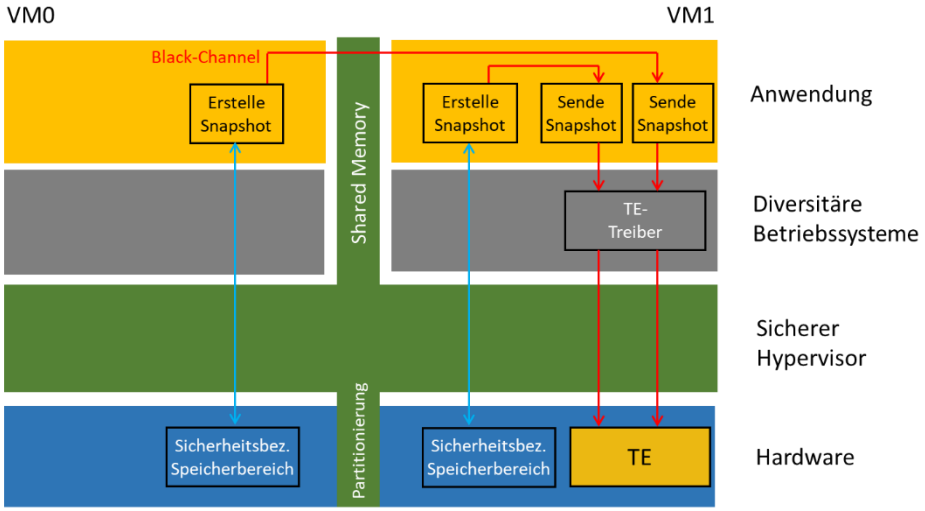


Abbildung 51: Softwarearchitektur des Software-Kreuzvergleichs für ein zweikanaliges System mit Hypervisor.

6 Implementierung der Systemarchitektur

In diesem Kapitel wird die Implementierung der im vorhergehenden Kapitel entwickelten Systemarchitektur beschrieben. Außerdem werden die verwendeten Hardware- und Softwarekomponenten vorgestellt.

6.1 Intel Atom x6000E Prozessor

6.1.1 Allgemeine Informationen

Basiert der Ansatz der Compound-SPS auf einer Single-Chip-Lösung mit externer TE (auf separatem Silizium-Die), ist ein für die funktionale Sicherheit zertifiziertes SoC erforderlich. Ebenso sollte das SoC mindestens vier Prozessorkerne besitzen, um wie oben beschrieben zwei Kerne für die erforderliche Redundanz des sicherheitsbezogenen Teils und zwei Kerne für den rechenintensiven Anwendungsteil bereitzustellen.

Die Intel Atom x6000E Serie kombiniert IoT-Funktionen, Echtzeitfähigkeit, Verwaltbarkeit, Security und funktionale Sicherheit in einem SoC [57]. Der Intel Atom x6427FE ist als Dual- und Quad-Core SoC verfügbar. Zwei Modelle der Reihe erfüllen die Anforderungen der funktionalen Sicherheit nach IEC 61508 SIL2 und ISO 13849-1 Kategorie 3, PL d. Der x6427FE beinhaltet eine Intel Safety Island (ISI) auf einem zweiten Silizium-Die als externe TE, welche mit der CPU zusammenarbeitet, um Fehler aufzudecken, Fehler zu melden und Diagnosetests auszuwerten. In dieser Arbeit wird der Quad-Core SoC x6427FE mit einer Taktfrequenz von 1,9 GHz verwendet.

6.1.2 Single-Chip Kategorie 3 Architektur

Analog zum Lockstep Prozessor lässt sich das Intel x6427FE SoC nicht direkt auf eine der vorgesehenen Architekturen der ISO 13849-1 abbilden. Folglich gelten höhere Diagnoseanforderungen, um mit dem x6427FE SoC ein System zu entwickeln, das eine Kategorie 3, PL d äquivalente Risikoreduzierung bietet. Das Intel x6427FE SoC verfügt über Hardware-Maßnahmen wie Error Correction Code (ECC) und Parity-Bits, um Fehler zu erkennen und ggf. zu korrigieren. Darüber hinaus sind Software-Diagnosemaßnahmen implementiert, die sowohl Selbsttests des Systems als auch einen Software-Kreuzvergleich der Prozessorkerne für die Sicherheitsanwendung umfassen. Die Lockstep vergleichbare Architektur erreicht mit den Hardware- und Software-Diagnosemaßnahmen einen DC_{avg} von hoch [58]. Eine Anforderung für die Erfüllung des hohen DC_{avgs} ist die Durchführung aller Selbsttests in jedem Zyklus der Sicherheitsanwendung [57].

6.1.3 Externe Testeinrichtung

Die externe Testeinrichtung ISI ist ein eigenständiger Prozessor auf einem zweiten Silizium-Die, der die ordnungsgemäße Funktion des Systems überprüft und als Sicherheits-Monitor fungiert. Sie ist vergleichbar mit der Testeinrichtung eines ISO 13849-1 Kategorie 2 Systems und eines Lockstep Prozessors. Sie überwacht die Alarme, die durch die im SoC implementierten Diagnosemaßnahmen erzeugt werden. Zusätzlich überwacht sie die Durchführung von periodischen Software-Diagnosemaßnahmen und überführt den SoC bei Fehlererkennung in den sicheren Zustand [57].

6.2 Hardwareausführung

Smart Mobility Architecture (SMARC) ist eine Spezifikation für Computer-on-Modules. SMARC Computer-on-Modules sind speziell für die Entwicklung von kompakten Low-Power-Systemen ausgelegt. In der vorliegenden Arbeit wird das SMARC 2.1.1 Modul mit dem Intel x6427FE der Firma SECO S.p.A. verwendet [130]. Das zugehörige Baseboard wurde im Zuge dieser Arbeit für den industriellen Einsatz in Bezug auf Schnittstellen und den Formfaktor angepasst. Abbildung 52 zeigt die für den Einsatz in FTFs und AMRs optimierte Hardwareausführung.

Abbildung 52: Hardwareausführung der Compound-SPS.

6.3 ACRN Hypervisor

6.3.1 ACRN Project

In dieser Arbeit wird die Rückwirkungsfreiheit zwischen dem sicherheitsbezogenen und nicht sicherheitsbezogenen Teil der Steuerung für gemischt-kritische Anwendungen mithilfe der hypervisorbasierten Isolation realisiert. Dieser Ansatz erfordert einen echtzeitfähigen Hypervisor, der für die funktionale Sicherheit zertifiziert ist.

Viele bestehende Hypervisor-Lösungen bieten nicht die richtige Größe, Echtzeitunterstützung und Flexibilität für IoT-Geräte. Ebenso ist der Code von vielen Hypervisoren zu umfangreich, bietet keine Sicherheits- oder harten Echtzeitfunktionen und erfordert zusätzlich zu viel Overhead für die Entwicklung von Embedded-Systemen. Der ACRN Hypervisor der Linux Foundation wurde von Intel als Referenz-Hypervisor für das Intel x6427FE SoC empfohlen. Weiterhin wird er stetig durch eine skalierbare Open-Source-Referenzplattform optimiert. ACRN wird in einem Github-Repository zur Verfügung gestellt und ist Stand Januar 2024 in der Version 3.2 verfügbar, wobei die Version 1.4 für die funktionale Sicherheit zertifiziert wurde. In dieser Arbeit wird die Version 2.7 verwendet, da diese Funktionen wie z. B. Shared Memory unterstützt, die für Umsetzung der vorgestellten Systemarchitektur relevant sind und in der sicherheitszertifizierten Version 1.4 noch nicht verfügbar waren. Es ist jedoch zu beachten, dass die Version 2.7 ohne Zertifizierung nicht in einer Steuerung für funktional sichere Anwendungen eingesetzt werden darf. Der ACRN Hypervisor in der Partition Mode Konfiguration dient als Referenz-Hypervisor, um die Umsetzung der Systemarchitektur zu verifizieren.

6.3.2 Inter-VM Kommunikation mit Shared Memory

Der ACRN Hypervisor unterstützt eine Inter-VM Kommunikation über Shared Memory. Der ACRN Hypervisor selbst emuliert ein virtuelles Peripheral Component Interconnect (PCI)-Gerät, ein sogenanntes IVSHMEM Device, um die virtuelle Basisadresse und die Größe des Shared Memorys bereitzustellen. Die Speicherbereiche des Shared Memorys werden dabei im Speicherbereich des Hypervisors reserviert. Der Datenpfad ist kurz und bietet eine hohe Bandbreite mit geringer Latenz. Da das IVSHMEM Device ein emuliertes PCI-Gerät ist, kann eine Anwendung innerhalb einer VM darauf wie auf ein Standard-PCI-Gerät zugreifen, ohne dass ein Kontextwechsel zwischen VM und Hypervisor ausgelöst wird. Mithilfe eines IVSHMEM Servers können VMs über das IVSHMEM Device einander Benachrichtigungen in Form von Interrupts senden. Dieser Mechanismus wird z. B. für die Synchronisation der VMs verwendet.

6.4 Betriebssysteme

6.4.1 Zephyr OS

Für die Realisierung des sicherheitsbezogenen Steuerungsteils ist entweder ein sicherheitsbezogenes zertifiziertes Betriebssystem oder zwei unterschiedliche Betriebssysteme für die diversitäre Redundanz erforderlich, um SIL2 und PL d zu erreichen. Zephyr OS basiert auf einem Kernel, der für den Einsatz auf ressourcenbeschränkten und eingebetteten Systemen von der Linux Foundation entwickelt wurde [131]. Der Zephyr Kernel unterstützt mehrere Architekturen, darunter ARM Cortex-M, ARM Cortex-A, ARM Cortex-R und Intel x86. Zephyr steht unter der Apache 2.0 Open-Source-Lizenz und wird in einem Github-Repository verwaltet. Der Zephyr Kernel ist Stand Januar 2024 in der Version 3.5 verfügbar, jedoch wird die Version 2.7 in dieser Arbeit verwendet. Das Zephyr Betriebssystem wird in der vorliegenden Arbeit als Betriebssystem in einem der sicherheitsbezogenen Kanäle verwendet, da es von Intel als Referenz Betriebssystem beim Intel x6427FE SoC empfohlen wird. Auch wenn keine Zertifizierung aufgrund der diversitären Redundanz erforderlich ist, gibt es von der Community Bestrebungen das Betriebssystem Zephyr zu zertifizieren [132].

Der Zephyr Kernel bietet eine platzsparende und leistungsstarke Multi-Threading Ausführungsumgebung. Der Rest des Zephyr-Ökosystems, einschließlich der Gerätetreiber, des Netzwerkstacks und des Anwendungs-Codes, nutzt die Funktionen des Kernels, um eine vollständige Anwendung zu erstellen. Da der Zephyr Kernel konfigurierbar ist, werden nur die Funktionen integriert, die für die spezifische Anwendung benötigt werden. Dadurch eignet sich der Zephyr-Kernel besonders für Anwendungen mit begrenztem Speicher oder für funktional sichere Anwendungen, da weniger Code in der Regel zu weniger Fehlern führt. Da das Zephyr Betriebssystem eine sehr ressourcensparende Ausführungsumgebung ist, sind in der verwendeten Version 2.7 einige Funktionen klassischer Echtzeitbetriebssysteme nicht enthalten, wie z. B. die Unterstützung von Multitasking mit virtuellen Adressen [131].

6.4.2 Echtzeitfähiges Linux Betriebssystem

In der vorgestellten Systemarchitektur ist ein echtzeitfähiges Linux Betriebssystem für die Ausführung der echtzeitkritischen, nicht sicherheitsbezogenen Soft-SPS und für den ersten sicherheitsbezogenen Kanal zuständig. Linux ist eines der leistungsfähigsten Betriebssysteme aufgrund der großen Anzahl unterstützter CPU-Architekturen, der hohen Anzahl von Treibern und der guten Portabilität und Skalierbarkeit. Linux wurde als

Allzweck-Betriebssystem entwickelt, gefolgt von mehreren Ansätzen zur Einführung von Echtzeitfähigkeit in den Kernel. Der entwickelte PREEMPT_RT-Patch zielt darauf ab, die Vorhersagbarkeit zu erhöhen und die Latenzzeiten des Kernels zu reduzieren [133]. Der für diese Arbeit verwendete Linux Kernel (Version 5.4.143) mit dem PREEMPT_RT-Patch (Version 63) wurde mit dem Buildsystem Yocto Project [134] erstellt, um den Kernel an die Anforderungen der Architektur anzupassen.

6.5 CODESYS Laufzeitsysteme

6.5.1 CODESYS Development System

Das CODESYS Development System ist eine herstellerunabhängige IEC 61131-3 Automatisierungssoftware zur Projektierung von Steuerungssystemen [135]. Das Programmiersystem ist lizenzfrei und unterstützt die IEC 61131-3 Programmiersprachen. Zusätzlich zu den IEC 61131-3 konformen Programmiersprachen können in CODESYS Programme mit der Programmiersprache Continuous Function Chart (CFC) programmiert werden.

6.5.2 CODESYS Control

CODESYS Control ist das Laufzeitsystem zum CODESYS Development System. Mithilfe des CODESYS Control Laufzeitsystems als Soft-SPS-Erweiterung ist es möglich aus einem Embedded oder PC-basierten Gerät, eine IEC 61131-3 konforme Industriesteuerung zu realisieren. CODESYS Control for Linux SL ist das Laufzeitsystem für linuxbasierte Embedded- und PC-Systeme. Es bietet in der Standardinstallation diverse Feldbusunterstützungen für CANopen, EtherCAT, EtherNet/IP, PROFIBUS und PROFINET. Zusätzlich können weitere Produkte und Bibliotheken nachlizensiert werden.

6.5.3 CODESYS Safety SIL2

Neben dem Laufzeitsystem CODESYS Control für Standardsteuerungen bietet CODESYS die Möglichkeit, Sicherheits-SPSen mit dem sicherheitsbezogenen Laufzeitsystem CODESYS Safety SIL2 zu entwickeln [136]. Es ist bis SIL2 zertifiziert und basiert auf dem Standard Laufzeitsystem. Sicherheits-SPSen mit dem SIL2 Laufzeitsystem werden ebenso wie die nicht sicherheitsbezogenen Steuerungen mit dem CODESYS Development System programmiert. Es ist nur eine Programmierumgebung notwendig, um sowohl die Sicherheits- als auch die Standardanwendung zu projektieren und zu programmieren.

6.5.4 Compound-SPS

Mit dem CODESYS Safety SIL2 Plug-In wird die CODESYS Entwicklungsumgebung um eine Compound-SPS erweitert [137]. Voraussetzung dafür ist, dass ein Standard Laufzeitsystem und ein Safety SIL2 Laufzeitsystem in einer Steuerung integriert sind. Es ist dabei unerheblich, ob diese innerhalb desselben Prozessors, auf getrennten Prozessoren oder sogar auf getrennten Boards mit getrennter Stromversorgung implementiert sind. Für den End-Anwender verhält sich die Compound-SPS immer gleich. Eine Compound-SPS besteht aus einer logischen Root-Steuerung, welche zwei physische Steuerungen unterschiedlicher Kritikalität beinhaltet. Der Anwender kann die beiden Applikationen unabhängig voneinander programmieren und die Programme auf die jeweilige Steuerung laden. Diese Trennung ermöglicht ein Sicherheitskonzept, bei dem die Änderung der Standardapplikation keine erneute Abnahme der Sicherheitsapplikation bedingt.

CODESYS ermöglicht den direkten Datenaustausch zwischen nicht sicherheitsbezogener und sicherheitsbezogener SPS innerhalb einer Compound-SPS. Die Exchange Variable Connection (EVC) mit der Exchange Variable List (EVL) wird hier in der hypervisorbasierten Architektur mit Shared Memory realisiert. Mithilfe von EVC stehen ausgewählte Variablen aus der Sicherheitsapplikation in der Standardapplikation zur Verfügung und können wie herkömmliche Variablen verwendet werden. Diese Variablen sind als Messpunkte zu verstehen, die ohne Eingriff in die bestehende Sicherheitsanwendung analysiert werden können. Die EVL ist ein spezieller Typ einer globalen Variablenliste, die den Austausch von nicht sicherheitsbezogenen Daten zwischen einer Sicherheits- und einer Standards-SPS innerhalb einer Compound-SPS ermöglicht. In der EVL können Variablen wie in einer herkömmlichen Variablenliste deklariert und mit nicht sicherheitsbezogenen Daten aus der Standardapplikation verknüpft werden. Diese nicht sicherheitsbezogenen Daten stehen dann in der Sicherheitsapplikation bereit. Dadurch sind auch Schreibzugriffe auf diese Variablen möglich.

6.6 Systemarchitektur der Compound-SPS

Die verwendeten Komponenten, Verfahren und Methoden werden hier in diesem Kapitel zu einer Compound-SPS als Modulsteuerung für gemischt-kritische Anwendungen in der Automatisierung zusammengefasst. Ein Quad-Core SoC mit On-Chip Diagnosemaßnahmen und hoher Rechenleistung bildet hierbei die Basis. Das Intel Atom x6427FE SoC erfüllt die Anforderungen und wird in dieser Arbeit als Referenz-SoC verwendet.

Der ACRN Hypervisor in der Version 2.7 bietet bis auf die fehlende Sicherheitszertifizierung alle notwendigen Features, wie Shared Memory, Device Passthrough und die erforderliche Partitionierung. Der ACRN als Typ 1 Hypervisor ermöglicht die Ausführung verschiedener Betriebssysteme auf derselben Hardware in Form von VMs. Da der Einsatz eines zertifizierten Betriebssystems aus Kostengründen oft nicht erwünscht ist, wird zur Realisierung der beiden sicherheitsbezogenen Kanäle die diversitäre Redundanz, d. h. der Einsatz von zwei unterschiedlichen Betriebssystemen, eingesetzt. Eine vollständige Isolation der beiden sicherheitsbezogenen Kanäle wird gewährleistet, da beide Kanäle in getrennten VMs implementiert werden. Jede VM bekommt ihre eigenen Prozessorkerne, RAM und nichtflüchtigen Speicher (SSD, SD-Karte etc.) zugewiesen. Die beiden Kanäle können sich somit nicht gegenseitig beeinflussen. Eine Verwendung von Hardwareressourcen aus mehreren VMs ist nicht möglich, da der Zugriff auf die Hardware mit Device Passthrough umgesetzt ist. Mit der Inter-VM Kommunikation über Shared Memory bietet der Hypervisor eine effiziente Methode für den Datenaustausch zwischen den VMs. SPDUs, die Ergebnisse der Selbsttests und die Snapshots für den Software-Kreuzvergleich können über Shared Memory ausgetauscht werden. Die Interrupt-Funktionalität vom IVSHMEM beschreibt einen Ansatz zur Synchronisation von VMs. Das in dieser Arbeit verwendete Linux Betriebssystem wird mit dem Yocto Project erstellt und wird von einer VM gehostet. Innerhalb dieser VM werden sowohl das sicherheitsbezogene als auch das nicht sicherheitsbezogene CODESYS Laufzeitsystem in getrennten Prozessen ausgeführt. Dieser VM sind drei Prozessorkerne fest zugeordnet, wobei zwei Prozessorkerne für den rechenintensiven SPS-Prozess und weitere IoT-Anwendungen genutzt werden, während ein Prozessorkern den ersten sicherheitsbezogenen Kanal mit dem CODESYS SIL2 Laufzeitsystem darstellt. Der zweite sicherheitsbezogene Kanal wird in Form einer zweiten VM mit dem Zephyr Betriebssystem realisiert. Dieser VM wird vom Hypervisor ein Prozessorkern exklusiv zugewiesen und das CODESYS SIL2 Laufzeitsystem wird redundant ausgeführt. Die Abbildung 53 zeigt die Systemarchitektur der Compound-SPS auf Basis des ACRN Hypervisors und zeigt die verwendeten Komponenten.

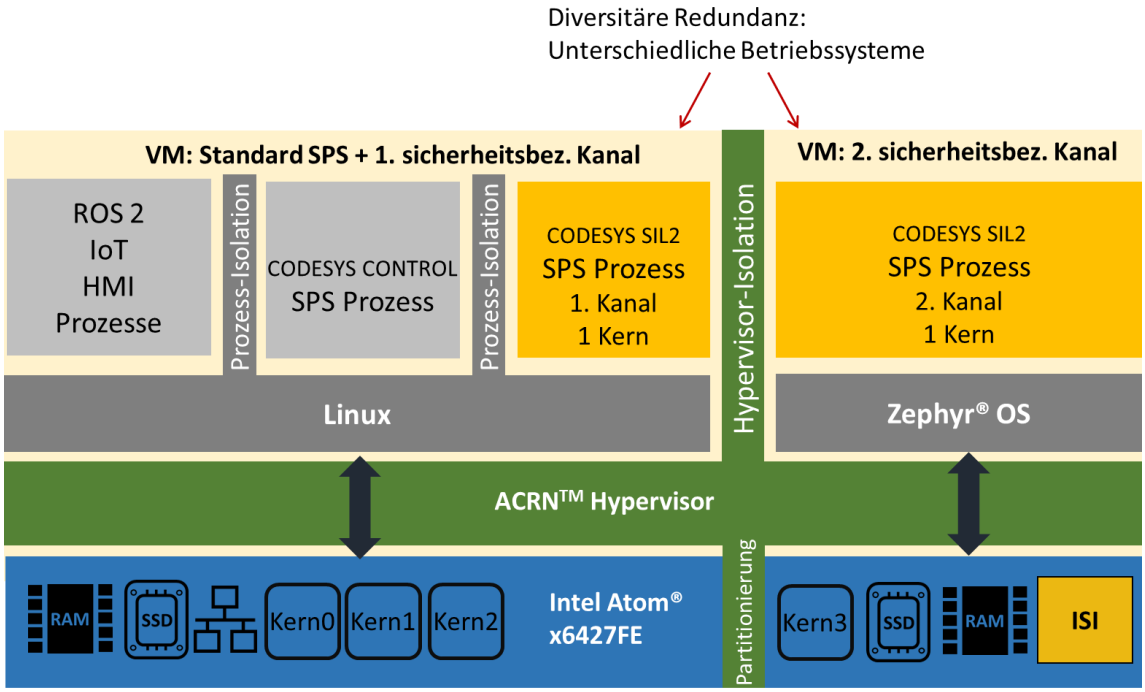


Abbildung 53: Systemarchitektur der Compound-SPS auf Basis verwendeter Komponenten.

7 Validierung des Konzepts

Die Validierung der in dieser Arbeit entwickelten Konzepte und Methoden erfolgt durch eine praktische Umsetzung. Im Folgenden wird die Umsetzung der zentralen Sicherheitsfunktionen, der zentralen Diagnose der Komponenten und der dafür notwendigen sicherheitsbezogenen Datenübertragung vorgestellt. Die Compound-SPS als zentrale Standard- und Sicherheits-SPS für gemischt-kritische Anwendungen für Industrie 5.0 wird mithilfe der Entwicklungsumgebung CODESYS Development System projektiert und programmiert.

7.1 Allgemeiner Aufbau

Zur Umsetzung und Veranschaulichung der in dieser Arbeit erarbeiteten Konzepte wird ein Demonstrator vorgestellt. Die Abbildung 54 zeigt den Technologiedemonstrator auf dem Messestand der Firma SEW-EURODRIVE auf der Messe SPS – Smart Production Solutions 2022 in Nürnberg.

Abbildung 54: Technologiedemonstrator auf der SPS - Smart Production Solutions 2022.

Im Mittelpunkt des Demonstrators steht die Compound-SPS auf Basis des Intel x6427FE SoCs mit CODESYS Control und CODESYS Safety SIL2. Zunächst ist anzumerken, dass das verwendete CODESYS Safety SIL2 Laufzeitsystem derzeit noch kein offizielles Produkt ist und die im Rahmen dieser Arbeit vorgenommenen Anpassungen für das Intel x6427FE SoC noch nicht sicherheitszertifiziert sind.

Da das vorgestellte Sicherheitskonzept mit einer Compound-SPS für gemischt-kritische Anwendungen insbesondere bei Maschinenmodulen wie Robotern seine Vorteile zur Geltung bringt, wird mit dem Demonstrator ein Delta-Parallel-Roboter der Firma autonox Robotics [138] gezeigt, der durch eine nichtlineare Kinematik charakterisiert wird. Der Tripod-Roboter wird von drei PMAC-Synchronservomotoren, die jeweils 120° zueinander ausgerichtet sind, gesteuert. Für die Ansteuerung der drei Servomotoren werden drei Servoregler nach dem in [110] vorgestellten Konzept für feldbusbasierte Sicherheitsfunktionen verwendet.

Die Kommunikation zwischen der Compound-SPS und den Servoreglern basiert auf dem Feldbus EtherCAT mit dem sicherheitsbezogenen Protokoll FSoE. Für die Implementierung von Sicherheitsfunktionen, die eine Bewegung oder eine Kraft überwachen, werden sicherheitsbezogene Messwerte wie die Winkel und Ströme der drei Motoren im Sicherheitsteil der Compound-SPS benötigt. In den Servomotoren sind Motoreinbau-Drehgeber mit der sicherheitsbezogenen digitalen Schnittstelle EnDat 3 integriert. EnDat 3 basiert auf dem Gray-Channel-Prinzip und sichert die redundanten Positionswerte mit Maßnahmen für eine sicherheitsbezogene Datenübertragung ab. Es wird der unterlagerte Feldbus EtherCAT als Gray-Channel verwendet. Die Ströme werden nach dem Prinzip aus [112] mit dem FSoE-Frame verkettet und nutzen den unterlagerten EtherCAT-Feldbus als Gray-Channel. Um ein Eindringen in den Arbeitsbereich des Roboters sicher zu erkennen, wird ein sicherheitsbezogener Lichtvorhang über eine FSoE-Slave-Klemme angebunden. Neben der zyklischen Feldbuskommunikation bietet DDS eine effiziente C2C-Vernetzung.

Die gesamten Daten (OT und IT) werden von dem nicht sicherheitsbezogenen Teil (in der Linux VM) empfangen. Die sicherheitsbezogenen Daten, also die FSoE-, Strom-, EnDat 3- und optional OPC UA Safety SPDUs werden über Shared Memory an die zwei sicherheitsbezogenen Kanäle der Sicherheits-SPS weitergeleitet. Das logische Blockschaltbild des Demonstrators ist in der Abbildung 55 zu sehen. Die Sicherheits-SPS kann nach außen hin als eine logische Einheit betrachtet werden. Für die Anwendung und das Sicherheitskonzept ist die physische Architektur der Compound-SPS unerheblich.

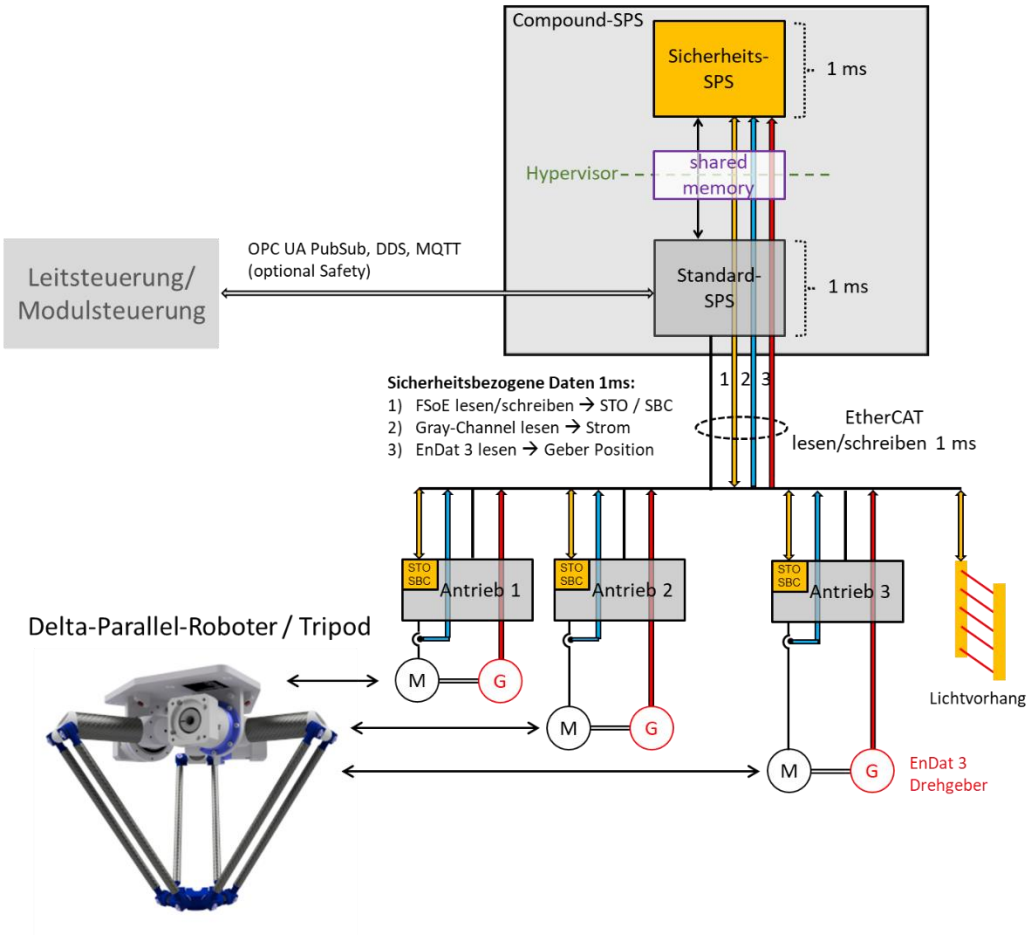


Abbildung 55: Übersichtsbild für eine gemischt-kritische Compound-SPS mit zentralen Sicherheitsfunktionen für einen Tripod-Roboter.

7.2 Konfiguration des SPS-Projekts

Als Projektier- und Programmier-Entwicklungsumgebung für die Compound-SPS wird das CODESYS Development System in der Version V3.5.17.0 mit dem SIL2 Plug-In verwendet. Der Projektbaum ist in Abbildung 56 dargestellt. Die Compound-SPS besteht aus einer übergeordneten Root-Steuerung, die die beiden Steuerungen unterschiedlicher Kritikalität implementiert. In der Sicherheits-SPS wird eine EVL definiert, die nicht sicherheitsbezogene Variablen enthält, die für den Datenaustausch zwischen Sicherheits- und Standard-SPS verwendet werden. In der EVL werden die Drehgeber-SPDUs, Strom-SPDUs und FSoE-SPDUs deklariert. In der Standard-SPS wird die verwendete Kinematik als Tripod Achsgruppe hinzugefügt. Die Zykluszeiten der beiden SPSen betragen 1 ms. Es ist auch möglich, die Standardanwendung mit einer schnelleren Zykluszeit auszuführen, jedoch ist es für die Synchronisation einfacher, die sicherheitsbezogene und nicht sicherheitsbezogene Anwendung mit der gleichen Zykluszeit auszuführen.

Das Programm der Standardanwendung ist in unterschiedlichen IEC 61131-3 Programmiersprachen programmiert. Die Handbedienung ist in CFC programmiert während das Hauptprogramm und der Automatikbetrieb in ST programmiert sind. Die Sicherheitsanwendung ist in FBD, d. h. in einer LVL-Programmiersprache, programmiert und unterliegt den geringeren Anforderungen von SRASW gemäß den geltenden Normen. Die FBs für die Sicherheitsfunktionen und für die SCLs sind in ST programmiert. Damit unterliegen sie den Regeln und Empfehlungen des Extended Levels der PLCopen [91].

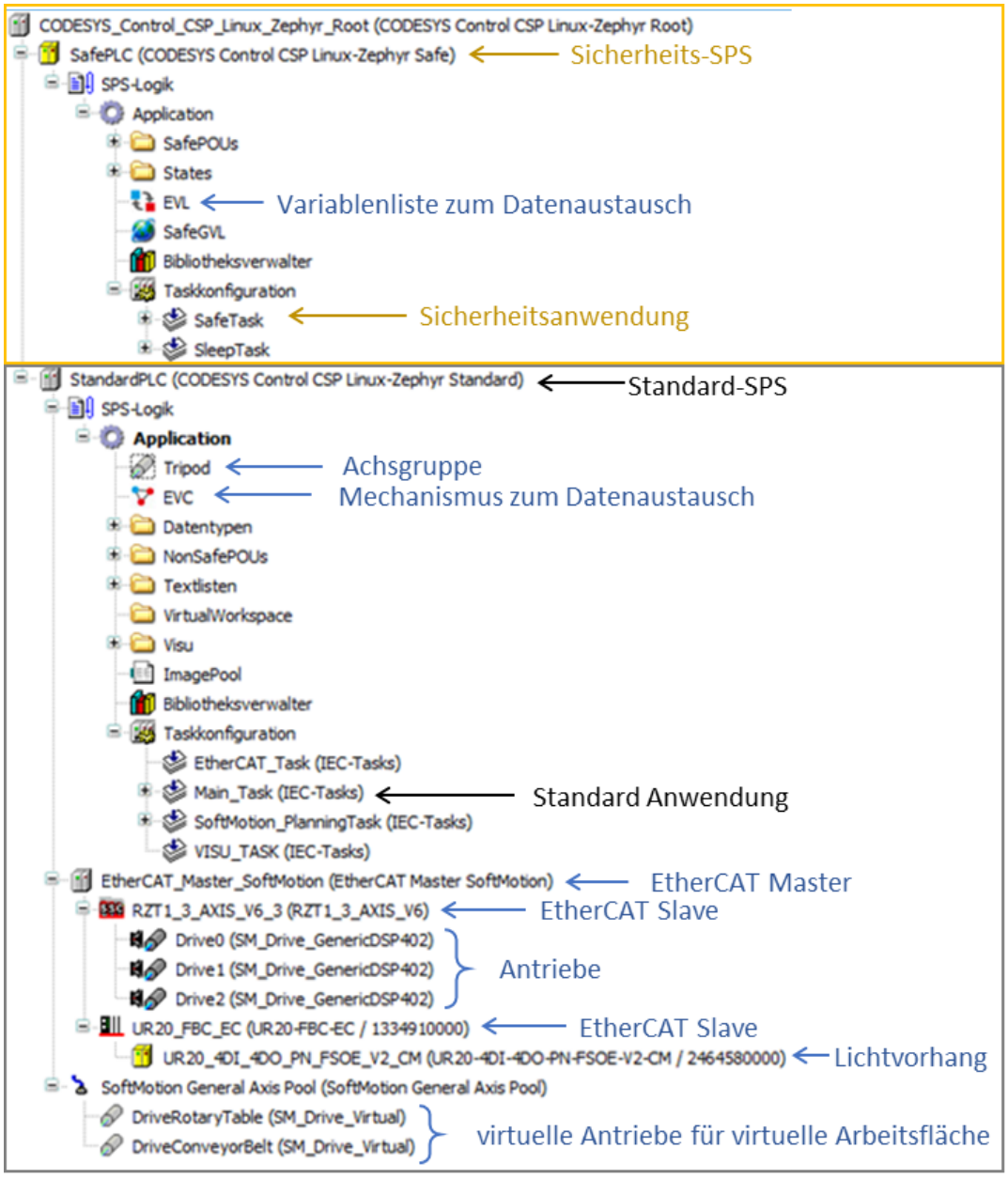


Abbildung 56: CODESYS Projektbaum.

7.3 Roboter Bewegungssteuerung

CODESYS unterstützt die PLCopen FBs für Robotik- und CNC-Anwendungen wie in [22] beschrieben. Die FBs können per Lizenz aktiviert werden und in das CODESYS-Projekt als IEC 61131-3 Bibliothek hinzugefügt werden. Die Abbildung 57 zeigt beispielhaft den FB *MC_MoveDirectAbsolute*, der über ein Interface die Tripod-Achsgruppe ansteuert. Die Sollwerte werden über eine Koordinatentransformation berechnet. Anschließend greift der FB über die Tripod-Achsgruppe auf die verknüpften Antriebe zu und sendet über EtherCAT die Sollwerte an die Antriebe. Die Variablen für die Sollwertvorgabe sind dabei mit Buttons einer grafischen Visualisierungsoberfläche verknüpft und können dort von einem Benutzer per Handbetrieb bedient werden.

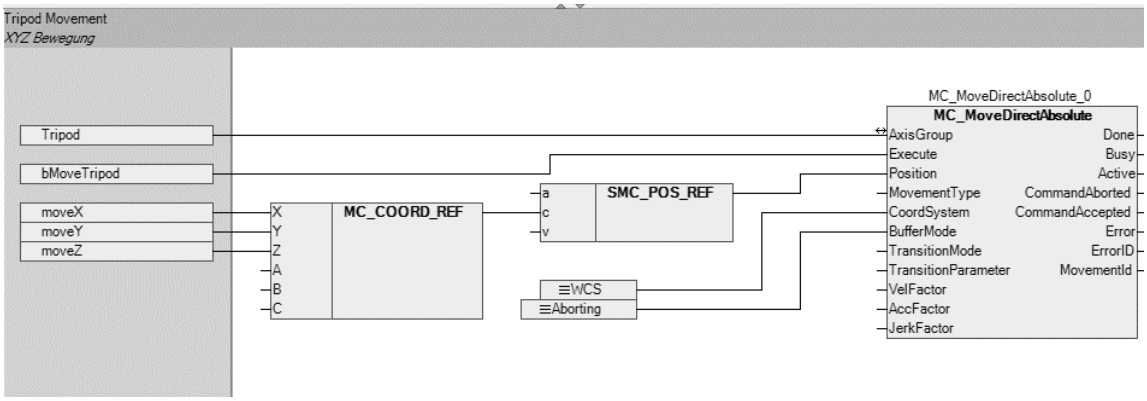


Abbildung 57: FBs für die Ansteuerung von Roboter-Systemen.

Der Demonstrator bietet neben dem Handbetrieb einen virtuellen Pick-and-Place Automatikbetrieb mit einem 32 Zoll Bildschirm als virtuelle Arbeitsfläche. Diese Anwendung wird hier nicht gezeigt, basiert jedoch auch auf den PLCopen Motion-Control FBs.

7.4 Publish-Subscribe-Kommunikation

7.4.1 Publish-Subscribe-Methoden

Die bereits genannten Vorteile der PubSub-Kommunikation, wie das Hinzufügen und Entfernen von Geräten im laufenden Betrieb und die deutlich geringere Datenmenge pro Sekunde, machen die PubSub-Methoden aus dem IT-Sektor auch für die industrielle Automatisierung interessant. CODESYS und TIA-Portal (Siemens) bieten heute schon Bibliotheken für die PubSub-Kommunikation über OPC UA und MQTT an. DDS als Middleware für ROS 2 kommt im Gegensatz zu MQTT ohne Broker aus und bietet sich für den echtzeitkritischen Datenaustausch zwischen SPSen an, wird heute jedoch noch nicht von CODESYS als IEC 61131-3 Bibliothek angeboten. Da die vorgestellte Compound-SPS

den IPC-basierten Ansatz verfolgt und der Standardteil aufgrund des Linux Betriebssystems flexibel und skalierbar ist, können Anwendungen wie DDS als eigenständiger Linux-Prozess implementiert werden. eProsimas DDS Implementierung unterstützt neben Windows, Mac OS und QNX 7.1 auch Linux. Eine Interprozesskommunikation auf Basis von Shared Memory erlaubt den Datenaustausch zwischen Linux-Prozessen und der Standard-SPS-Anwendung.

7.4.2 Synchronisation von Prozessdaten über DDS

Im Folgenden wird die praktische Umsetzung der vorgestellten Methoden aus Kapitel 4.3 zur Synchronisation von Prozessdaten zwischen zwei Standard-SPSen gezeigt. Eine in IEC 61131-3 ST programmierte Soft-PLL reduziert den Jitter und berücksichtigt die gemittelte Latenz in der Empfänger-SPS. In Abbildung 58 wird der FB für die Soft-PLL für die Prozessdaten-Synchronisation mit DDS dargestellt. Mit *tx_cycle_time_ns* wird die Zykluszeit der Sender-SPS in Nanosekunden konfiguriert. Der Zeiger *ptrDDSdata* auf den Shared Memory der Interprozesskommunikation ermöglicht den Zugriff auf die empfangenen DDS-Daten wie auf den empfangenen Motorwinkel und den zugehörigen Zeitstempel. Die boolesche Variable *pll_locked* zeigt an, dass Sender und Empfänger synchronisiert sind, wodurch ein Jitter von wenigen µs keine Auswirkungen mehr hat. Der synchronisierte und geschätzte Winkel *angle* mit dem ein von einer Modulsteuerung gesteuerter Roboter arbeitet, lässt sich dann mit der geschätzten Latenz berechnen.

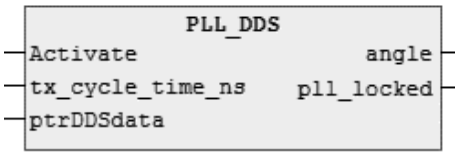


Abbildung 58: Soft-PLL FB für die Synchronisation von Prozessdaten zwischen zwei SPSen.

Um die Methoden zur Prozessdatensynchronisation zwischen zwei Steuerungen zu validieren, wird für einen Testaufbau jeweils eine analoge Ausgangsklemme von Beckhoff [139] als EtherCAT-Slave an beide SPS angeschlossen. Zur Visualisierung der analogen Signale beider SPSen wird ein Oszilloskop eingesetzt. Das Oszilloskop zeigt den abgetasteten Winkel in der 1. SPS und den berechneten Winkel in der 2. SPS. Abbildung 59 zeigt den abgetasteten Winkel des Förderbands (orange) und den synchronisierten geschätzten Winkel (türkis). Die beiden Signale werden aus Übersichtsgründen auf dem Oszilloskop mit einem Offset dargestellt.

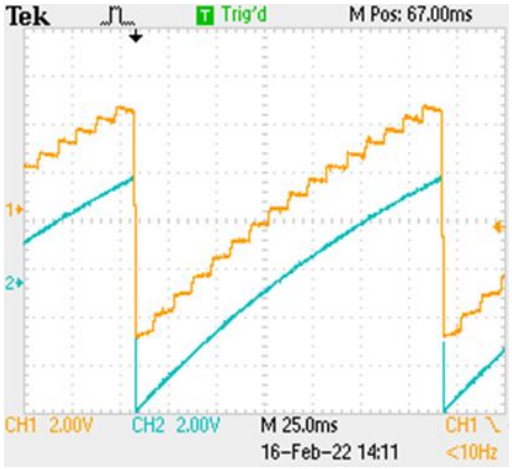


Abbildung 59: Channel 1 in orange: abgetasteter Winkel von SPS1. Channel 2 in türkis: synchronisierter Winkel in SPS2.

Die Ergebnisse zeigen, dass die Soft-PLL auf Basis von präzisen Zeitstempeln eine effiziente Synchronisation von SPSen ermöglicht. Ein Jitter von wenigen µs und die Latenz bei der Datenübertragung über DDS werden bei der zyklischen Ausführung der Soft-PLL berücksichtigt. TSN, PTP oder andere komplexe und teure Methoden zur Zeitsynchronisation werden nur noch dann benötigt, wenn eine mikrosekundengenaue Synchronisation erforderlich ist.

7.5 Interner Prozessdaten-Austausch

Die EVL ist eine Liste von Variablen, die sich im nicht sicherheitsbezogenen Speicherbereich der Sicherheitsanwendung befindet. Nur Daten im nicht sicherheitsbezogenen Speicherbereich können mit der Standardanwendung ausgetauscht werden, da der Zugriff auf den sicherheitsbezogenen Speicherbereich geschützt ist und demzufolge nicht im Shared Memory liegt. Das Konzept für den Datenaustausch wurde bereits im Kapitel 5.3.3 beschrieben und wird hier umgesetzt. Die EVL ist in der Abbildung 60 zu sehen und enthält die SPDUs.

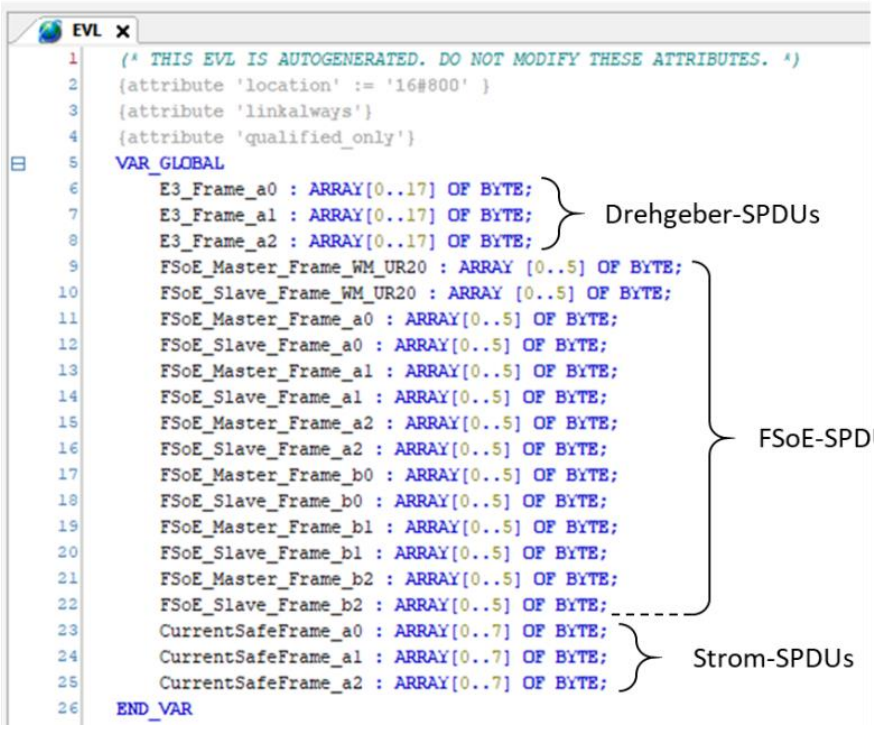


Abbildung 60: EVL für die Übertragung von sicherheitsbezogenen Prozessdaten.

Pro Achse gibt es eine SPDU für die Drehgeber-Daten und eine SPDU für die Strom-Daten. Zusätzlich sind zwei FSoE-SPDUs pro Kanal innerhalb einer Achse vorgesehen. Die Drehgeber- und Strom-Daten werden von der zentralen Sicherheits-SPS nur gelesen. Die FSoE-Kommunikation basiert auf einer bidirektionalen Kommunikation mit zwei getrennten Kanälen, wobei pro Achse zwei Slave- und zwei Master-SPDUs verwendet werden. Wird eine sicherheitsbezogene Kommunikation zwischen zwei Modulsteuerungen implementiert, werden die OPC UA Safety SPDUs ebenfalls in der EVL definiert.

7.6 Sicherheitsbezogene Kommunikation

Die SCLs der jeweiligen Sicherheitsprotokolle sind in der vorliegenden Arbeit als FBs in der Programmiersprache ST programmiert und werden zyklisch von der Sicherheitsanwendung ausgeführt. Es gibt weiterhin die Möglichkeit, die SCLs als Embedded-Komponente in dem Laufzeitsystem in C/C++ zu implementieren. Dies ist jedoch deutlich aufwendiger und soll deshalb hier nicht weiter betrachtet werden. Der SCL als FB ist für den Anwender übersichtlich, da die FBs im Sicherheitsprogramm grafisch miteinander verschaltet werden können. Abbildung 61 zeigt die drei FBs für die Drehgeber-SPDUs für die verwendete digitale Schnittstelle EnDat 3. Die FBs für FSoE und für den Strom-SCL sind dem Anhang zu entnehmen.

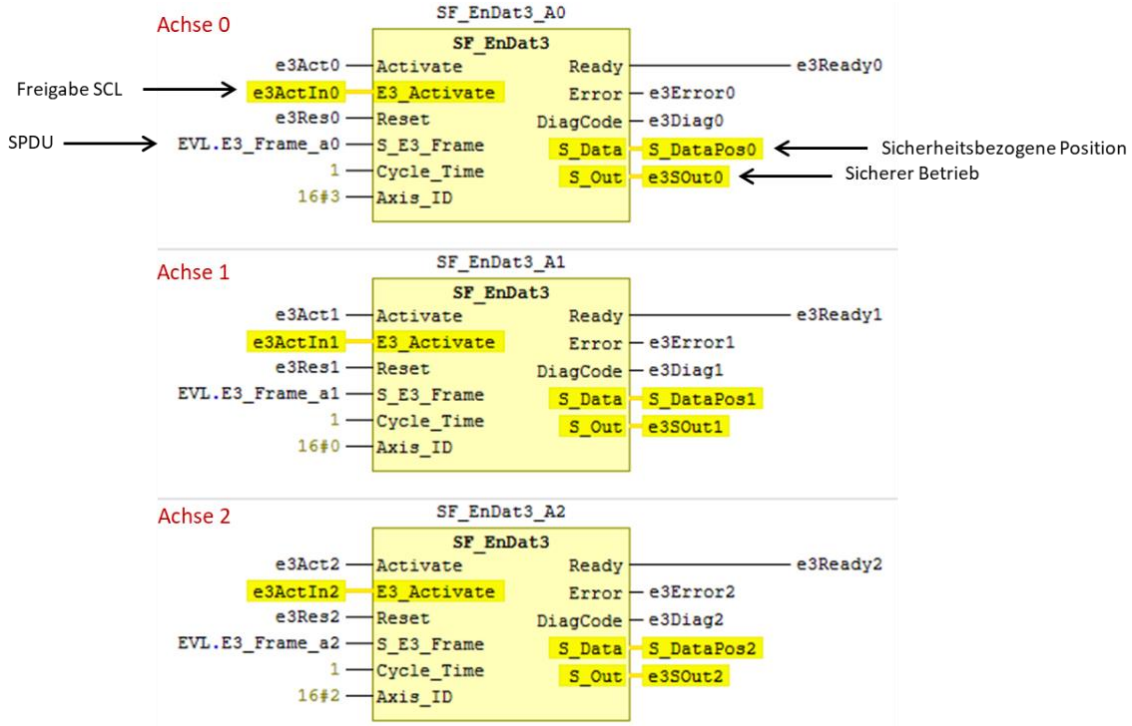


Abbildung 61: Drehgeber-SCL als FBs.

Über den Eingang *E3_Activate* kann die Auswertung der Datenübertragung sicher angefordert werden. Die FBs lesen die SPDUs zyklisch aus der EVL (*EVL.E3_Frame_aX*). Der SCL hat die Aufgabe die Datenübertragung gegen Fehler auszuwerten, die sicherheitsbezogenen Informationen zu dekodieren und diese der Sicherheitsanwendung bereitzustellen.

Über den Ausgang *S_Out* wird angezeigt, ob die Datenübertragung fehlerfrei oder fehlerhaft ist. Ist der Ausgang TRUE, wird damit signalisiert, dass der SCL im sicheren Betriebszustand ist. Die dekodierten sicherheitsbezogenen Daten werden über den Ausgang *S_Data* ausgegeben und in die sicherheitsbezogene Variable *S_DataPosX* in den sicherheitsbezogenen Speicherbereich kopiert. In dem vorliegenden Beispiel ist das die sicherheitsbezogene Position des jeweiligen Motors des Tripod-Roboters.

7.7 Mehrachs-Sicherheitsfunktionen

7.7.1 Überblick

Nachfolgend wird beschrieben, wie bewegungsüberwachende Sicherheitsfunktionen für ein Mehrachs-System realisiert werden können. Die Voraussetzung für die Umsetzung der zentralen Sicherheitsfunktionen für ein Mehrachs-System ist eine hohe sicherheitsbezogene Rechenleistung mit zertifizierter FPU. Im Vergleich zu Einzelachs-Sicherheitsfunktionen benötigen die Mehrachs-Sicherheitsfunktionen überproportional mehr Code. Darüber hinaus werden für die kinematischen Transformationen trigonometrische Funktionen wie Sinus und Cosinus verwendet. Abbildung 62 zeigt das Blockschaltbild für die Implementierung der Sicherheitsfunktionen SLP, SLS und SLA für einen Tripod-Roboter. Die implementierten Sicherheitsfunktionen werden aktiviert, wenn der Lichtvorhang ein Eindringen in den Arbeitsbereich des Roboters signalisiert.

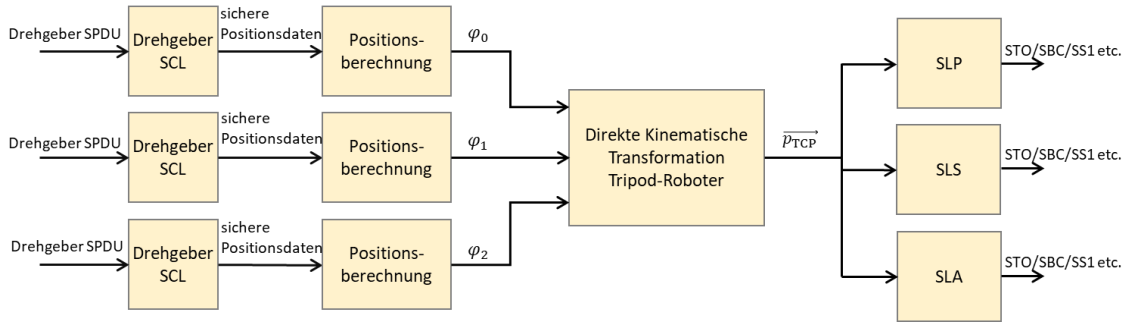


Abbildung 62: Blockschaltbild für Sicherheitsfunktionen die eine Bewegung von Mehrachs-Systemen überwachen.

7.7.2 Umrechnung in Benutzereinheiten

Die sicherheitsbezogenen Winkel werden über die Drehgeber-SPDUs (EnDat 3) zur Sicherheitsanwendung übertragen. Der Positionswert ist nicht in Benutzereinheiten (Grad, Radiant) dargestellt und wird über einen weiteren Software-Baustein umgerechnet. Für die Umrechnung gibt der Hersteller die Anzahl an Single-Turn und Multi-Turn Bits der Position an und der Anwender berücksichtigt zusätzlich das Übersetzungsverhältnis des Getriebes. Der Umrechnungs-FB *Safe_Position_Generation*, dargestellt in Abbildung 63, ist ebenso in der Lage, die Geschwindigkeit der Achsen zu bestimmen. Diesbezüglich berücksichtigt er die Zykluszeit um die Position numerisch zu differenzieren.

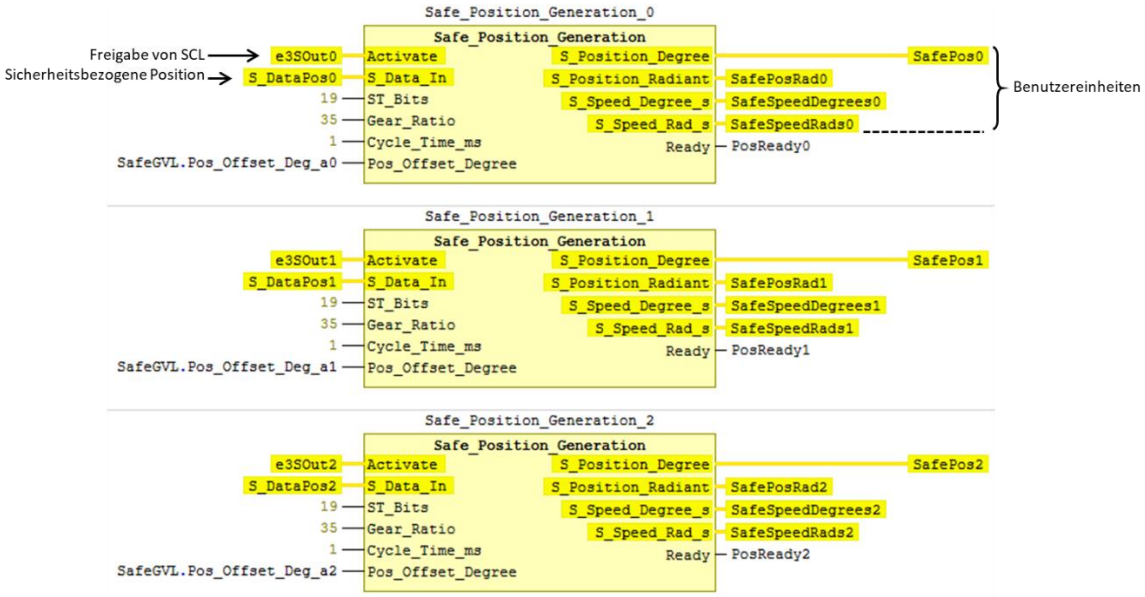


Abbildung 63: FB für die Umrechnung der Position in Benutzereinheiten.

7.7.3 Direkte Kinematik

Die direkte kinematische Transformation berechnet aus den Gelenkwinkeln des Tripod-Roboters die Position des TCPs. Abbildung 64 zeigt den FB für die direkte Transformation. Die Berechnung basiert auf einer geometrischen Herangehensweise und nutzt trigonometrische Funktionen wie Sinus und Cosinus. Die Mechanik des verwendeten Roboters wird parametriert, um die direkte kinematische Transformation für das verwendete Modell zu bestimmen. Durch die direkte Transformation wird die dreidimensionale Orientierung der relevanten und zu überwachenden Punkte des Roboters bestimmt. Dies sind die gefährlichsten und schnellsten Teile des Roboters: die POIs, d. h. die Ellenbogen des Tripods, und der TCP.

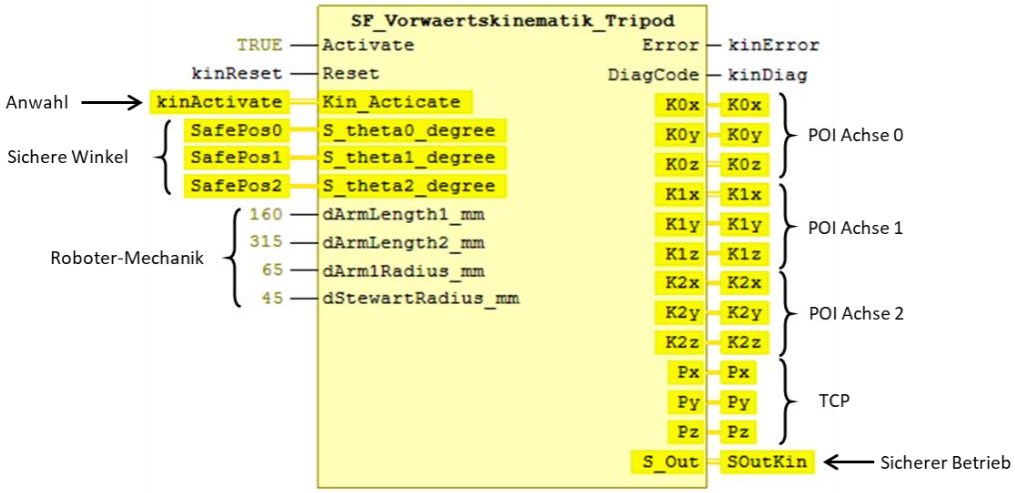


Abbildung 64: FB für die Vorwärtskinematik eines Tripod-Roboters.

Die drei POIs und der TCP des Roboters werden anschließend sicherheitstechnisch mithilfe von zentralen Sicherheitsfunktionen wie SLP, SLS und SLA überwacht.

7.7.4 Sicher begrenzte Position

Um Arbeitsbereiche entsprechend der den Körperregionen zugeordneten Belastungsgrenzen definieren und abgrenzen zu können, wird in der Regel eine sicher überwachte Position für die MRK gefordert [113]. Wird die Sicherheitsfunktion SLP auf ein Mehrachs-System wie einen Roboter übertragen, werden die gefährlichen beweglichen Teile des Roboters sicher in festgelegten Grenzen überwacht. Die Begrenzung der Positionsbereiche kann dabei im dreidimensionalen Raum durch Ebenen bestimmt werden. Der Einfachheit halber wird beim Demonstrator eine Begrenzung in x-, y- und z-Richtung definiert. Nach Verlassen des freigegebenen Bereichs, wird eine geeignete Stoppreaktion wie SS1, SS2, STO oder SBC ausgelöst. Die Sicherheitsfunktion SLP kann auch einen Schritt weitergehen, indem sie den Bremsweg berücksichtigt, damit der Roboter vor den festgelegten Grenzen zum Stillstand kommt. Damit wird sichergestellt, dass der Roboter den sicheren Bereich nicht verlässt und keine Gefährdung darstellt, solange der Mensch diesen Bereich nicht betritt. Für diese Art der Implementierung von SLP ist eine sehr leistungsfähige sicherheitsbezogene Steuerung notwendig, um die Bremsprofile berechnen zu können. Für jeden beweglichen Teil des Roboters, der sicherheitsbezogen überwacht werden soll, wird eine Instanz von SLP implementiert (siehe Abbildung 65).

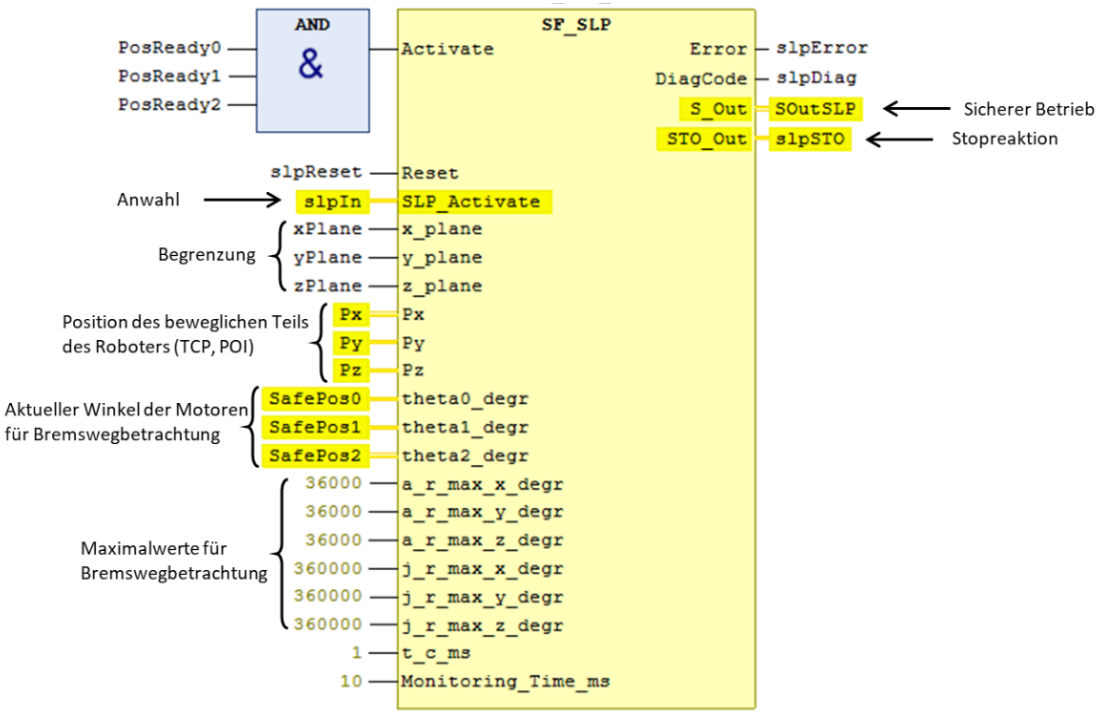


Abbildung 65: SLP als FB für ein Mehrachs-System.

7.7.5 Sicher begrenzte Geschwindigkeit

Die Sicherheitsfunktion SLS für eine einzelne Achse überwacht die Drehzahl innerhalb festgelegter Grenzen. Durch die Aktivierung über einen sicherheitsbezogenen Eingang erlaubt SLS zunächst die möglicherweise zu hohe Drehzahl innerhalb einer festgelegten Zeit zu reduzieren. Nach Ablauf der festgelegten Zeit (Monitoring-Time) wird die Drehzahlüberwachung aktiviert. Sollte die Drehzahl die vorgegebene Maximaldrehzahl überschreiten, wird eine geeignete Stoppreaktion ausgelöst. Bei SLS von Mehrachs-Systemen wird nicht die Drehzahl, sondern ein Geschwindigkeitsvektor im dreidimensionalen Raum überwacht. Es wird der Betrag des Geschwindigkeitsvektors mit der vorgegeben Maximalgeschwindigkeit verglichen. Um die translatorische Geschwindigkeit bestimmen zu können, wird die dreidimensionale Position des gefährlichen Roboterelements numerisch differenziert. Für jedes zu überwachende bewegliche Roboterelement wird eine Instanz des SLS-FBs aufgerufen (siehe Abbildung 66).

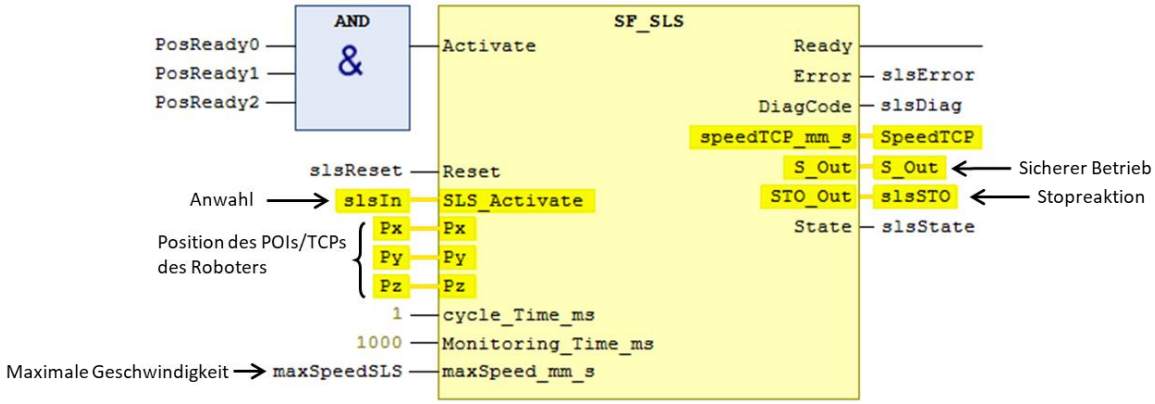


Abbildung 66: SLS als FB für ein Mehrachs-System.

7.7.6 Sicher begrenzte Beschleunigung

Die Sicherheitsfunktion SLA für ein Mehrachs-System wird auf die gleiche Weise wie die Sicherheitsfunktion SLS implementiert (siehe Abbildung 67). Bei SLA wird aus der dreidimensionalen Ausrichtung des Roboters zunächst durch zweifache numerische Differentiation die translatorische Beschleunigung bestimmt und anschließend mit einem Grenzwert verglichen. Bei Überschreiten des Grenzwerts wird ebenfalls eine geeignete Stoppfunktion aufgerufen.

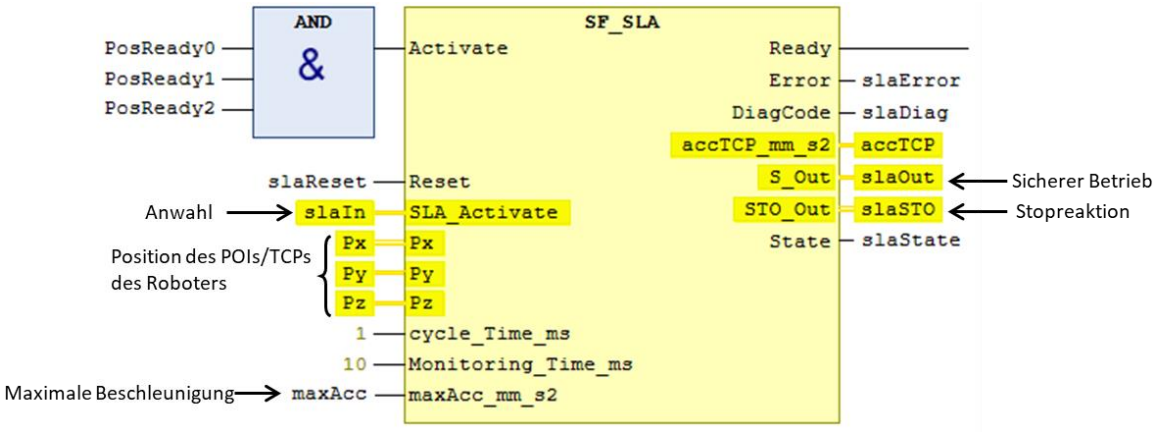


Abbildung 67: SLA als FB für ein Mehrachs-System.

7.7.7 Sicher begrenztes Drehmoment und Testsignale

Für die sichere Kollaboration nach Schutzprinzip PFL können sicherheitsbezogene Ströme in der Sicherheits-SPS verwendet werden, um das sicherheitsbezogene Drehmoment der einzelnen Achsen zu bestimmen. Die Ströme werden nach dem Gray-Channel-Prinzip über EtherCAT zur Sicherheits-SPS übertragen und gegen Übertragungsfehler ausgewertet. Die Messwerte werden in der SPS mit der Trace-Funktion aufgezeichnet. In der Abbildung 68 sind vier Messungen dargestellt. Im Folgenden werden die Diagramme von oben nach unten beschrieben.

- Dieses Diagramm zeigt die Testsignale, die über FSoE an die Antriebe gesendeten werden. Die Testsignale bzw. Testbits werden jede Sekunde gesetzt, um die Strommessung im Antrieb zu testen. Die Testsignale werden ebenfalls für die Funktionsprüfung der Abschaltwege (STO) und für die Bremsansteuerung (SBC) verwendet.
- Dieses Diagramm stellt die Ströme i_u, i_v, i_w eines Motors dar. Die Strommessung wird durch die Testsignale dynamisiert, indem der Bitstream der Dezimierungsfilter nacheinander kurzzeitig ausgeschaltet wird.
- Die Stromkomponenten i_α und i_β werden mit der Clarke-Transformation bestimmt. Mit i_α, i_β, dem sicherheitsbezogenen Kommutierungswinkel und der Park-Transformation wird die drehmomentbildende Stromkomponente i_q berechnet.
- Mit i_q und unter Berücksichtigung bestimmter Toleranzen (Fertigung, Temperatur und Rotor) kann das Drehmoment m eines PMAC-Synchronmotors bestimmt werden.

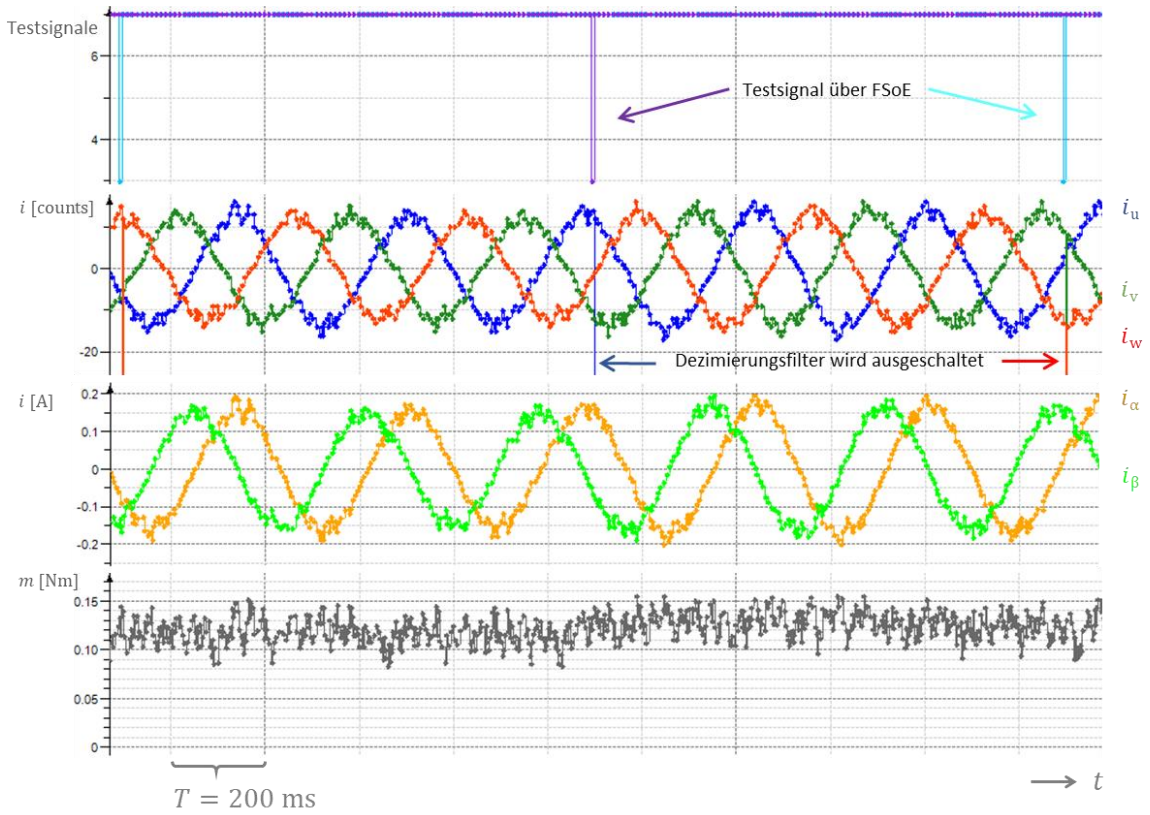


Abbildung 68: Testsignale zur Dynamisierung der Dezimierungsfilter, Motorströme und das sicherheitsbezogene Drehmoment.

Diese vorgestellte Implementierung wird mit FBs in der Programmiersprache ST realisiert. Ein FB als Strom-SCL überprüft die gray-channel-basierte Stromübertragung gegen Fehler und extrahiert die Ströme aus der SPDU. Ein SLT-FB berechnet aus den Strömen das aktuelle Drehmoment des jeweiligen Motors und vergleicht es mit einem zulässigen Grenzwert. Bei Überschreitung wird wie bei den Bewegungsüberwachungsfunktionen eine geeignete Reaktionsfunktion aufgerufen.

7.7.8 Anhaltezeit für eine Geschwindigkeitsüberwachung

Die Anhaltezeit ist die Zeit, die eine Maschine nach dem Auslösen des Stoppsignals benötigt, um zum Stillstand zu kommen. In Kapitel 3.1.2 wurden zunächst die Reaktionszeiten für das Auslösen von Reaktionsfunktionen bei einer Geschwindigkeitsüberwachung hergeleitet. Bei dem Ansatz, die Sicherheitsfunktionen in der übergeordneten Sicherheits-SPS auszuführen, ist die Zykluszeit der Sicherheitsapplikation und die Zykluszeit des Sicherheitsprotokolls von hoher Relevanz. Darüber hinaus ist die Totzeit für die Bestimmung der Geschwindigkeit durch numerische Differentiation des Winkels für kollaborative Anwendungen von Bedeutung. Die Motoren

des Roboters wurden im Laborbetrieb mit 48 V und einer Strombegrenzung von 1 A betrieben. Dadurch wird eine zulässige maximale Beschleunigung des Roboter-TCPs von ca.

$$a_{\mathrm{TCP,max}} \approx 20\,\frac{\mathrm{m}}{\mathrm{s}^2} \tag{30}$$

erreicht.

Die Abbildung 69 zeigt vier Diagramme zur Bestimmung der Anhaltezeit des verwendeten Robotersystems.

- In blau ist die TCP-Geschwindigkeit des Tripod-Roboters dargestellt, die aus den sicherheitsbezogenen Gelenkwinkeln des Roboters und der direkten kinematischen Transformation berechnet wird. Die Punkte zeigen die Abtastzeitpunkte mit der eingestellten Zykluszeit T_{SSPS} von 1 ms. In schwarz ist die auf 100 mm/s begrenzte Grenzgeschwindigkeit für die Sicherheitsfunktion SLS dargestellt. Beim Überschreiten der Geschwindigkeit werden in diesem Beispiel die Reaktionsfunktionen STO und SBC im Antriebssystem angesteuert.
- Das zweite Diagramm zeigt das STO/SBC-Signal mit negierter Logik. Die negierte Logik wird in der funktionalen Sicherheit für die Drahtbrucherkennung angewendet. Das STO/SBC-Signal fällt von logisch 1 auf logisch 0 sobald eine Geschwindigkeitsüberschreitung erkannt wird.
- Die sicherheitsbezogene Kommunikation zwischen Sicherheits-SPS und den Antrieben basiert beim vorgestellten Demonstrator auf FSoE. Das FSoE-Protokoll nutzt die doppelte Zykluszeit der Sicherheitsanwendung und des unterlagerten Feldbusses EtherCAT. Dieses Diagramm zeigt den Fall, dass FSoE erst einen Zyklus später das STO/SBC-Signal zu den Antrieben sendet.
- Im vierten Diagramm werden die Ströme eines Motors gezeigt. Aus Gründen der Übersichtlichkeit ist nur ein Motor dargestellt. Das STO/SBC-Signal schaltet jedoch alle drei Motoren zur gleichen Zeit aus, weshalb das Abschaltverhalten aller drei Motoren des Tripods äquivalent ist.

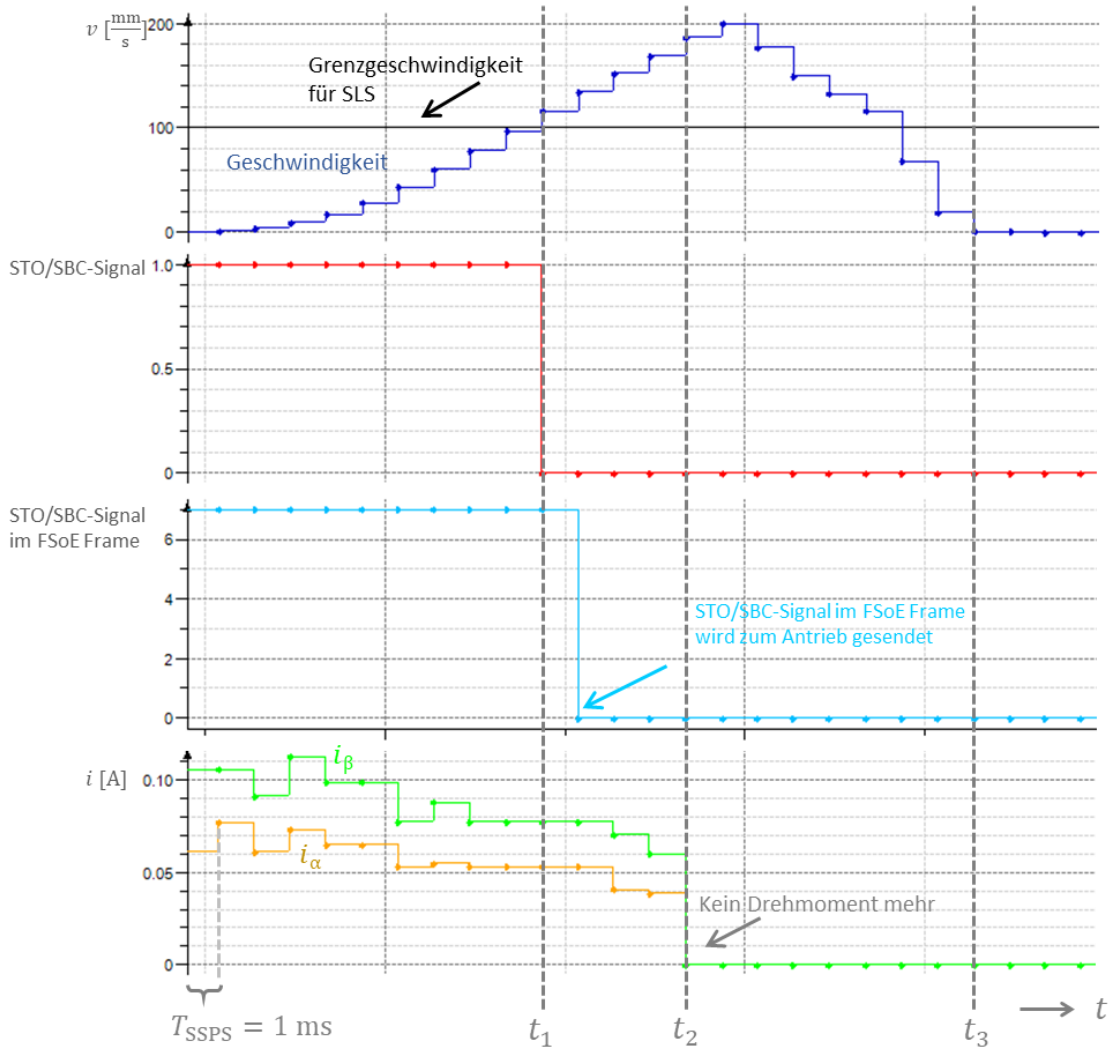


Abbildung 69: Bestimmung der Anhaltezeit bei einer Geschwindigkeitsüberschreitung von einem Roboter.

Zum Zeitpunkt t_1 wird eine Überschreitung der Geschwindigkeit in der Sicherheitsanwendung der Sicherheits-SPS erkannt. Dabei ist zu berücksichtigen, dass die eigentliche Geschwindigkeitsüberschreitung aufgrund der hergeleiteten Reaktionszeit (siehe Kapitel 3.1.2) bereits zeitlich vorher stattgefunden hat. Ein Zyklus nach t_1 wird das STO/SBC-Signal zu den Antrieben gesendet. Anschließend werden die Motoren stromlos geschaltet und die Bremse fällt ein. Die Ströme durch die Motorwicklungen sinken dann vergleichsweise schnell. In der übergeordneten Sicherheits-SPS sind zum Zeitpunkt t_2, also 4 ms nach dem Erkennen der Geschwindigkeitsüberschreitung, die Ströme und damit das Drehmoment in den Motoren Null. Der Roboter kommt zum Zeitpunkt t_2 jedoch noch nicht zum Stillstand, da der Abbau der Magnetfelder der elektromagnetischen Haltebremsen Zeit in Anspruch nimmt.

Der Stillstand des Roboters wird zum Zeitpunkt t_3, also 12 ms nach dem Erkennen der Geschwindigkeitsüberschreitung, in der Sicherheits-SPS festgestellt. Die Messungen zeigen, dass mit dem vorgestellten Ansatz der zentralen Sicherheits-SPS das Robotersystem bei maximal zulässiger Beschleunigung zwar den doppelten Wert der zulässigen reduzierten Geschwindigkeit erreicht, der drehmomentlose Zustand aber vergleichsweise schnell erreicht wird. Ist ein noch schnellerer Stillstand des Roboters erforderlich, kann eine zusätzliche sichere Begrenzung der Motorströme die maximal mögliche Beschleunigung reduzieren.

Unter Berücksichtigung des Einflusses der Zykluszeiten der sicherheitsbezogenen Protokolle und der Sicherheitsanwendung zeigt dieser Ansatz, dass eine hohe sicherheitstechnische Rechenleistung mit Gleitkomma-Arithmetik erforderlich ist, um die Maschine mit einer schnellen Reaktions- und Anhaltezeit in den sicheren Zustand zu bringen. Die heute üblichen Sicherheits-SPSen mit Zykluszeiten von 10-20 ms sind für kollaborative und kooperative Anwendungen weniger geeignet, da die Reaktions- und Anhaltezeit der Maschine eine Gefahr für den Menschen darstellt. Dies hat zur Folge, dass ausreichend große Abstände zwischen Maschine und Mensch definiert werden müssen, was wiederum nicht dem Grundgedanken der MRK entspricht.

7.8 Sicherheitsbezogene Rechenleistung

Das hier vorgestellte Sicherheitsprogramm mit sicherheitsbezogenen Kommunikationsprotokollen und einer Bewegungsüberwachung mit komplexen mathematischen Operationen besteht aus den folgenden IEC 61131-3 Softwaremodulen:

- Auswertung des black-channel-basierten FSoE-Protokolls gegen Übertragungsfehler (FSoE-SCL).
- Auswertung der gray-channel-basierten sicherheitsbezogenen digitalen Drehgeberschnittstelle EnDat 3 gegen Übertragungsfehler (Endat 3-SCL).
- Plausibilisierung der Drehgeber-Daten mit Kreuzvergleich.
- Auswertung der gray-channel-basierten Strom-Übertragung (Strom-SCL).
- Plausibilisierung der sicherheitsbezogenen Stromwerte mit Kreuzvergleich und 1. Kirchhoffschem Gesetz.
- Umrechnung von Winkel- und Stromwerten in Benutzereinheiten.

- Sicherheitsfunktionen, die die Bewegung (Position, Geschwindigkeit und Beschleunigung) der relevanten Teile des Roboters (drei POIs und TCP) überwachen.
- Sicherheitsbezogene Strom- und Drehmoment-Überwachung aller drei Antriebssysteme.
- Ansteuerung der Reaktionsfunktionen STO und SBC in den Antriebssystemen.
- Zyklische Testung und Auswertung der Abschaltpfade (STO), der Bremse (SBC), der Komponenten für die Strommessung (für SLT) und der Drehgeber (für SLP, SLS, SLA).

In der CODESYS Entwicklungsumgebung ist es möglich, das Zeitverhalten der ausgeführten Tasks zu überwachen (siehe Abbildung 70). Das gesamte Sicherheitsprogramm wird in einer durchschnittlichen Ausführungszeit von 22 µs ausgeführt. Bei fast 2,5 Millionen Aufrufen der Sicherheitsanwendung beträgt die maximale Ausführungszeit 31 µs.

Taskkonfiguration

Überwachung | Variablenverwendung | Eigenschaften

Task	Status	IEC-Zyklusanzahl	Zyklusanzahl	Konfigurie...	Letzte Zykluszeit ...	Durchschnittliche Zykl...	Max. Zykluszeit (µs)	Min. Zykluszeit (µs)
SafeTask	Gültig	0	2489228	n/a	22	22	31	19

Abbildung 70: Ausführungszeit der Sicherheitsanwendung.

Das Sicherheitskonzept des verwendeten Intel x6427FE Multi-Core SoCs basiert auf der Implementierung einer Prozessor-STL und einem Software-Kreuzvergleich. Die Abbildung 71 zeigt eine Laufzeitmessung für die Diagnosemessungen auf dem Prozessorkern, der auch die Kommunikation mit der externen TE durchführt. Es ist zu sehen, dass bei 5000 Messungen die längste Ausführungszeit bei ca. 330 µs liegt und die zyklische Ausführung innerhalb 1 ms oder schneller eingehalten werden kann.

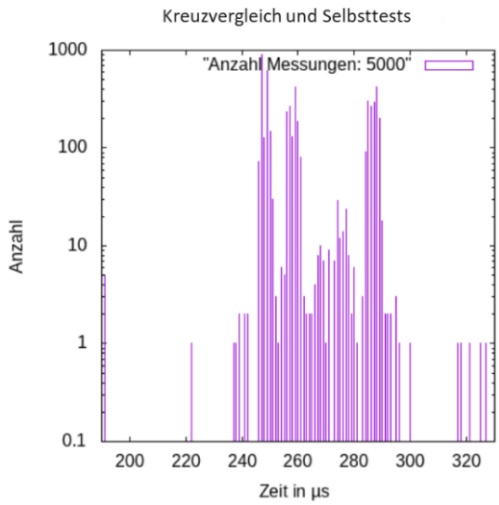


Abbildung 71: Laufzeitmessung der Diagnosemaßnahmen auf dem Intel x6427FE SoC.

8 Fazit

Funktionale Sicherheit ist heute ein integraler Bestandteil von Automatisierungslösungen. Häufig besteht jedoch noch eine klare Trennung zwischen dem sicherheitsbezogenen Teil und dem funktionalen Anwendungsteil von Maschinen oder Anlagen. Diese Trennung führt dazu, dass Sicherheitsaspekte erst am Ende der Entwicklung berücksichtigt werden und nicht von Anfang an in die Gesamtsystemphilosophie integriert sind. Dies ist jedoch für die Gesamtsicherheit nicht förderlich.

Mit den Bestrebungen zu Industrie 5.0 und dem damit verbundenen menschenzentrierten Ansatz wird die Zusammenarbeit von Menschen und autonomen Maschinen und in diesem Zusammenhang die funktionale Sicherheit immer mehr an Bedeutung gewinnen. Roboter, AMRs und FTFs sollen ohne menschliches Eingreifen weiterhin effizient mit hoher Geschwindigkeit und Traglast arbeiten. Wenn sich jedoch Menschen in der Nähe befinden, müssen sie mit zusätzlichen Sicherheitsmaßnahmen, wie einer Geschwindigkeits- und Kraftbegrenzung, betrieben werden und somit genauso ungefährlich wie Cobots sein. Die Anforderungen an eine zentrale Sicherheits-SPS für die Kollaboration sind hoch, da eine schnelle Verarbeitung komplexer kinematischer Transformationen mit Gleitkomma-Arithmetik und schnelle Reaktionszeiten erforderlich sind.

Sicherheitsmethoden wurden lange Zeit in verschiedenen Branchen unabhängig voneinander entwickelt. Die jeweiligen Sicherheitsnormen der verschiedenen Branchen bauen alle auf der IEC 61508 auf und verfolgen damit ähnliche Ziele. Diese Synergien zwischen den Normen haben zur Folge, dass Komponenten, die für den Automobilsektor entwickelt wurden, in Verbindung mit entsprechender Dokumentation den Entwicklungs- und Zertifizierungsaufwand für die funktionale Sicherheit auch in Anwendungsbereichen wie der Fertigungs- und Maschinenautomatisierung reduzieren können. Der Anteil an sicherheitsrelevanter Software in der Automatisierung nimmt stetig zu, auch wenn er noch nicht das Niveau von Automobilanwendungen erreicht hat. Die schnelle Verbreitung von Robotern, Cobots, FTFs und AMRs in der Automatisierung trägt jedoch dazu bei, diesen Prozess zu beschleunigen. Automobil- und Industrieanwendungen einschließlich ihrer Softwarealgorithmen nähern sich immer mehr an. Die Möglichkeit sicherheitsrelevante Halbleiter, wie Lockstep oder vergleichbare Prozessoren, und Softwarealgorithmen mit Gleitkomma-Arithmetik in beiden Bereichen einzusetzen, eröffnet insbesondere für Motion-Control-Anwendungen neue Möglichkeiten.

Erst die heutige Rechenleistung für funktional sichere Steuerungen, die insbesondere durch den aus dem Automobilbereich bekannten Lockstep-Ansatz auch für die

Fertigungs- und Maschinenautomatisierung zur Verfügung steht, sowie die schnellen ethernetbasierten Feldbussysteme einschließlich ihrer funktional sicheren Protokolle ermöglichen die sicherheitstechnische Bewertung ganzer Maschinen und vereinfachen damit die Umsetzung der MRK.

Die in dieser Arbeit vorgestellte Motion-Control-Architektur für Industrie 5.0 basiert auf einem IPC für gemischt-kritische Anwendungen. Die IPC-Systemarchitektur setzt auf Multi-Core SoCs mit Lockstep- oder vergleichbarer Eigenschaft, Hardwarevirtualisierung und sicherheitsbezogener Gleitkomma-Arithmetik. Diese Architektur ist prädestiniert für die kosten- und energieeffiziente gemischt-kritische Automatisierung. Die integrierte zentrale Sicherheits-SPS als eine Fortführung der IPC-basierten Automatisierungsphilosophie unterstützt die Sicherheitslösung mit zentraler Diagnose und ermöglicht die Berechnung komplexer Sicherheitsfunktionen für ganze Maschinen.

Eine hohe Flexibilität und Skalierbarkeit der Infrastruktur ist für die zunehmende Digitalisierung und Automatisierung in der Industrie erforderlich. Die Integration unterschiedlicher Anwendungen auf einer gemeinsamen Hardwareplattform wird durch die Virtualisierungstechnologie ermöglicht. Dies gewährleistet die geforderte Rückwirkungsfreiheit in der vorgestellten Systemarchitektur durch Kapselung der sicherheitsbezogenen und nicht sicherheitsbezogenen Anwendungen in verschiedene VMs. Soft-SPS-Erweiterungen und die einhergehende Programmierung in IEC 61131-3 Programmiersprachen bieten eine deutlich höhere Flexibilität als gängige Robotersteuerungen mit proprietären Programmiersprachen.

Der vorgestellte Ansatz der zentralen Sicherheits-SPS vereinfacht die Umsetzung der Schutzprinzipien und Kollaborationsarten für die MRK. Insbesondere Kollaborationsarten, bei denen ein direkter Kontakt zwischen Menschen und Roboter möglich sein soll, erfordern eine zentrale Bewertung physikalischer Größen wie Winkel, Geschwindigkeit und Strom bzw. Drehmoment der einzelnen Achsen eines Robotersystems. Mithilfe komplexer kinematischer Transformationen kann z. B. die Geschwindigkeit des Roboter-TCPs sicher überwacht werden. Der Ansatz der Zentralisierung von Sicherheitsfunktionen und die damit verbundene Gray-Channel-Übertragung von sicherheitsbezogenen Messwerten (Strom, Winkel) zur zentralen Sicherheits-SPS erhöht neben der Wirtschaftlichkeit der MRK auch die Flexibilität und die Verfügbarkeit der Maschinen. Mit dem vorgestellten Ansatz ist eine Aussage über die dreidimensionale Ausrichtung des Roboters im funktional sicheren Kontext ohne großen Aufwand und Kostenzusatz möglich. Statt dem klas-

sischen Schutzprinzip, bei dem sich der Mensch nur im Arbeitsraum des Roboters befinden darf, wenn dieser sicher stillsteht, können mit dem vorgestellten Konzept die anderen Schutzprinzipien umgesetzt werden.

Komplexität, Kosten und Aufwand sind wichtige Faktoren in der sicheren Fertigungs- und Maschinenautomatisierung. Die vorgestellte zentrale Sicherheits-SPS unterstützt die zentrale Diagnose und vereinfacht den Einsatz von nicht zertifizierten Standardkomponenten im funktional sicheren Kontext. Die zentrale Sicherheits-SPS führt Diagnosetests aus und ermöglicht neben der Einbindung kostengünstiger, diversitärer Standardkomponenten auch eine einfache Umsetzung des degradierten Betriebs. Sicherheitsbezogene Sensoren und Aktoren, die mit hohen Kosten verbunden sind, werden durch qualitätsgesicherte Standardkomponenten und Softwaremodule in der zentralen Sicherheits-SPS ersetzt. Durch die Ausführung von Sicherheitsfunktionen und der SCLs in der zentralen Sicherheits-SPS ändern sich somit auch die Anforderungen an die Entwicklung sicherheitsbezogener Software. Sicherheitsfunktionen, programmiert in ST, können als FBs zusammengefasst und im grafischen und übersichtlichen Sicherheitsprogramm miteinander verschaltet werden. Das Sicherheitsprogramm basiert auf LVL und unterliegt den Anforderungen an SRASW, was wiederum den Inbetriebnahmeprozess verkürzt und die Zertifizierungskosten niedrig hält. Die SPS- und Motion-Control-Funktionalitäten sowie die Programmierung der Sicherheitsanwendung bilden ein Baukastensystem für eine IEC 61131-3 Entwicklungsumgebung.

Um die Anforderungen neuer Paradigmen wie Industrie 4.0 und 5.0 zu erfüllen, erfordern Automatisierungssysteme einen hohen Vernetzungsgrad. Die bisher getrennten Ebenen der IT und OT werden zunehmend zusammengeführt. Neue Steuerungssysteme, die für die operative Prozessführung verantwortlich sind, verbinden die Feldebene der OT mit der IT-Welt. Als Teil der vorgestellten Motion-Control-Architektur für Industrie 5.0 wird eine effiziente Vernetzung von Automatisierungskomponenten auf Basis von OT- und IT-Kommunikationsmethoden gezeigt. Für die lokale Anbindung von Sensoren und Aktoren gibt es derzeit keinen erkennbaren Grund, die bewährten OT-Feldbusse mit synchroner Abtastung durch IT-Kommunikationsverfahren zu ersetzen. Mit schnellen Feldbussen und der damit verbundenen Black- und Gray-Channel-Kommunikation zur Übertragung sicherheitsrelevanter Daten an die zentrale Sicherheits-SPS lassen sich komplexe Sicherheitsfunktionen zur Überwachung ganzer Maschinen realisieren. Bei der C2C-Kommunikation überzeugt das PubSub-Modell durch einen ereignisgesteuerten Ansatz. Gerade bei komplexen Automatisierungsanlagen mit einer Vielzahl von vernetzten Teilnehmern ist es von Vorteil, dass bei der ereignisgesteuerten Kommunikation in der Regel deutlich we-

niger Prozessdaten pro Sekunde übertragen werden. Ebenso ist das Hinzufügen und Entfernen von Teilnehmern im laufenden Betrieb vorteilhaft, da dadurch die Verfügbarkeit der Produktionsanlagen erhöht wird.

Die erarbeitete IPC-Systemarchitektur wurde zum Nachweis der vorgestellten Konzepte implementiert. Ein Intel Atom x6427FE Quad-Core SoC mit On-Chip-Diagnosemaßnahmen und vergleichsweise hoher sicherheitsbezogener Rechenleistung auch für Gleitkomma-Operationen bildet die Basis der vorgestellten Systemarchitektur. Ein Typ 1 Hypervisor, der die Partitionierung von Anwendungen unterschiedlicher Kritikalität unterstützt, sowie die diversitären Echtzeit-Betriebssysteme für die beiden sicherheitsbezogenen Kanäle ermöglichen eine schlanke Implementierung für eine skalierbare gemischtkritische Steuerung. Die Compound-SPS für Maschinenmodule auf Basis der vorgestellten Systemarchitektur zeigt die Validierung der Konzepte für die Sicherheitslösung mit zentraler Diagnose, der MRK und effizienter Vernetzung mit OT- und IT-Kommunikationsmethoden in der Automatisierung. Der Nachweis der Funktionstüchtigkeit und der Leistungsfähigkeit des in dieser Arbeit entworfenen und realisierten Systems wurde anhand von Applikationen für einen Industrieroboter erbracht. Ein Delta-Parallel-Roboter (Tripod), der von der Compound-SPS sowohl gesteuert als auch sicher überwacht wird, stellt dabei den Nutzen der erarbeiteten Konzepte unter Beweis.

Die Arbeit mit den Betriebssystemen Zephyr OS und Linux hat insgesamt alle Erwartungen erfüllt. Diese Open-Source-Projekte zeichnen sich durch eine aktive und engagierte Community aus, die kontinuierlich zur Weiterentwicklung und Verbesserung der Systeme beiträgt. Die Sicherheitszertifizierung des Zephyr OS wird sogar von der Community vorangetrieben. Ein Grund für die stetige Weiterentwicklung ist sicherlich, dass beide Betriebssysteme sowohl auf x86- als auch ARM-Prozessorarchitekturen eingesetzt werden können.

Da die zertifizierte Version 1.4 des Open-Source-Hypervisors ACRN nicht alle notwendigen Eigenschaften zur Umsetzung der geplanten Systemarchitektur bietet, wurde die Systemarchitektur mit einer nicht zertifizierten Version umgesetzt. Eine Neuzertifizierung ist mit erheblichen Kosten verbunden und für ein einzelnes Unternehmen nicht wirtschaftlich. Wenn die Community die Zertifizierung vorantreibt, können Arbeitspakete und Kosten geteilt werden. Mit den wachsenden Anforderungen an Flexibilität und Modularität von Automatisierungskomponenten gewinnt die Virtualisierung zunehmend an Bedeutung und wird in Zukunft ein wichtiger Bestandteil von Automatisierungslösungen sein. Insbesondere die Möglichkeit, Anwendungen voneinander zu isolieren, bietet ein großes Potenzial nicht nur für Safety-, sondern auch für die immer wichtiger werdenden Security-

Anwendungen im industriellen Umfeld. Es ist daher davon auszugehen, dass Hypervisor- oder Separationskernel-Lösungen mit entsprechenden Sicherheitszertifizierungen für die Fertigungs- und Maschinenautomatisierung z. B. durch Open-Source-Projekte oder durch weit verbreitete Produkte mit hohen Stückzahlen zu angemessenen Kosten verfügbar sein werden.

Anhang

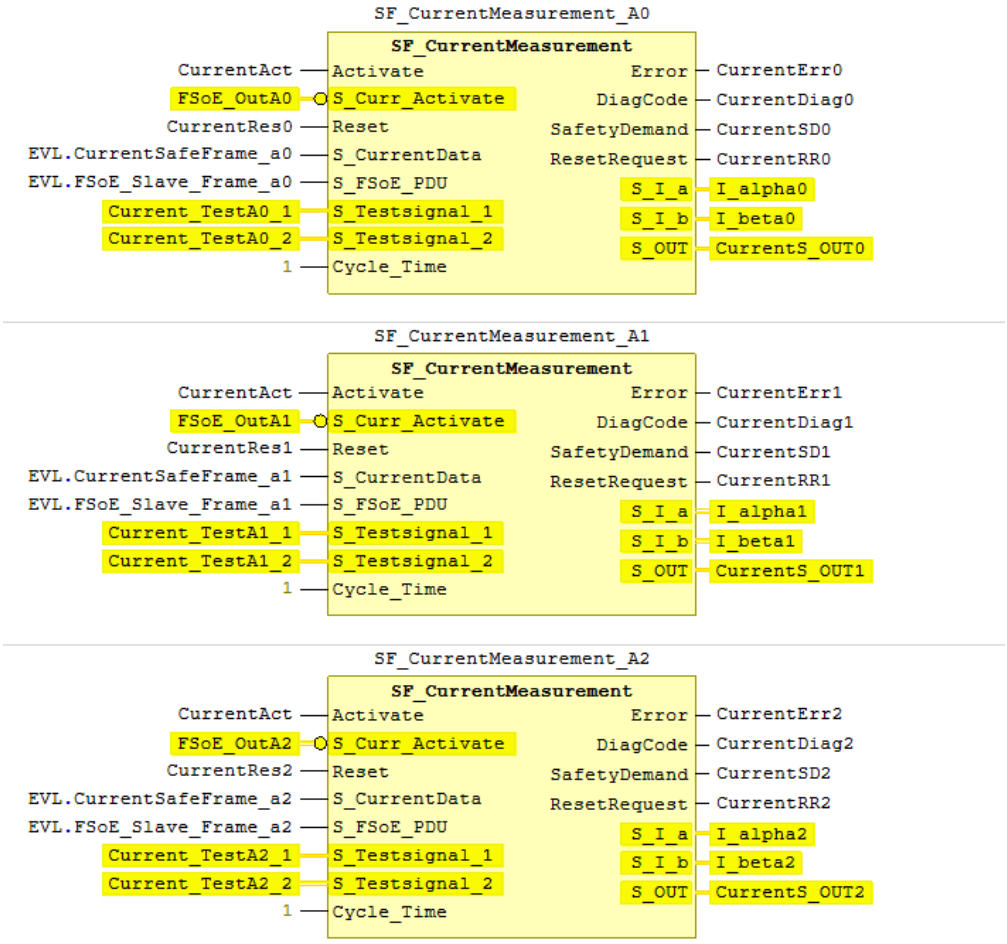


Abbildung 72: FBs für die Strommessung.

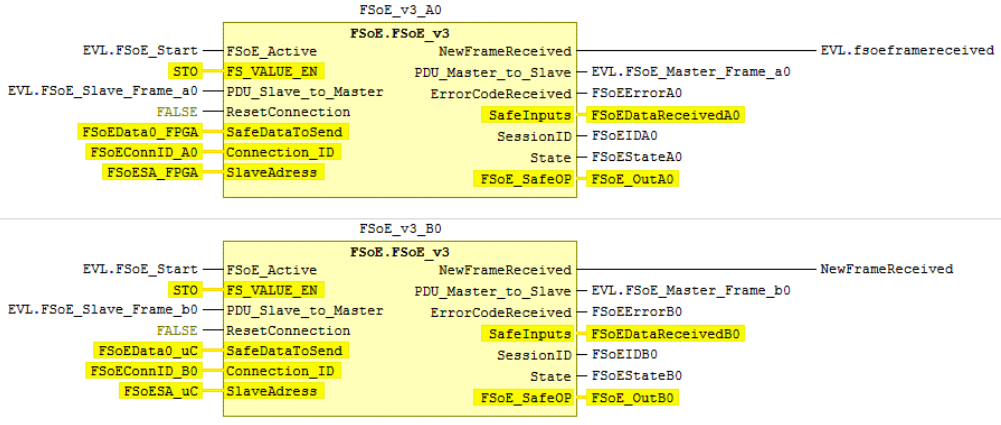


Abbildung 73: FBs für die zweikanalige FSoE-Kommunikation mit einem Antrieb.

Literaturverzeichnis

[1] E. Commission, D.-G. for Research, Innovation, M. Breque, L. De Nul und A. Petridis, Industry 5.0 : towards a sustainable, human-centric and resilient European industry, Publications Office, 2021.

[2] M. Azarmipour, Virtualisierung prozessnaher Steuerungen in der Prozessautomatisierung, VDI Verlag GmbH, 2022.

[3] A. Ustundag und E. Cevikcan, Industry 4.0: managing the digital transformation, Springer, 2017.

[4] E. O. B. Nara, M. B. da Costa, I. C. Baierle, J. L. Schaefer, G. B. Benitez, L. M. A. L. do Santos und L. B. Benitez, „Expected impact of industry 4.0 technologies on sustainable development: A study in the context of Brazil's plastic industry," *Sustainable Production and Consumption,* Bd. 25, pp. 102-122, 2021.

[5] X. T. R. Kong, H. Luo, G. Q. Huang und X. Yang, „Industrial wearable system: the human-centric empowering technology in Industry 4.0," *Journal of Intelligent Manufacturing,* Bd. 30, p. 2853–2869, 2019.

[6] A. Papetti, F. Gregori, M. Pandolfi, M. Peruzzini und M. Germani, „A method to improve workers' well-being toward human-centered connected factories," *J Comput Des Eng,* Bd. 7, p. 630–643, October 2020.

[7] Y. Lu, H. Zheng, S. Chand, W. Xia, Z. Liu, X. Xu, L. Wang, Z. Qin und J. Bao, „Outlook on human-centric manufacturing towards Industry 5.0," *Journal of Manufacturing Systems,* Bd. 62, p. 612–627, 2022.

[8] H. Nguyen Ngoc, G. Lasa und I. Iriarte, „Human-centred design in industry 4.0: case study review and opportunities for future research," *Journal of Intelligent Manufacturing,* Bd. 33, p. 35–76, 2022.

[9] K. Dohrmann, J. Toy, E. Pitcher, J. Selders und T. Grauf, „The Logistics Trend Radar 6.0," DHL Trend Research, Bonn, Germany, 2022.

[10] DIN EN 61800-5-2:2017-11, Elektrische Leistungsantriebssysteme mit einstellbarer Drehzahl - Teil 5-2: Anforderungen an die Sicherheit - Funktionale Sicherheit (IEC 61800-5-2:2016).

[11] DIN EN 61508:2011-02, Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme (IEC 61508:2010).

[12] ISO 26262-1:2018-12, Straßenfahrzeuge - Funktionale Sicherheit.

[13] ISO 13849-1:2023-04, Sicherheit von Maschinen - Sicherheitsbezogene Teile von Steuerungen - Teil 1: Allgemeine Gestaltungsleitsätze.

[14] R. Ungerer, „Use of Safety MCUs for Industrial and Automotive Applications," in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2019.

[15] G. Corradi, „Design single chip mixed criticality motion systems with ZYNQ Ultrascale+ SoC SIL3 HFT=1," in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2019.

[16] J. Karim, H. Mike und R. Bharat, „Optimized solutions for safe motion control applications," in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2019.

[17] DIN EN 61131-3:2014-06, Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen (IEC 61131-3:2013).

[18] P. Bilancia, J. Schmidt, R. Raffaeli, M. Peruzzini und M. Pellicciari, „An Overview of Industrial Robots Control and Programming Approaches," *Applied Sciences*, Bd. 13, p. 2582, 2023.

[19] R. Raffaeli, P. Bilancia, F. Neri, M. Peruzzini und M. Pellicciari, „Engineering Method and Tool for the Complete Virtual Commissioning of Robotic Cells," *Applied Sciences*, Bd. 12, p. 3164, 2022.

[20] DIN 66312-1:1993-06, Manipulating industrial robots; Industrial Robot Languages (IRL).

[21] B. Lüdemann-Ravit, „Ein System zur Automatisierung der Planung und Programmierung von industriellen Roboterapplikationen," 2005.

[22] PLCopen Technical Committees 2 - Task Force, „Function Blocks for motion control: Part 4 - Coordinated Motion," 2008.

[23] Bosch Rexroth AG, „Betriebsanleitung ctrlX CORE Steuerungen," 2022. [Online]. Available: https://docs.automation.boschrexroth.com/iirds/cdp-metadata.boschrexroth.de~iiDC~Product-ctrlX-CORE/. [Zugriff am 01 05 2023].

[24] SEW-EURODRIVE GmbH & Co KG, „MOVI-C CONTROLLER progressive UHX65A mit Feldbus-Schnittstelle PROFINET IO (mit PROFIsafe)," 02 2019. [Online]. Available: https://download.sew-eurodrive.com/download/pdf/25868721.pdf. [Zugriff am 01 05 2023].

[25] Kingstar, „White Paper 5 Real-Time, Ethernet-Based Fieldbuses Compared," 2016. [Online]. Available: https://kingstar.com/resources/get-fieldbuses-compared-paper/. [Zugriff am 02 04 2022].

[26] Beckhoff Automation GmbH & Co. KG, „EtherCAT System-Dokumentation Version: 5.6," Verl, Germany, 2022.

[27] T. Wilkening, J. O. Krah und H. Goergen, „Safety-Related Interfaces for Position Encoders-a Survey," in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2019.

[28] 5G-ACIA, „5G for connected Industries and Automation," ZVEI - German Electrical and Electronic Manufacturers' Association, 2019.

[29] VDMA und Fraunhofer ISS, „5G im Maschinen- und Anlagenbau - Leitfaden für die Integration von 5G in Produkt und Produktion," 2019.

[30] Verband der Automobilindustrie, Verein deutscher Maschinen und Anlagenbauer, „Schnittstelle zur Kommunikation zwischen Fahrerlosen Transportfahrzeugen (FTF) und einer Leitsteuerung - VDA 5050," Verband der Automobilindustrie, Berlin, Germany, 2022.

[31] Object Management Group, „OMG Data Distribution Service (DDS) Version 1.4,“ 04 2015. [Online]. Available: https://www.omg.org/spec/DDS/1.4. [Zugriff am 29 04 2023].

[32] DIN EN 61499-1:2014-09, Funktionsbausteine für industrielle Leitsysteme - Teil 1: Architektur (IEC 61499-1:2012).

[33] T. Wilkening, J. Randerath, M. Avendano, J. Holtz und J. O. Krah, „Modular System Architecture for Large Multi-Axis Motion Control Systems in Automation,“ in *PCIM Europe 2022; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2022.

[34] Richtlinie 98/37/EG des Europäischen Parlaments und des Rates vom 22. Juni zur Angleichung der Rechts- und Verwaltungsvorschriften der Mitgliedsstaaten für Maschinen, „ABl. EG (1998) Nr. L 207, S. 1-46; geänd. durch Richtlinie 98/79/EG - ABl. EG (1998) Nr. L 331, S. 1-37“.

[35] DIN EN ISO 12100:2011-03, Sicherheit von Maschinen - Allgemeine Gestaltungsleitsätze - Risikobeurteilung und Risikominderung (ISO 12100:2010).

[36] DIN EN 62061:2016-05, Sicherheit von Maschinen - Funktionale Sicherheit sicherheitsbezogener elektrischer, elektronischer und programmierbarer elektronischer Steuerungssysteme (IEC 62061:2005 + A1:2012 + A2:2015).

[37] M. Hauke, M. Schaefer, R. Apfeld, T. Bömer, M. Huelke, T. Borowski, K.-H. Büllesbach, M. Dorra, H.-G. Foermer-Schaefer, J. Uppenkamp, O. Lohmeier, K.-D. Heimann, B. Köhler, H. Zilligen, S. Otto, P. Rempel und G. Reuß, „Funktionale Sicherheit von Maschinensteuerungen - Anwendung der DIN EN ISO 13849. IFA Report 2/2017,“ Deutsche Gesetzliche Unfallversicherung e. V. (DGUV), Berlin, Germany, 2017.

[38] DIN EN IEC 62061:2023-02, Sicherheit von Maschinen - Funktionale Sicherheit sicherheitsbezogener Steuerungssysteme (IEC 62061:2021).

[39] Pilz GmbH & Co. KG, „PNOZ m B0 Operating Manual,“ Ostfildern, Germany, 2022.

[40] Siemens AG, „Programming Guideline for S7-1200/S7-1500,“ 2017.

[41] P. Wratil und M. Kieviet, Sicherheitstechnik für Komponenten und Systeme, VDE-Verlag, 2010.

[42] M. Mai und G. Reuss, „Selbsttests für Mikroprozessoren mit Sicherheitsaufgaben," Hauptverband der gewerblichen Berufsgenossenschaften (HVBG), Berufsgenossenschaftliches Institut für Arbeitsschutz - BGIA, Sankt Augustin, Germany, 2006.

[43] CODESYS GmbH, "CODESYS SAFETY," 2022. [Online]. Available: https://de.codesys.com/produkte/codesys-safety/safety-fuer-sil3-steuerungen.html. [Accessed 03 05 2023].

[44] A. Hayek, B. Machmur, M. Schreiber and J. Börcsök, "HICore1: "Safety on a Chip" Turnkey Solution," in *25th International Conference on Application-Specific Systems, Architectures and Processors*, Zurich, Switzerland, 2014.

[45] DGUV Test Prüf- und Zertifizierungssystem der Deutschen gesetzlichen Unfallversicherung, „Kann mit einer Standard-SPS PL c erreicht werden?," Sankt Augustin, Germany, 2022.

[46] PHOENIX CONTACT GmbH & Co. KG, „User manual - Installation and operation of the RFC 4072S Remote Field Controller with integrated safety-related PROFINET controller," Blomberg, Germany, 2022.

[47] Intel Corporation, „Case Study - Build a better robot, and automation gets easy," Santa Clara, USA, 2010.

[48] F. Reichenbach, J. Endresen und S.-E. Ellevseth, „Maximizing diversity in CPUs: Using GPUs as coprocessors to achieve safety integrity," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014.

[49] H. Gebuhr, „Functional Safety with failsafe Software-Controller by use of Coded Processing," in *SPS/IPC/DRIVES*, Nuremberg, Germany, 2009.

[50] SIListra Systems GmbH, „Hardware-abstrahierte Safety-Lösung: Erstmals auf der SPS Messe 2022," 04 10 2022. [Online]. Available: https://www.silistra-systems.com/documents/press_releases/SIListra-Systems-SPS-2022-de.pdf. [Zugriff am 05 01 2023].

[51] M. Fischer, O. Riedel und A. Lechler, „Arithmetic coding for floating-points and elementary mathematical functions,“ in *2021 5th International Conference on System Reliability and Safety (ICSRS)*, 2021.

[52] J. Börcsök, W. Müller, E. Hahn, M. Schwarz und M. Abdelawwad, „Safe-System-on-Chip for Functional Safety,“ in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2021.

[53] A. Hayek und J. Börcsök, „Safety chips in light of the standard IEC 61508: Survey and analysis,“ in *2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, 2014.

[54] Texas Instruments Incorporated, „User's Guide: Safety Manual for TMS570LS31x and TMS570LS21x Hercules™ ARM®-Based Safety Critical Microcontrollers,“ Dallas, USA, 2015.

[55] TÜV Süd Rail Gmbh, „Technical Report on the Concept Study of a Safety Architecture, Report no. TF85875T,“ Munich, Germany, 2014.

[56] G. Wenderlein und M. Wendt, „Chip Set for functional Safety in Embedded Control Units,“ in *SPS/IPC/DRIVES 2011*, 2011.

[57] Intel Corporation, „Datasheet, Volume 1: Intel Atom® x6000E Series, and Intel® Pentium® and Celeron® N and J Series Processors for IoT Applications,“ 03 2023. [Online]. Available: https://www.intel.com/content/www/us/en/content-details/636112/intel-atom-x6000e-series-and-intel-pentium-and-celeron-n-and-j-series-processors-for-iot-applications-datasheet-volume-1.html. [Zugriff am 03 05 2023].

[58] M. Salardi, E. Spano, M. Chiavacci und J. O. Krah, „Achieving Mixed Criticality and Cat. 3 PL = d on single System on Chip,“ in *Embedded World 2021 digital Exhibition & Conference*, 2021.

[59] Beckhoff Automation, „Operating instructions for AX5805 / AX5806 TwinSAFE drive option cards for the AX5000 servo drive,“ Verl, Germany, 2020.

[60] R. P. Weicker, „Dhrystone: a synthetic systems programming benchmark,“ *Communications of the ACM*, Bd. 27, p. 1013–1030, 1984.

[61] T. Wilkening, J. Holtz und J. O. Krah, „Mixed-Critical Control Architecture for Industry 5.0," in *4th International Conference on Smart Grid and Renewable Energy (SGRE)*, 2024.

[62] STMicroelectronics N.V., „Arm® Cortex®-M4 32-bit MCU+FPU Datasheet, STM32F446xC/E, DS10693 Rev 10," 01 2021. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f446re.pdf. [Zugriff am 07 06 2023].

[63] Infineon Technologies AG, „AURIX™ 32-bit microcontrollers for automotive and industrial applications," 2020.

[64] ARM White Paper, „The ARM Cortex-A9 Processors," 2009.

[65] Texas Instruments Incorporated, „Application Note - Sitara™AM64x /AM243x Benchmarks," 2021.

[66] DIN EN IEC 61784-3:2022-02, Industrielle Kommunikationsnetze - Profile - Teil 3: Funktional sichere Übertragung bei Feldbussen - Allgemeine Regeln und Festlegungen für Profile (IEC 61784-3:2021).

[67] S. Ditting, „The Oxymoron of Modern Automation," in *14th Int. TÜV Rheinland Symposium - Functional Safety and Cybersecurity in Industrial Applications*, Cologne, Germany, 2022.

[68] F. Schiller, D. Judd, P. Supavatanakul, T. Hardt und F. Wieczorek, „Enhancement of safety communication model," *at-Automatisierungstechnik*, Bd. 70, p. 38–52, 2022.

[69] C. Werner, H. Zilligen, B. Köhler und R. Apfeld, „Sichere Antriebssteuerungen mit Frequenzumrichtern. IFA Report 4/2018," Deutsche Gesetzliche Unfallversicherung e. V. (DGUV), Berlin, Germany, 2018.

[70] Siemens AG, „Funktionshandbuch SINAMICS S120 Safety Integrated," Erlangen, Germany, 2020.

[71] SEW-EURODRIVE GmbH & Co KG, „MOVIDRIVE Modular, MOVIDRIVE System MOVISAFE CS..A Safety Card," Bruchsal, Germany, 2017.

[72] DIN EN 60204-1:2019-06, Sicherheit von Maschinen - Elektrische Ausrüstung von Maschinen - Teil 1: Allgemeine Anforderungen (IEC 60204-1:2016, modifiziert).

[73] DIN EN ISO 10218-1:2012-01, Industrieroboter - Sicherheitsanforderungen - Teil 1: Roboter (ISO 10218-1:2011).

[74] DIN ISO/TS 15066:2017-04, Roboter und Robotikgeräte - Kollaborierende Roboter (ISO/TS 15066:2016).

[75] DIN EN ISO 3691-4:2020-11, Flurförderzeuge - Sicherheitstechnische Anforderungen und Verifizierung - Teil 4: Fahrerlose Flurförderzeuge und ihre Systeme (ISO 3691-4:2020).

[76] M. Bdiwi, S. Krusche, J. Halim, P. Eichler, S. Hou, A. Rashid, I. A. Naser und S. Ihlenfeldt, „Situational zone-based robot control for heterogeneous safety sensors in agile HRI applications," in *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2022.

[77] J. Arents, V. Abolins, J. Judvaitis, O. Vismanis, A. Oraby und K. Ozols, „Human–robot collaboration trends and safety aspects: A systematic review," *Journal of Sensor and Actuator Networks*, Bd. 10, p. 48, 2021.

[78] A. Rashid, K. Peesapati, M. Bdiwi, S. Krusche, W. Hardt und M. Putz, „Local and Global Sensors for Collision Avoidance," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020.

[79] S. Kumar, S. Arora und F. Sahin, „Speed and Separation Monitoring using On-Robot Time-of-Flight Laser-ranging Sensor Arrays," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019.

[80] R. Apfeld, M. Hauke, P. Rempel, B. Ostermann, C. Werner, T. Bömer und M. Huelke, Das SISTEMA-Kochbuch 1: Vom Schaltbild zum Performance Level - Quantifizierung von Sicherheitsfunktionen mit SISTEMA - Version 2.0 (DE), Sankt Augustin, Germany: Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA), 2020.

[81] DR. JOHANNES HEIDENHAIN GmbH, „Technische Information - EnDat 3: Bewährte Sicherheitstechnologie konsequenz weiterentwickelt,“ 11 2019. [Online]. Available: https://endat.heidenhain.com/fileadmin/pdf/de/01_Produkte/Technische_Dokumentation/TI_EnDat3_ID1305415_de.pdf. [Zugriff am 02 01 2023].

[82] T. Wilkening, T. Cetin, H. Reiter und J. O. Krah, „EnDat 3 – Safety-Related Fully Digital Encoder Interface from the Application Point of View,“ in *PCIM Europe digital days 2020; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2020.

[83] SICK STEGMANN GmbH, „HIPERFACE DSL® SAFETY,“ Donaueschingen, Germany, 2016.

[84] Hengstler GmbH, „SCS open link - Hengstler GmbH,“ [Online]. Available: https://www.hengstler.de/de/specials/scs_open_link.php. [Zugriff am 13 05 2023].

[85] R. Apfeld, „Brauchen sichere Antriebssteuerungen auch sichere Positionsgeber?,“ Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA), Sankt Augustin, Germany, 2017.

[86] T. Bömer und M. Schaefer, „Unterschiede bei der Verwendung von fertigen Sicherheitsbauteilen und Standardbauteilen für die Realisierung von Sicherheitsfunktionen an Maschinen,“ Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA), Sankt Augustin, Germany, 2011.

[87] J. Croyle, „Holistic Functional Safety in Robotics, Practical Considerations for Functional Safety Success in Autonomy,“ in *14th Int. TÜV Rheinland Symposium - Functional Safety and Cybersecurity in Industrial Applications*, Cologne, Germany, 2022.

[88] T. Huckle, *Kleine BUGs, große GAUS - Softwarefehler und ihre Folgen*, http://www5.in.tum.de/~huckle/bugsn.pdf, 2003.

[89] M. Huelke, N. Becker und M. Eggeling, „Sicherheitsbezogene Anwendungssoftware von Maschinen. IFA Report 2/2016,“ Deutsche Gesetzliche Unfallversicherung e. V. (DGUV), Berlin, Germany, 2016.

[90] T. Bömer, K.-H. Büllesbach, M. Hauke, S. Otto und C. Werner, „Praxisgerechte Umsetzung der Anforderung für sicherheitsbezogene Embedded-Software nach DIN EN ISO 13849-1. IFA Report 1/2020," Deutsche Gesetzliche Unfallversicherung e. V. (DGUV), Berlin, Germany, 2020.

[91] PLCopen -Technical Committee 5, „Safety Software Technical Specification Part 1: Concepts and Function Blocks," 2020.

[92] R. P. Goldberg, „Architectural principles for virtual computer systems," 1973.

[93] S. Biemueller und U. Dannowski, „L4-Based Real Virtual Machines - An API Proposal -," in *MIKES 2007 First International Workshop on Microkernels for Embedded Systems,* Australia, 2007.

[94] Project ACRN™, „Project ACRN Documentation," 2023. [Online]. Available: www.projectacrn.github.io. [Zugriff am 28 04 2023].

[95] Real-Time Systems GmbH, „RTS Safe Hypervisor," Ravensburg, Germany, 2022.

[96] R. Kaiser und S. Wagner, „Evolution of the PikeOS microkernel," in *First International Workshop on Microkernels for Embedded Systems,* 2007.

[97] R. West, Y. Li, E. Missimer und M. Danish, „A virtualized separation kernel for mixed-criticality systems," *ACM Transactions on Computer Systems (TOCS),* Bd. 34, p. 1–41, 2016.

[98] F. Armand und M. Gien, „A practical look at micro-kernels and virtual machine monitors," in *2009 6th IEEE Consumer Communications and Networking Conference,* 2009.

[99] A. Iqbal, N. Sadeque und R. I. Mutia, „An overview of microkernel, hypervisor and microvisor virtualization approaches for embedded systems," *Report, Department of Electrical and Information Technology, Lund University, Sweden,* Bd. 2110, p. 15, 2009.

[100] G. Heiser und B. Leslie, „The OKL4 Microvisor: Convergence point of microkernels and hypervisors," in *Proceedings of the first ACM asia-pacific workshop on Workshop on systems,* 2010.

[101] BlackBerry QNX, „Product Brief - QNX Hypervisor for Safety,“ Ottawa, Canada, 2019.

[102] R. Pickles, „System-on-a-Chip certifiable OS Solution,“ 2017. [Online]. Available: https://www.sysgo.com/blog/article/system-on-chip-certifiable-os-solution. [Zugriff am 28 04 2023].

[103] Intel Corporation, „Intel® 64 and IA-32 Architectures Software Developer’s Manual Combined Volumes:1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4,“ 2023. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html. [Zugriff am 03 05 2023].

[104] C. Dall, S.-W. Li, J. T. Lim und J. Nieh, „ARM Virtualization: performance and architectural implications,“ *ACM SIGOPS Operating Systems Review*, Bd. 52, p. 45–56, 2018.

[105] P. I. Corke, „A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters,“ *IEEE Transactions on Robotics,* Bd. 23, pp. 590-594, 2007.

[106] Z. Shareef, „Path planning and trajectory optimization of delta parallel robot,“ 2015.

[107] R. L. Williams, „The Delta Parallel Robot: Kinematic Solutions,“ 2016. [Online]. Available: https://www.ohio.edu/mechanical-faculty/williams/html/PDF/DeltaKin.pdf. [Zugriff am 27 04 2022].

[108] D. Reinert und M. Schaefer, Sichere Bussysteme für die Automation, Heidelberg: Hüthig Verlag, 2001.

[109] EtherCAT Technology Group, „Safety over EtherCAT Protocol specification,“ 2011.

[110] T. Schmidt, J. O. Krah und J. Holtz, „High-Performance Control Architecture for Automation Drives based on a Low-Cost Microcontroller in Combination with a Low- Cost FPGA,“ in *PCIM Europe digital days 2021; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2021.

[111] PLCopen Technical Committees 5, „Safe Motion,“ 2017.

[112] J. O. Krah, M. Katz, T. Schmidt und B. Jeppesen, „Lean Safe Drive Architecture with Fully Integrated Multi-Axis Safety Functions due to an Extremely Fast Safety-related Fieldbus Interface,“ in *PCIM Europe digital days 2021; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2021.

[113] DGUV Fachbereich Holz und Metall, „DGUV Information Kollaborierende Robotersysteme - Planung von Anlagen mit der Funktion "Leistungs- und Kraftbegrenzung",“ Germany, 2017.

[114] J. O. Krah, B. Koehler und J. Koss, „Safety Related Current Monitoring for Multiphase Motors built with Digital Current Transducers,“ in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2019.

[115] T. Wilkening, J. O. Krah, M. Salardi und F. Heinzelmann, „Safety-Related High-Performance Motion Control based on a Quad-Core SoC,“ in *PCIM Europe digital days 2021; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2021.

[116] M. Winzenick, „Fehlertoleranz in der Maschinensicherheit Teil 1 - Grundlagen,“ ZVEI - Zentralverband Elektrotechnik, 2019.

[117] M. Winzenick, „Fehlertoleranz in der Maschinensicherheit Teil 2 - Anforderungen,“ ZVEI - Zentralverband Elektrotechnik, 2021.

[118] T. Schmidt, F. Heinzelmann, J. Holtz und J. O. Krah, „Fault-Tolerant Regenerative Sensorless Braking of PMAC Motors Enables Degraded Mode of Operation for Functional Safety,“ in *PCIM Europe 2022; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2022.

[119] J. O. Krah und H. Goergen, „High-Performance IEC 61131-3 PLC for Mixed-Critical Applications Enables a Centralized Diagnostic Architecture for Autonomous Systems and optional Degraded Operation,“ in *14th Int. TÜV Rheinland Symposium - Functional Safety and Cybersecurity in Industrial Applications*, Cologne, Germany, 2022.

[120] P. Gölzer, „Big Data in Industrie 4.0-Eine strukturierte Aufarbeitung von Anforderungen, Anwendungsfällen und deren Umsetzung,“ 2017.

[121] C. J. Bartodziej und C. J. Bartodziej, The concept industry 4.0, Springer, 2017.

[122] B. Vogel-Heuser, „Die Auflösung der Automatisierungspyramide: Die Maschinenkommunikation in der Smarten Fabrik,“ 2016.

[123] OPC Foundation, „OPC 10000-14 OPC Unified Architecture Part 14: PubSub,“ 01 11 2022. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/171. [Zugriff am 29 04 2023].

[124] M. Silveira Rocha, G. Serpa Sestito, A. Luis Dias, A. Celso Turcato und D. Brandão, „Performance Comparison Between OPC UA and MQTT for Data Exchange,“ in *2018 Workshop on Metrology for Industry 4.0 and IoT*, 2018.

[125] eProsima, „www.eProsima.com,“ [Online]. Available: www.eprosima.com. [Zugriff am 08 01 2023].

[126] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau und R. Dillmann, „Performance evaluation of real-time ROS2 robotic control in a time-synchronized distributed network,“ in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021.

[127] C. S. V. Gutiérrez, L. U. S. Juan, I. Z. Ugarte, I. M. Goenaga, L. A. Kirschgens und V. M. Vilches, „Time Synchronization in modular collaborative robots,“ *arXiv preprint arXiv:1809.07295,* 2018.

[128] OPC Foundation, „OPC 10000-15 OPC Unified Architecture Part 15: Safety,“ 01 11 2022. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/172. [Zugriff am 08 01 2023].

[129] A. S. Tanenbaum, Moderne Betriebssysteme, Pearson Deutschland GmbH, 2009.

[130] SECO S.p.A., „SMARC® Rel 2.1.1 compliant module with the Intel® Atom® x6000E Series and Intel® Pentium® and Celeron® N and J Series processors (formerly Elkhart Lake) for FuSa applications,“ [Online]. Available: https://edge.seco.com/de/halley.html. [Zugriff am 31 08 2023].

[131] Zephyr® Project, a Linux Foundation Project, „Zephyr® Project,“ [Online]. Available: https://zephyrproject.org/. [Zugriff am 03 05 2023].

[132] Zephyr® Project, a Linux Foundation Project, „Zephyr Project RTOS – First Functional Safety Certification Submission for an Open Source Real Time Operating System,“ [Online]. Available: https://www.zephyrproject.org/zephyr-project-rtos-first-functional-safety-certification-submission-for-an-open-source-real-time-operating-system/. [Zugriff am 03 05 2019].

[133] F. Reghenzani, G. Massari und W. Fornaciari, „The real-time linux kernel: A survey on Preempt_RT,“ *ACM Computing Surveys (CSUR),* Bd. 52, p. 1–36, 2019.

[134] Yocto Project®, A Linux Foundation Collaborative Project, „Yocto® Project,“ [Online]. Available: https://www.yoctoproject.org/. [Zugriff am 03 05 2023].

[135] CODESYS GmbH, [Online]. Available: https://www.codesys.com/. [Zugriff am 03 05 2023].

[136] CODESYS GmbH, „CODESYS SAFETY SIL2,“ [Online]. Available: https://de.codesys.com/produkte/codesys-safety/safety-sil2.html. [Zugriff am 05 03 2023].

[137] CODESYS GmbH, „CODESYS Online Help: Compound Safety PLC,“ [Online]. Available: https://content.helpme-codesys.com/de/CODESYS%20Safety%20SIL2/sil2_f_compound_safety_plc.html. [Zugriff am 03 05 2023].

[138] autonox Robotics GmbH, „DELTA RL4-400-0,5kg,“ [Online]. Available: https://autonoxfinder.com/de/A_00806. [Zugriff am 15 05 2023].

[139] Beckhoff Automation GmbH & Co. KG, „EL41xx Analoge Ausgangsklemmen (16 Bit),“ [Online]. Available: https://www.beckhoff.com/en-us/products/i-o/ethercat-terminals/el4xxx-analog-output/el4132.html. [Zugriff am 03 05 2023].