Valon Osmani, BSc

Dynamische Sicherheit in kollaborativen Arbeitsumgebungen der Zukunft (Industrie 4.0)

Safety Industrie 4.0

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur
Masterstudium Information and Computer Engineering

eingereicht an der
Technischen Universität Graz

Betreuer:
Dr. rer. nat. Diwold Konrad
TU Graz, Institut für Technische Informatik

Graz, Mai 2019

KOOPERATIONSPARTNER

SIEMENS
Ingenuity for life

Abstract

Industry 4.0 is a future project for the digitization of the classic industry through the latest information technologies. The challenges of this transformation is a self-organized, more efficient and individualized production. Previous automation systems are not able to achieve these characteristics. Therefore, new intelligent systems are necessary, which are capable of learning and can reconfigure at runtime. This is possible through the use of cyber-physical systems (CPS). CPS are distributed, intelligent objects connected through the Internet of Things and Services. This allows humans and machines to communicate with one another in future intelligent factories. The human being will not only be the operator of a machine, but will be the focus, guiding the machines through intelligent assistance systems and collaborating with robots. Collaboration states that man and machine can work together without structural restrictions. As a result of these developments, working environments are also affected. Future factories are expected to adapt industrial workspaces to highly dynamic environments. These become more dynamic as people and robots can work in future factories without strict physical separations. Therefore, an extension of fail-safe operations will be required. The traditional, static approach to fail-safe methods will not be sufficient in such scenarios. When dangerous errors occur, the operators have previously pressed the emergency stop button to prevent further damage to people and systems involved. This allowed the system to be brought to a safe state. Or, for example, the intrusion of people into the robot work areas, this sets immediately to a standstill. However, this impairs the flexibility and delays the production. In this work, it was investigated how the traditional, static approach for fail-safe operation can be adapted for use in future dynamic production scenarios. With regard to this study, a potential collaborative work scenario of the future was designed. Building on this, the analysis of the scenario with regard to the necessary safety aspects for the adaptation of existing fail-safe concepts was carried out. Based on the analysis, an experimental Demonstrator was developed to demonstrating how Fail-Safe can be implemented in dynamic work environments. The industrial environment is divided into three so-called "Virtual Safety Areas"(VSA). Each includes part of the collaborative workspace. Two emergency stop buttons are instantiated in each VSA which are connected to the automation system. The movements of objects (robot, human) in the VSA is monitored by a camera. The use of machine vision (computer vision) determines which object is currently in which VSA and how the objects move. The static approach could be adapted in the mass, so that the safety in dangerous situations acts only on appropriate VSA. The system is thus able to provide a selective action use emergency stop buttons, safe state for the entry of prohibited areas as well as a course of motion for the detection / prevention of safety-critical movements.

ZUSAMMENFASSUNG

Ein erklärtes Ziel von Industrie 4.0 ist die Digitalisierung von Produktionsprozessen zur Realisierung einer selbstorganisierten, effizienten und individualisierten Produktion. Dazu werden neue intelligente Produktionssysteme benötigt, welche lernfähig sind und sich zur Laufzeit rekonfigurieren lassen. Die Produktionsumgebung wird zu einem cyber-physischen System in dem intelligente Objekte und Menschen stetig interagieren und kommunizieren.

Der Mensch steht in solchen Ansätzen im Mittelpunkt, leitet die Maschinen durch intelligente Assistenzsysteme an und kollaboriert mit Robotern und Maschinen. Kollaboration bedeutet hier, dass Mensch und Maschine ohne bauliche Einschränkungen zusammenarbeiten. Diese Entwicklungen werden auch die Arbeitsumgebungen betreffen. Für zukünftige Fabriken bedeutet dies, dass industrielle Arbeitsbereiche an die hoch dynamischen Umgebungen angepasst werden müssen um eine Zusammenarbeit von Mensch und Maschine ohne strikte physische Trennungen zu ermöglichen. Zur Realisierung bedarf es einer Erweiterung von industrieller Fail-Safe Operationen, da der traditionelle, statische Fail-Safe Ansatz für solche Szenarien nicht mehr ausreicht. Derzeit erfolgt eine strikte Trennung von Mensch und Maschine, ein Verletzen dieser Trennung führt zur Überführung des Systems in einen sicheren Zustand außerdem sind bestehende Sicherheitskonzepte auf statische Systeme ausgelegt. Beim Auftreten von gefährlichen Fehlern wurde bisher von den Operatoren der Not-Aus-Taster gedrückt, um weiteren Schaden für beteiligte Menschen und Systeme zu verhindern. Dadurch konnte das System in einen sicheren Zustand gebracht werden. Der Einsatz solcher klassischen Methoden im Kontext von kollaborativen Arbeitsumgebungen beeinträchtigt deren Flexibilität und damit einhergehend deren Produktivität. Diese Arbeit untersucht wie der traditionelle, statische Ansatz für Fail-Safe Operation für die Anwendung in zukünftigen dynamischen Produktionsszenarien angepasst werden kann. Zwecks dieser Untersuchung wurde ein potenzielles zukünftiges kollaboratives Arbeitsszenario entworfen. Darauf aufbauend erfolgte die Analyse des Szenarios hinsichtlich der notwendigen Sicherheitsaspekte zur Adaption bestehender Fail-Safe Konzepte. Auf Basis der Analyse wurde ein experimenteller Demonstrator entwickelt, welcher demonstriert, wie Fail-Safe in dynamischen Arbeitsumgebungen umgesetzt werden kann. Die entwickelte industrielle Umgebung ist in drei so genannte "Virtual Safety Areas "(VSA) unterteilt. Jede Area umfasst einen Teil des kollaborativen Arbeitsbereichs. In jeder VSA sind zwei Not-Aus Schalter instanziiert, welche mit dem Automatisierungssystem verbunden sind. Die Bewegungen von Objekten (Roboter, Mensch) in und zwischen den VSAs wird per Kamera überwacht. Durch den Einsatz von maschinellem Sehen wird erfasst, welches Objekt sich gerade in welcher VSA befindet und wie sich die Objekte bewegen. Dadurch kann der statische Sicherheitsansatz in einen dynamischen Ansatz überführt werden, sodass die Sicherheit in Gefahrensituationen nur auf entsprechende VSA einwirkt. Das System ist somit fähig einen selektiven Einwirkung Einsatz von Not-Aus Schaltern, Safe-State für die Betretung von verbotenen Bereichen sowie einen Bewegungsverlauf für die Erkennung/Prävention von sicherheitskritischen Bewegungen zu schaffen.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

UNTERSCHRIFT:..... DATUM: .....

ABKÜRZUNGSVERZEICHNIS

AI Analogeingabebaugruppe

AQ Analogausgabebaugruppe

CPS Cyber-Physischen Systeme

E/E/PE S Elektrische Elektronische und programmierbare Elektronische

DF FA Digital Factory Factory Automation

FSM Fail-Safe Module

HAZOP Hazard Operability

HMI Human-Machine Interaction

IKT Informations- und Kommunikationstechnologie

IoT Internet of Thinks

OPC UA Open Platform Communications Unified Architecture

M2M Machine-To-Machine

MRK Menschen Roboter Kollaboration

MLFB Maschinenlesbare Fabrikatebezeichnung

RPI Raspberry PI

SIL Safety Integrity Level

SPS Speicherprogrammierbare Steuerung

TIA Totally Integrated Automation

TS Technische Spezifikation

INHALTSVERZEICHNIS

Tabellenverzeichnis

Abbildungsverzeichnis

KAPITEL 1

EINLEITUNG

Mit dem Zukunftsprojekt Industrie 4.0 der deutschen Regierung (November 2011 verabschiedet [23]) wird die Digitalisierung der klassischen Industrie durch neueste Informationstechnologien durchgeführt. Teile dieser Technologien sind Internet der Dinge, Cloud Computing, Big-Data Analytics, M2M-Communication, Augmented Reality usw. [26], [1], [22]. Das Hauptziel dieser Wandlung ist eine selbstorganisierte, effizientere und individualisierte Produktion [35]. Bisherige Automatisierungssysteme sind nicht fähig diese Eigenschaften zu erreichen [58]. Daher sind neue intelligente Systeme notwendig, die lernfähig sind und sich zur Laufzeit rekonfigurieren können [25]. Dies ist durch den Einsatz von Cyber-Physischen Systemen (CPS) möglich. CPS sind verteilte, intelligente Objekte, die durch das Internet der Dinge und Dienste verbunden sind [41].Dadurch können Mensch und Maschine in den zukünftigen intelligenten Fabriken miteinander kommunizieren. Der Mensch wird nicht nur der Operator einer Maschine sein, sondern steht im Mittelpunkt, leitet die Maschinen durch intelligente Assistenzsysteme und kollaboriert mit Robotern [56], [54]. Kollaboration heißt in diesem Fall, dass das Zusammenarbeiten zwischen Menschen und Robotern ohne bauliche Einschränkung ermöglicht wird [53].

Im Zuge dieser Entwicklungen werden auch die Arbeitsumgebungen beeinflusst. Es wird erwartet, dass die industriellen Arbeitsbereiche in der Zukunft an die hoch dynamische Arbeit angepasst werden und dadurch hoch dynamische Arbeitsumgebungen entstehen. Diese werden dadurch dynamischer, indem Menschen und Roboter in den zukünftigen Fabriken ohne strikte physische Trennungen arbeiten können. Außerdem können mögliche Rekonfigurationen der Arbeitsumgebung während des Produktionsprozesses zu nachhaltigen Änderungen des Betriebsablaufs führen.

Durch diese Entwicklungen wird auch eine Erweiterung bzw. eine Überarbeitung von Fail-

Safe-Methoden erforderlich sein. Der Grund dafür ist, dass der traditionelle, statische Ansatz für Fail-Safe-Operationen in solchen Szenarien nicht ausreichen. Wenn derzeit gefährliche Fehler in der Produktion auftreten, ist die gewöhnliche Reaktion von Arbeitern, den Not-Aus-Taster zu drücken, um weiteren Schaden für beteiligte Menschen und Systemen zu verhindern. Oft führt dies dazu, dass das gesamte System in einen sicheren Zustand versetzt wird, und dadurch seine Aufgabe nur noch in stark eingeschränkter Weise wahrnehmen kann. Im schlimmsten Fall kommt es zu einer (kostspieligen) Totalabschaltung. Wenn beispielsweise derzeit der Arbeitsbereich eines Roboters von Menschen betreten wird, wird dieser sofort in Stillstand gesetzt. Das vermindert die Flexibilität und verzögert die Produktion.
Für zukünftige hoch dynamische Arbeitsumgebungen ist dieses Prinzip aber nur bedingt einsetzbar, da es die neuen Dimensionen der Flexibilität nicht berücksichtigen kann. Die Fragestellung dieser Masterarbeit lautet daher: Wie kann und muss der traditionelle, statische Ansatz für Fail-Safe-Operation für zukünftige hoch dynamische Arbeitsumgebungen angepasst bzw. adaptiert werden?

Um diese Frage zu beantworten, wird im ersten Schritt ein potenzielles kollaboratives Arbeitsszenario der Zukunft entworfen. Das Szenario stellt eine hoch dynamische Arbeitsumgebung dar. In dem Szenario soll es möglich sein, dass Mensch und Maschine ohne räumliche Trennung miteinander arbeiten, außerdem soll die flexible Rekonfiguration der Arbeitsumgebung hinsichtlich der notwendigen Sicherheitsanforderungen gewährleistet sein. In einem zweiten Schritt erfolgt die Analyse des Szenarios hinsichtlich der notwendigen Sicherheitsaspekte zur Erarbeitung neuer bzw. Adaption bestehender Fail-Safe-Konzepte.

Auf Basis der Analyse wurde ein experimenteller "Dynamic Safety Demonstrator (DSD)" entwickelt, welcher demonstriert, wie Failsafe und Safety in dynamischen Arbeitsumgebungen umgesetzt werden können. Die industrielle Umgebung wurde in mehrere "Virtuelle Safety Areas" (VSA) geteilt. Jede VSA umfasst einen Teil des kollaborativen Arbeitsraums. In jeder VSA sind zwei Not-Aus- Taster instanziiert, welche mit dem Automatisierungssystem verbunden sind. Die Bewegungen von Objekten (also Roboter oder Menschen) in den VSAs werden durch eine Kamera überwacht. Durch den Einsatz von Methoden des maschinellen Sehens ("Computer Vision") wird festgestellt, welches Objekt sich gerade in welcher VSA befindet und wie sich die Objekte bewegen. Sofern Sicherheitsregeln von Mensch oder Maschine vorliegen, ist es dem System möglich die spezifische VSA in einen sicheren Zustand zu fahren, wobei andere VSA unbeeinflusst bleiben. Außerdem ermöglicht das Fail-Safe-System eine dynamische Rekonfiguration der Sicherheitsregeln während des Betriebs. Der Demonstrator wurde mit Standard-Automatisierungs- und Sicherheitskomponenten der Simatic-Automatisierungsfamilie realisiert.

Die vorliegende Arbeit gliedert sich wie folgt: Kapitel 2 beschreibt die Entwicklungen der industriellen Revolution und ihre Konsequenzen für das Zusammenarbeiten von Mensch und Maschine und entwirft das Anwendungs-Szenario, das Gegenstand dieser Masterarbeit ist. Kapitel 3 gibt eine Einführung in funktionale Sicherheit und relevante Sicherheitsnormen für kolla-

borierende Arbeitsbereiche. Kapitel 4 führt eine Sicherheitsanalyse des Anwendungsszenarios durch. Kapitel 5 beschreibt die Komponenten des Sicherheitssystems. Kapitel 6 beschreibt die praktische Implementierung des Demonstrators. In Kapitel 7 sind die Ergebnisse der Arbeit mit Fokus auf Safety dargestellt. In Kapitel 8 erfolgen eine Zusammenfassung der Arbeit sowie ein Ausblick auf mögliche Erweiterungen des Demonstrators.

KAPITEL 2

INDUSTRIE 4.0

In diesem Kapitel werden die Digitalisierung der Industrie sowie die Auswirkungen dieser Digitalisierung auf bestehende Arbeitsprozesse veranschaulicht. Dazu erfolgt zuerst ein kurzer historischer Überblick über die vorangegangenen Industriellen Revolutionen, sowie die derzeitige 4. Industrielle Revolution. Danach erfolgt eine Diskussion derzeitiger Forschungsergebnisse hinsichtlich der zukünftigen Kollaboration zwischen Menschen und Maschinen. Auf Basis dieser Ergebnisse wird ein potentielles Kollaborationsszenario der Zukunft entworfen.

2.1 Industrie 4.0

Da es hierbei um die vierte industrielle Revolution geht, wird im folgenden ein kurzer Überblick auf die vergangenen Revolutionen (Abbildung 2.1) gegeben.

1. **Industrielle Revolution**

   Der Prozess der Industrialisierung beginnt nach der Erfindung der Dampfmaschine im 18. Jhdt (genauer gesagt 1750 [2]). Als erste Revolution der Industrie wird die Einführung mechanischer Produktionsanlagen mithilfe von Wasser- und Dampfkraft bezeichnet [37]. Von da an wird die schwere körperliche Arbeit von Maschinen übernommen. Damit erreichte man die Mechanisierung des Produktionsprozesses.

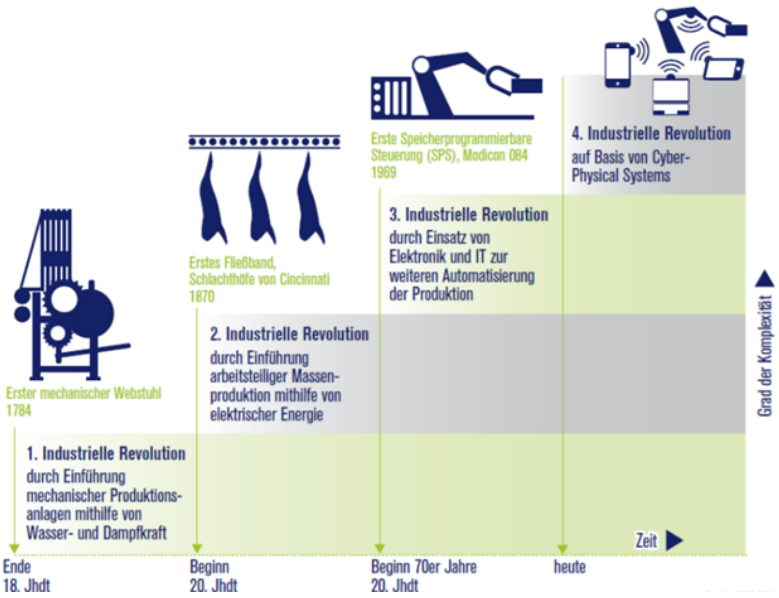


Abbildung 2.1: Die vier Stufen der Industrielle Revolution [23]

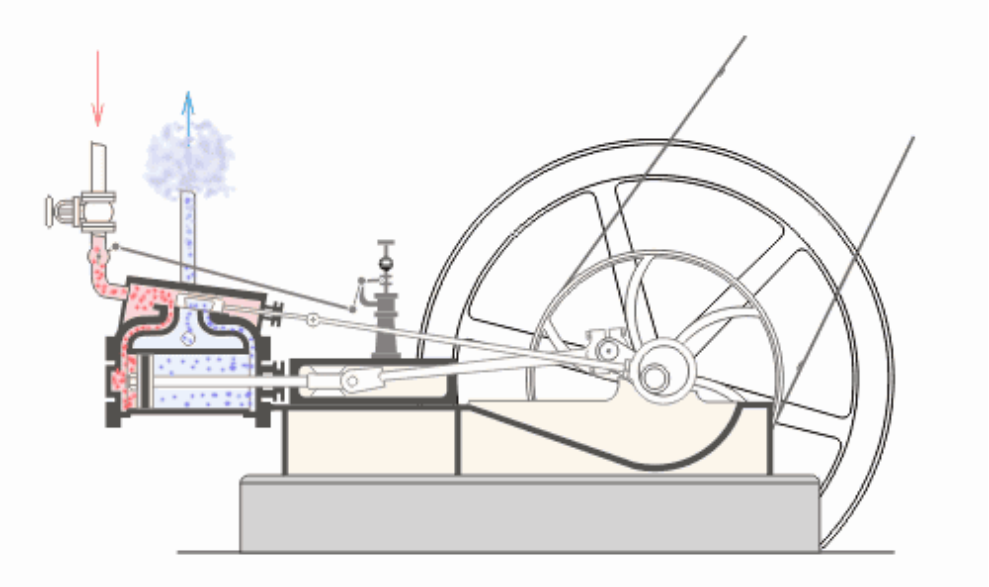


Abbildung 2.2: Dampfmaschine [42].

2. **Industrielle Revolution**

   Die zweite industrielle Revolution beginnt mit der Erfindung des Elektromotors. Die Dynamomaschine von Werner von Siemens ermöglichte die Generierung von elektrischem Strom in größeren Mengen [11] und die Elektrifizierung von Produktionsprozessen. Das ermöglichte eine Dezentralisierung der elektrifizierten Antriebssysteme, wobei für die Arbeitsmaschinen ein dezentraler Antrieb möglich war. Arbeitsintensive Massenproduktion

mithilfe von Elektrizität kennzeichnet diese Revolution [2].

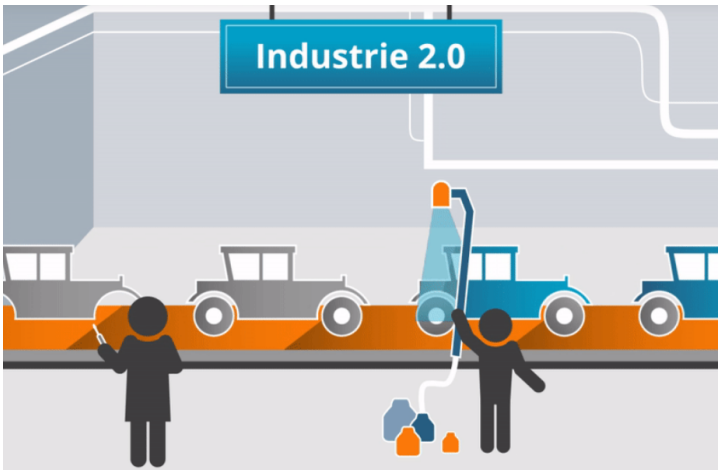


Abbildung 2.3: Fließband [54]

3. **Industrielle Revolution** Diese beginnt mit der Einführung von Elektronik und Informationstechnologie zur Automatisierung der Industrie-Produktion [23]. Ende der 1970er Jahre kommen die ersten speicherprogrammierbaren Steuerungen. Im Jahr 1968 wurde eine speicherprogrammierbare Steuerung (SPS) von einer US-amerikanischen Firma entwickelt [11]. Weil sich dadurch SPS und Mechanik zusammenschließen konnten, führte das schließlich zur Automatisierung der Produktionsprozesse [9].

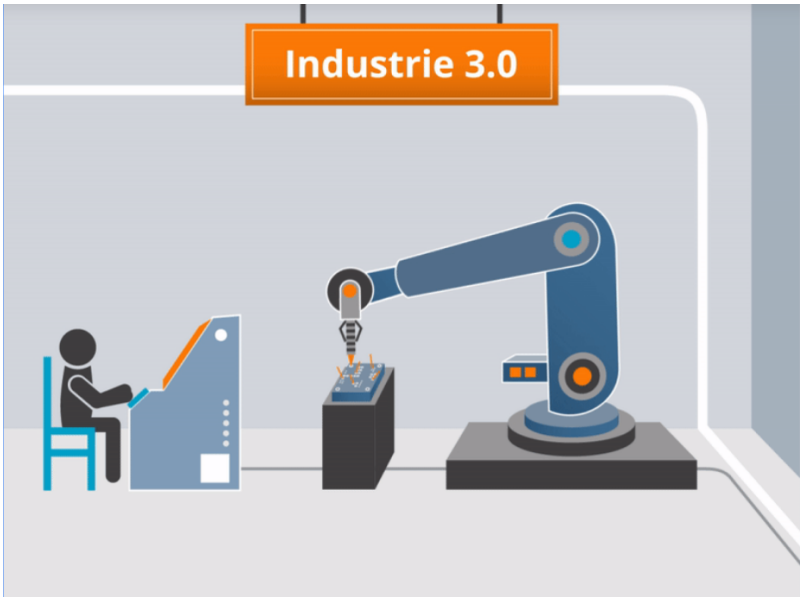


Abbildung 2.4: Einsatz von Elektronik und IT für weitere Automatisierung [54]

4. **Industrielle Revolution** Mit dem Zukunftsprojekt Industrie 4.0 der deutschen Regierung

(November 2011 verabschiedet [23]) wird die Digitalisierung der klassischen Industrie durchgeführt. Durch neueste Informationstechnologien wird eine Erweiterung der bisherigen Technologie erfolgen. Teile dieser Technologien sind Internet der Dinge, Cloud Computing, Big-Data Analytics, M2M-Communication, Augmented Reality usw. [26], [1] [22].

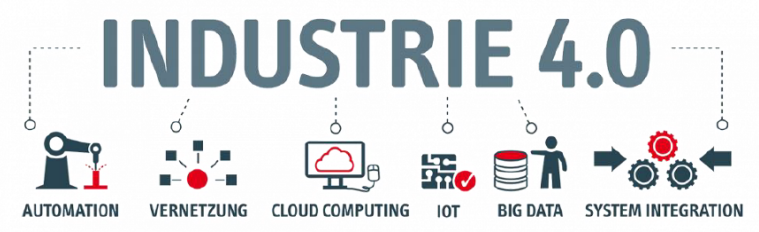


Abbildung 2.5: Basistechnologie für die Industrie 4.0 [22]

Die Ergebnisse dieser Wandlung erzielen eine höhere Variabilität der Produkte und verkürzen zugleich den Produktionslebenszyklus. Bisherige Automatisierungssysteme sind nicht fähig diese Flexibilität zu erreichen [58]. Daher sind neue intelligente Systeme notwendig, die während der Laufzeit lernfähig sind und sich rekonfigurieren können. Diese sollen in der Lage sein auch die virtuelle Computerwelt mit physischen Systemen zu verknüpfen. Durch den Einsatz von cyber-physischen Systemen ist das möglich. Diese umfassen "intelligente " Geräte, Maschinen, Produktions-, Logistik-, Engineering-, Koordinations- und Managementprozesse [20], die durch das Internet der Dinge (IoT) und Dienste verbunden sind. Diese stellen die technischen Grundlagen von CPS [41]. Dinge können in der physikalischen Welt intelligente Objekte sein und die Verarbeitung von Daten auf dem Sever erfolgt durch die Dienste im Internet. Diese Vernetzung zählt als zentrale Entwicklungsperespektive der Industrie 4.0 [21]. CPS präsentiert aktuelle und bedeutsame Entwicklungen in der Informations- und Kommunikationstechnologie (IKT) sowie in der Informatik [1].

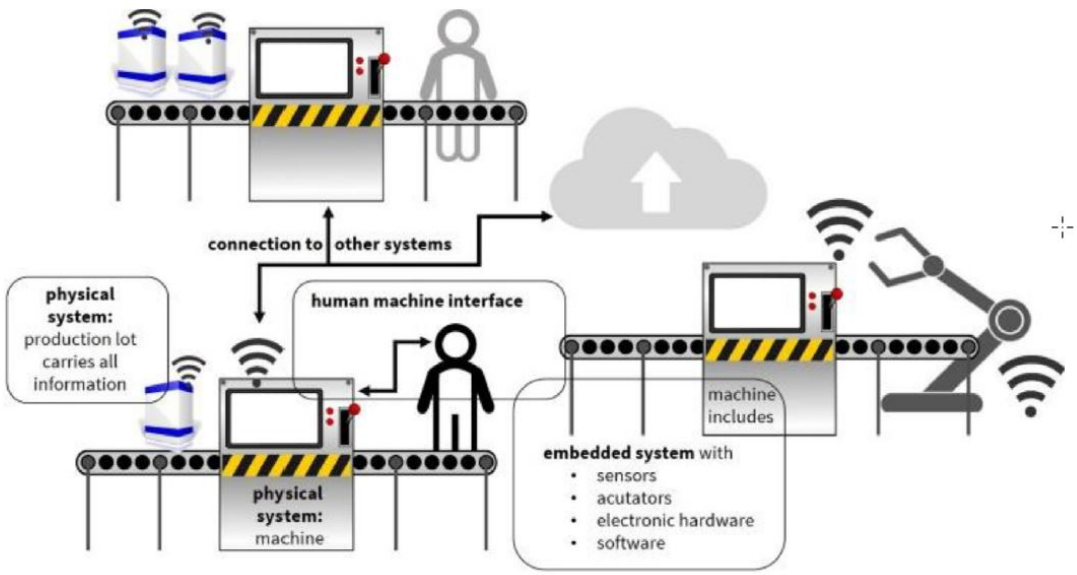


Abbildung 2.6: Struktur eines Fertigungs-CPS.

Durch den Einsatz von IKT werden Maschinen und Anlagen intelligenter und können autonom Informationen austauschen, Aktionen auslösen und sich unabhängig voneinander steuern. Mittels Sensorik werden Daten aus der physikalischen Welt erfasst und verarbeitet. Und durch Aktuatoren wird auf physikalische Vorgänge eingewirkt [25]. Daher sollten sich sie für die externen Anforderungen selbsttätig optimieren und rekonfigurieren [21]. Menschen und Maschinen werden hier digital miteinander vernetzt. Der Mensch steht im Mittelpunkt und leitet die Maschinen durch intelligente Assistenzsysteme und arbeitet mit Robotern zusammen [54]. Die Sicherheit ist eine wesentliche Aspekt der Industrie 4.0. Dadurch ist eine Anpassung der Sicherheit und Sicherheitsnormen in solchen Umgebungen erforderlich [25], da die Umgebung dynamischer wird und die Maschinen (z. B. Roboter) autonomer werden. Genauer gesagt, geht es um die Sicherheit der Zusammenarbeit zwischen Menschen und Maschinen in gemeinsamen und schutzzaunlosen Arbeitsbereichen [53]. Eine sicherheitstechnische Überwachung kann durch Verwendung der aktuellen Schutzeinrichtungen sowie Not-Aus-Taster, Lichtvorhänge, Laser-Scanner, Sicherheitsmatten, Safety EYE usw. den Kommunikationsansätzen aus Industrie 4.0 erfolgen.

2.1.1 Arbeitsbereiche der Zukunft

Nach der Einführung von Elektronik und Informationstechnologie zur Automatisierung der Industrie-Produktion übernahmen Roboter immer oefter Aufgaben, die für Menschen gefährlich und bzw. nicht ergonomisch waren [14]. Am Anfang war die Sicherheit ausschließlich auf die Aufmerksamkeit und das Urteil des Menschen begrenzt. Ausgehend davon, dass menschliche Fehler gefährliche Folgen haben können, entstand die Idee für ein Sicherheitskonzept. Dies kam

aus dem europäischen Raum und basiert auf dem Prinzip der "Isolation und des Stoppens ". Das Konzept ist zur Gewährleistung der Sicherheit durch das Design sicherer Maschinen entstanden [8]. Das bedeutete getrennte Räume für Arbeiter und Roboter. In den meisten Fällen geschieht die Trennung durch Schutzzäune und/oder Lichtschranken.

Wie zuvor dargestellt, wandeln sich bestehende Produktionsumgebungen im Rahmen von Industrie 4.0. Sie werden vernetzter, dynamischer und automatisierter, um eine möglichst automatische, effiziente und individualisierte Produktion zu ermöglichen. Auch die Zusammenarbeit zwischen Mensch und Maschine wird flexibler sein.

Die industrielle Umgebung wird jedoch durch Industrie 4.0 dynamischer bzw. vernetzter. Die Produktion wird effizienter und individualisierter sein. Jedoch ist es notwendig die Produktion bzw. Maschinen so wenig wie möglich anzuhalten, um die Flexibilität und Produktivität zu steigern. Deshalb ist es wichtig eine enge Zusammenarbeit zwischen Menschen und Robotern zu schaffen [14]. Daher wird es erforderlich sein eine zeitliche und räumliche Trennung dazwischen zu aufzuheben. Das bedeutet die schutzzaunlosen Arbeitsbereiche, die durch berührungslose Schutzeinrichtungen die Sicherheit garantieren.

2.1.2 Mensch-Roboter-Kollaboration in der Industrie 4.0

Es wird erwartet, dass engere Kollaborationen zwischen Menschen und Robotern auf industriellen Arbeitsplätzen zu einer schnelleren und besseren Produktion führen werden. Kollaboration heißt in diesem Fall, dass Menschen und Roboter einen Arbeitsraum teilen. Seit Erfindung von Robotern haben Menschen und Robotern immer getrennte Arbeitsbereiche gehabt. Roboter waren früher große "gefährliche " Maschinen, die in Käfigen lebten. In Abbildung 2.7 ist eine konventionelle Roboterapplikation mit Förderbändern dargestellt. Wobei verschiedene Schutzeinrichtungen wie Schutzzaun, die Lichtvorhänge, Not-Aus-Taster usw. eingebaut sind. Wenn ein Mitarbeiter die optischen Strahlen der Lichtvorhänge unterbricht oder eine Tür öffnet, wird der Roboter sofort in Stillstand versetzt. Somit wird die Sicherheit garantiert , aber sie ist nicht flexibel, sondern hart. Außerdem können in anderen Fertigungsbereichen, wie für Lebensmittel, Medizinprodukte, Kosmetik und Pflege- und Servicebereiche, die Menschen und Maschinen nicht isoliert werden [8].Hier ist das herkömmliche Sicherheitskonzept schwer implementierbar.

Abbildung 2.7: Industrie-Roboter in Montagelinie [8]

Dieses Konzept wurde Anfang des 21. Jahrhunderts verändert. Der heutige Einsatz der Roboter ist vielseitig. Er wird in verschiedenen Branchen eingesetzt. Um flexible und hohe Produktivitätzu ermöglichen, muss man neue Fertigungssysteme entwickeln, um damit zeitnah auf diversifizierte Anforderungen der Kunden reagieren zu können [8]. Im industriellen Feld spielen Industrie-Roboter eine zentrale Rolle. Seit wenigen Jahren gibt es einen neuen Art von Robotern. Die sogenannte "Cobots " [33]. Das ist eine Kombination aus Collabortionünd Robot", die für einen direkten Kontakt mit Menschen gestaltet wurden. Das entspricht der Zusammenarbeit zwischen Menschen und Robotern oder wie es auch genannt wird eine Mensch-Roboter-Kollaboration (MRK), die als eines der wichtigsten Konzepte der Industrie 4.0 gilt [53]. Eine Kombination zwischen Menschenstärke (Geschicklichkeit, Flexibilität, selbstständigen Problemlösung) und Roboterstärke (Zuverlässigkeit, Ermüdungsfreiheit, Wiederholgenauigkeit) erhöht die Effizienz der Gesamtprozesse. Die Sicherheit wird durch die Sensorik gewährleistet. Deswegen müssen Sicherheitseinrichtungen vorhanden sein, die in der Lage sind zu erkennen, dass sie sich bereits vor dem tatsächlichen Kontakt in der Nähe des Menschen befinden, und entsprechend reagieren oder mit so wenig Kraft arbeiten, dass auch bei einer Kollision kein Verletzungsrisiko für den Menschen besteht. Dafür müssen die Normen und Vorschriften berücksichtigt. Durch eine Erweiterung der Sicherheitseinrichtungen können auch die konventionellen Roboter in einem kollaborativen Arbeitsraum eingesetzt werden. Im Jahr 2016 kam mit der Erweiterung von ISO/TS 15066 eine technische Spezifikation für die Kollaboration zwischen Menschen und Robotern. Spezifikation umfasst, wie in Kapitel 3.1.1beschrieben, die vier Betriebsmodi [55].

2.1.2.1 Mechanismen - Sensorik

Für die technische Umsetzung benötigt man geeignete Sensorik- und Steuerungssysteme, die in Roboter oder in die Anlage (Arbeitsraum) einzubauen sind. Solche Systemen müssen in der Lage sein bei Gefahrsituationen für Menschen und Maschienen in jeden Arbeitsraum, wo sie eingebaut sind, die Sicherheit zu garantieren. JJe nach Anforderungen werden fehlersichere Steuerungssysteme und Sensorik verwendet. Verschiedene Hersteller auf der ganzen Welt haben derartige sichere Systeme hergestellt. Siemens produziert auch ein fehlersicheres Steuerungssystem. SIMATIC ist ein sehr bekanntes SPS-System und modular aufgebaut. Die Systeme können fehlersicher Digital- und Analog-Signale auswerten. Andere Hersteller bieten diverse Sicherheitssensoren. Wenn Roboter und Mensch im gleichen Arbeitsraum arbeiten, dann kann man berührungslose Schutzeinrichtungen verwenden. Es folgen einige barrierefreie Schutzeinrichtungen erwähnt, die schon am Markt sind. Sicherheitslichtvorhänge kommen hauptsächlich zum Einsatz bei Gefahrenstellen, die von Menschen des während normalen Betriebes nicht erreichbar sein dürfen. Die fehlersichere Halt-Funktion ist die Hauptanforderung an einen sicheren Lichtvorhang. Die sicheren Laserscanner bieten eine zweidimensionale Schutzzone. Je nach Hersteller wird die Flächen-Überwachung mit verschiedenen Radien angeboten und mit einem Öffnungswinkel bis zu 360 °C. Außerdem sind sie klein und sehr kompakt und können eine große Reichweite haben. Die Flächenüberwachung wird unterteilt in eine Schutzzone (geschützter Bereich) und eine Wahrnzone. Wenn der Mensch die Warnzone betritt, werden die Bewegungen der Maschinen reduziert. Wenn er noch weiter in die Richtung der Maschine kommt und die Schutzzone betritt, dann wird die Maschine gestoppt. Der Maschinenprozess wird automatisch fortgesetzt, wenn der Mensch weggeht. Das erhöht die Produktivität der Anlage, weil dadurch die Stillstandzeiten minimiert werden. Es geht wieder um Menschenschutz. Sicherheitsschaltmatten dienen für das sichere Abschalten von Maschinen in gefahrbringenden Zonen. Sicherheitsschaltmatten erkennen Personen oder Objekte, die sich in einem definierten Bereich bewegen. Bei gefahrbringenden Bewegungen können die Maschinen sicher abgeschaltet werden. Die Verbindung mit Sicherheitsschaltgeräten und Steuerungssystemen garantiert die Sicherheit in vordefinierten Bereichen. . Wenn jemand auf die Oberfläche der Matte tritt oder eine Last darauf platziert wird, werden zwei parallele Kontaktflächen zusammengedrückt und der elektrische Stromkreis hergestellt. Eine Einsatzapplikation ist meistens dort, wo berührungslose Schutzeirichtungen nicht geeignet sind. Kamerabasierte sichere Systeme gehören zu berührungslosen Schutzeinrichtungen. Der große Vorteil dabei ist, dass Positionen von mehreren Personen überwacht werden können. Die Überwachung der Absicherung im Sicherheitsbereich ist die Hauptfunktionalität. Dadurch kann das Eindringen von Personen in sicheren Bereichen detektiert werden. Die Positionen werden von einer Sicherheitssteuerung berechnet und je nach Abstand die erforderlichen Maßnahmen eingeleitet [14]. Die sichere Überwachung wird hauptsächlich in zwei Zonen oder Bereiche abgestuft. In Warnbereich und Schutzbereich. Beim Eindringen von Objekten oder Personen in den Warnbereich werden die Maschinengeschwindigkeit reduziert und der Abstand gemessen.

Wenn aber der Schutzraum betreten wird, dann werden die Maschinen sicher abgeschaltet. Wenn keine Objekte oder Menschen im Schutzbereich sind, wird die Maschine oder der Roboter selbsttätig gestartet. Safety EYE von Pilz GmbH and Co.KG ist ein sicheres kamerabasiertes System. Dadurch erfolgt ein sichere 3D-Raumüberwachung und ermöglicht die Erkennung von Menschen oder Objekten, die sich im Feldsatz befinden oder dort eindringen.

2.1.3 Szenario

Wie genau könnte ein industrieller Arbeitsbereich der Zukunft aussehen? Um diese Frage ausführlich zu beantworten, reicht eine Masterarbeit nicht, jedoch kann hier eine erste Einschätzung abgegeben werden. Es wird ein experimenteller Demonstrator aufgebaut, der eine industrielle Umgebung simuliert. Je nach Einsatz der Maschinen bzw. Roboter in solchen Umgebungen wird es verschiedene Arbeitsbereiche geben, die natürlich unterschiedliche Betriebsarten haben. Laut ISO/TS 15066 für Mensch-Roboter-Kollaboration gibt es vier verschiedene Kollaborationsarten, die auch hier miteinbezogen werden, sowie das Worker Only. Eine detaillierte Beschreibung davon wird im Kapitel 3.1.1 gegeben. Daher ist es notwendig die Umgebung in mehrere Arbeitsbereiche zu teilen. Da es um mehrere dynamischen und flexibleren Umfelder geht, wird die Aufteilung der Arbeitsbereiche virtuell durch ein Kamerasystem erfolgen. Deshalb werden diese Arbeitsbereiche als Virtuelle Safety Area (SA) genannt.

Es wird mindestens einen Bereich geben, in welchem alles automatisch läuft und "klassische " Roboterapplikationen im Einsatz sind **(SA5)**. In solchen gemeinsamen Arbeitsbereichen hat der Arbeiter nur zu stillstehenden Robotern oder Maschinen Zugang. Denn in so einem Kollaborationsraum kann eine Kollision zwischen Menschen und Robotern zu Verletzungen führen. In einem anderen Arbeitsbereich soll kein physischer Kontakt zwischen Menschen und Robotern stattfinden. Aber es können parallel Mensch und Roboter im gemeinsamen Arbeitsbereich arbeiten. Die Sicherheit wird durch den Abstand zwischen Mensch und Roboter gewährleistet **(SA3)**. Es gibt eine stufenweise Reduzierung der Geschwindigkeit der Roboterbewegungen, wenn sich ein Mensch nähert. Wenn der Abstand aber so gering ist, dass es zu einer Kollision kommen kann, muss der Roboter sicher anhalten.

Ein sehr interessantes Thema der MKRs ist die direkte Zusammenarbeit, wobei die Bewegungen unabhängig sind. Cobots sind für solche Situationen in Produktionsbereiche konzipiert, deswegen sind sie auch als Leichtbauroboter bekannt. Natürlich werden auch solche Arbeitsbereiche in dieser Arbeit miteinbezogen. Außerdem werden im selben Arbeitsbereich mobile Roboter operieren, die autonom navigieren können. Ein Beispiel dafür ist der mobile Roboter(MR) von KUKA "KMR iiwa " [27]. Das ist eine Kombination aus Cobot und einer mobilen sowie autonomen Plattform. Er wird auch Produktionsassistent genannt und kann auch in andere Arbeitsbereiche fahren. Das heißt, in solchen Arbeitsumgebungen können kollaborierende Roboter, mobile Roboter und Menschen zusammenarbeiten **(SA4)** und **(SA6)**. Trotz fortschreitender Technologie, Vernetzung und einer kommunkationsfähigen Fabrik (Smart Factory) bleibt der

Mensch immer noch im Mittelpunkt. Die Menschen (Arbeiter) brauchen freie Bereiche **(WorkerOnly)** hinsichtlich der Roboter oder Maschinen. Diese Arbeitsbereiche **(SA2)** können auch als freie Räume dienen oder für andere Arbeitsaktivitäten (z. B. kurze Besprechungen) genutzt werden. In der Zukunft werden Mobile-Roboter eingesetzt und bei schutzzaunlosen Bereichen ist es sehr wahrscheinlich, dass Mobile-Roboter(MR) solche Zonen (Virtuelle Safety Area) betreten, die nicht für MR erlaubt sind. Durch dieses Projekt wird experimentell versucht, in derartigen industriellen Umgebungen die Sicherheit für Menschen und Maschinen zu gewährleisten. Wie die Idee ausschauen wird, ist in Abbildung 2.9 dargestellt.

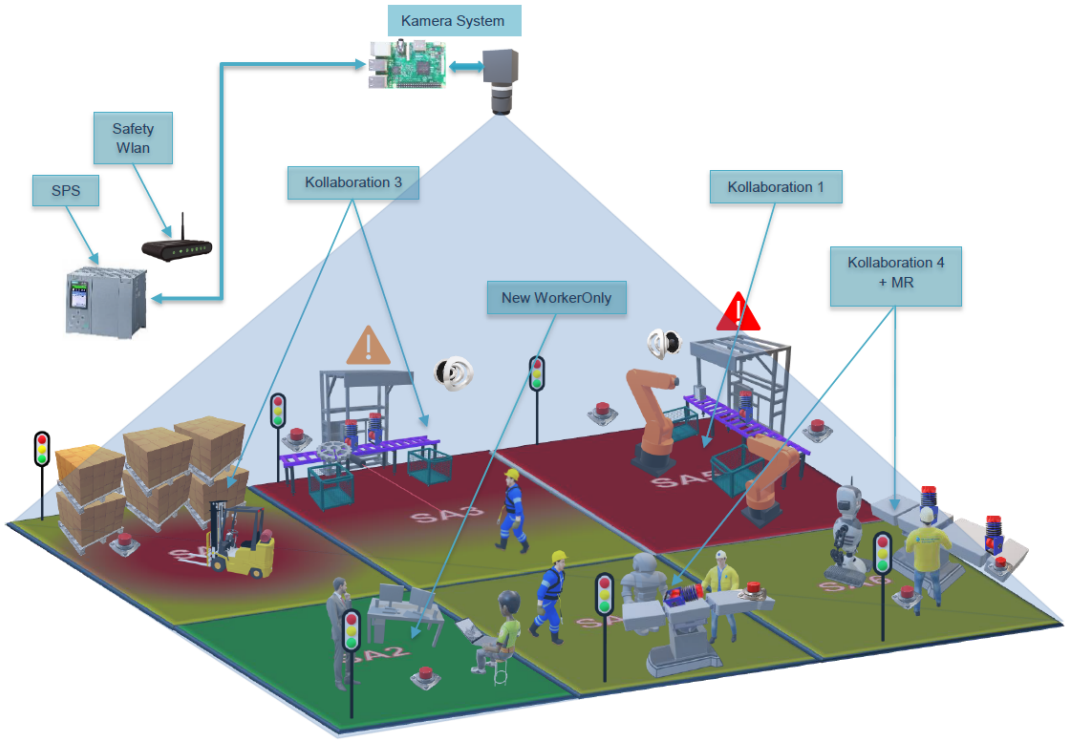


Abbildung 2.8: Scenario Arbeitsbereiche der Zukunft.

Die Idee ist Arbeitsbereiche zu schaffen, die keine getrennten Räume oder Zäune brauchen. Und möglichst ein kompaktes System "Alles in Einem " zu schaffen, das über die gesamte Anlage die Sicherheit gewährleistet und um nicht für jeden Arbeitsbereich diverse Lösungen zu suchen. Dafür ist mir etwas Schlaues eingefallen: Ein Kamerasystem für die Überwachung und ein Steuerungssystem für die Auswertung und das sichere Abschalten bzw. Anhalten. Um die Gefährdungen möglichst gering zu halten, kommt hierzu die Sicherheitstechnik ins Spiel. Diese Technik beschäftigt sich mit Risiken, denen Mensch und Umgebung ausgesetzt sind. Das Kamerasystem und Not-Halt sind Schutzeinrichtungen, die beide mit einem Steuerungssystem gekoppelt werden. Das Steuerungssystem ist für die Auswertung von Signalen und für eine sichere Reaktion zuständig.

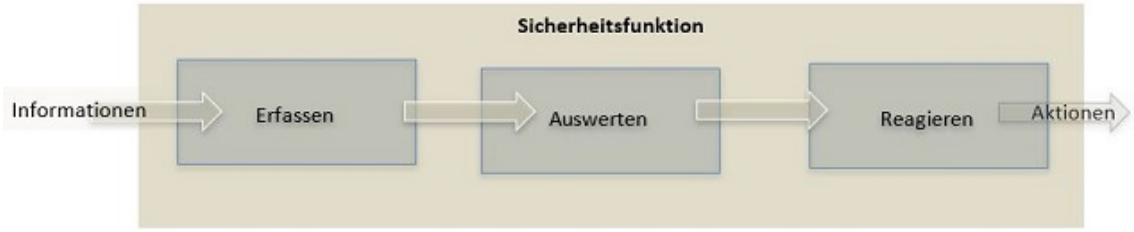


Abbildung 2.9: Architektur Sicherheitsfunktion.

Mit derartigen Sicherheitssystemen befasst sich die Norm IEC 61805 für funktionale Sicherheit elektrischer, elektronischer und programmierbarer Systeme. Diese ist auch als Basis Sicherheitsnorm bekannt und nicht abhängig von einem Anwendungsgebiet [16].Die funktionale Sicherheit hängt von der korrekten Funktion vom Sicherheitssystem ab. Das bedeutet, dass im Fehlerfall die Anlage in sicherem Zustand gehalten oder in sicherem Zustand gesetzt muss. Wie wir schon wissen, gibt es im Markt bereits durchgängige Automatisierungssysteme für die Sicherheitstechnik. Das Failsafe (Fehlersicher) Simatic-System von Siemens ist ein durchgängiges Automatisierungssystem und nach IEC 61508 Norm mit einer hohen Zuverlässigkeit an Sicherheit (SIL3) zertifiziert. Das heißt, ich kann Failsafe Simatic verwenden, aber nur das allein ist nicht genug, weil die Sicherheit sich jetzt nur auf Komponenten bezieht und nicht auf die Umgebung. Auch die meisten "Lösungen " von sicheren Robotern oder der Sensorik hängen bis jetzt von der jeweiligen Applikation ab [33]. Durch die Kombination aus verschiedenen kollaborativen Arbeitsarten und den Einsatz von Cobots und mobilen Robotern wird die Zukunft der industriellen Arbeitsbereiche gestaltet.

2.1.4 Fehler Szenario Not-Halt Befehlsgerät

Eine der wichtigsten Sicherheitsfunktionen, worauf diese Arbeit fokussiert, ist der Not-Halt durch Not-Aus-Taster. Laut EG-Maschinenrichtlinien 2006/42/EG muss jede Maschine und Anlage mit einem oder mehreren Not-Halt- oder Sicherheits-Befehlsgeräten ausgestattet sein [55]. Dieses ist manuell zu betätigen und muss so positioniert werden, dass es für den Bediener ungehindert und schnell zugänglich ist. Die Auslösung des Not-Halt-Befehlsgerätes erfolgt dann, wenn es sich dabei um gefahrenbringende Situationen oder einen Notfall handelt. Daher sollten die von der Maschine ausgehenden Gefahren verhindert oder gemindert werden. Als Not-Halt-Gerät wird ein Not-Aus-Taster verwendet, der von den Betriebsarten unabhängig ist. Deshalb sollte es in jedem Arbeitsbereich Not-Aus-Taster geben. Der traditionelle, statische Ansatz für ein Fehler-Szenario mit Not-Halt-Befehlsgeräte kann die gesamte Anlage in Stillstand versetzen. Der traditionelle, statische Ansatz ist in Abbildung 2.10 zu sehen.

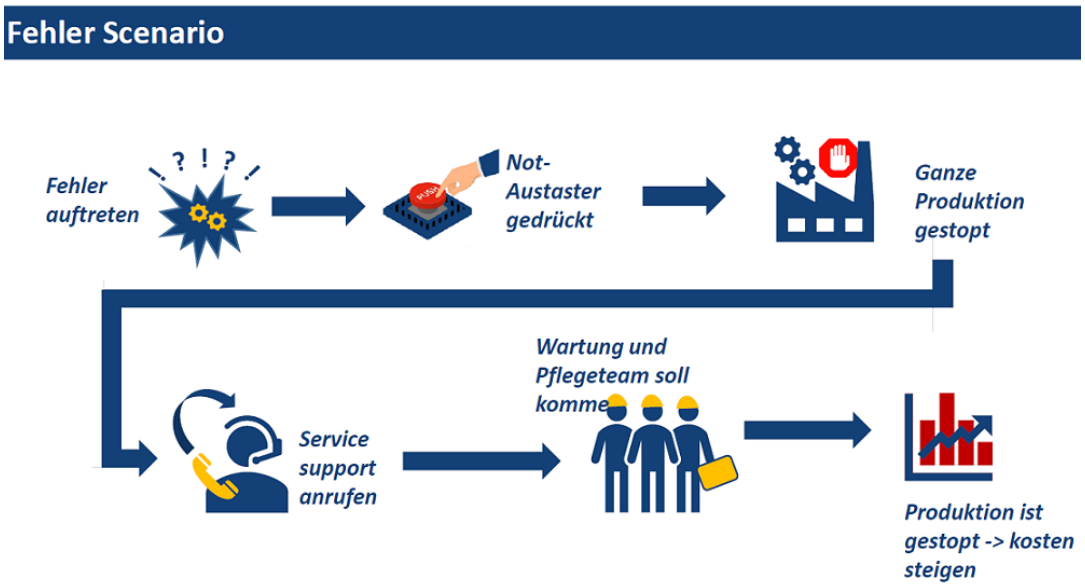


Abbildung 2.10: Fehler-Scenario.

Wir möchten die Sicherheit industrieller Umgebungen verbessern, indem wir Selektivität in der Sicherheitsumgebung implementieren. Wie erwähnt, wird durch diese Arbeit versucht, die Ausfallssicherheit für ein mehr dynamische Umgebungen zu verbessern, in die Selektivität in der Sicherheitsumgebung implementiert wird. Das bedeutet, dass bei Gefahrensituationen der Not-Aus-Taster die Wirkung nur auf die Maschinen bzw. Roboter begrenzt, die sich vorübergehend oder statisch in diesem Arbeitsbereich befinden. Dadurch wird eine Stillstandsetzung der gesamten Anlage verhindert.

KAPITEL 3

FUNKTIONALE SICHERHEIT

In diesem Kapitel wird die Funktionale Sicherheit von Sicherheitsfunktionen erläutert. Es werden die Sicherheitsnorm IEC 61508 für E/E/PES und SIL beschrieben. Außerdem folgt ein Überblick über Fehleranalyse und Safety-Konzept auf höchsten System-Ebene des Sicherheitssystems von Demonstrator.

3.1 Sicherheit

Man unterscheidet zwischen zwei Aspekten bei der Sicherheit von Anlagen und Maschinen in Industrie 4.0. Der erste Aspekt die Betriebssicherheit (engl. Safety), umfasst Risiken und Gefahren, die von Industrie-Systemen ausgehen können. Der zweite Aspekt, die Sicherheit, umfasst Mechanismen, die Anlagen und/oder Maschinen vor Angriffen und Missbrauch schützen (eng. Security) [23]. Sicherheit bedeutet die Freiheit von unvertretbaren Risiken. Grundvoraussetzung für die Betriebssicherheit ist die funktionale Sicherheit. Das fokussiert den Teil, wenn die Sicherheit von der korrekten Funktion des Systems abhängt. Das ist eine Grundanforderung für die Entwicklung der Maschinen und Systeme besonderes für kollaborative Arbeitsbereichen, wo die Menschen zusammen oder neben einanderarbeiten.

3.1.1 Sicherhetisnormen

In der Industrie wird Sicherheit durch die Normen geregelt. Bei der Entwicklung von neuen Maschinen, Robotern bzw. Prozessen erfolgt im ersten Schritt eine Risikobeurteilung. Die Risikobeurteilung befasst sich mit der Definition der Elemente, woraus das Sicherheitssystem besteht. Basierend auf den Elementen und der Hauptfunktionalität des Systems wird eine Risikoanalyse

durchgeführt, auf deren Basis ein Sicherheitskonzept bzw. Sicherheitsanforderungen des Systems erstellt werden.

Die Analyse ermöglicht die Identifizierung der Gefährdungen, die zu der Verletzung der Sicherheit führen können. Auf Basis der Analyse erfolgt die Definition geeigneter Maßnahmen, die das identifizierte Risiko auf ein vertretbares Restrisiko reduzieren. Das Verfahren ist durch verschiedene Sicherheitsnormen geregelt. Im Fall der Menschen-Roboter-Kollaboration (MRK) handelt es sich um verschiedene Betriebsarten von MRK [53] und [56] dargestellt wird, können die Sicherheitsnormen für kollaborierende Arbeitsbereiche zwischen Menschen und Robotern in drei Gruppen gegliedert werden. Die Gliederung ist in Abbildung 3.1 dargestellt.

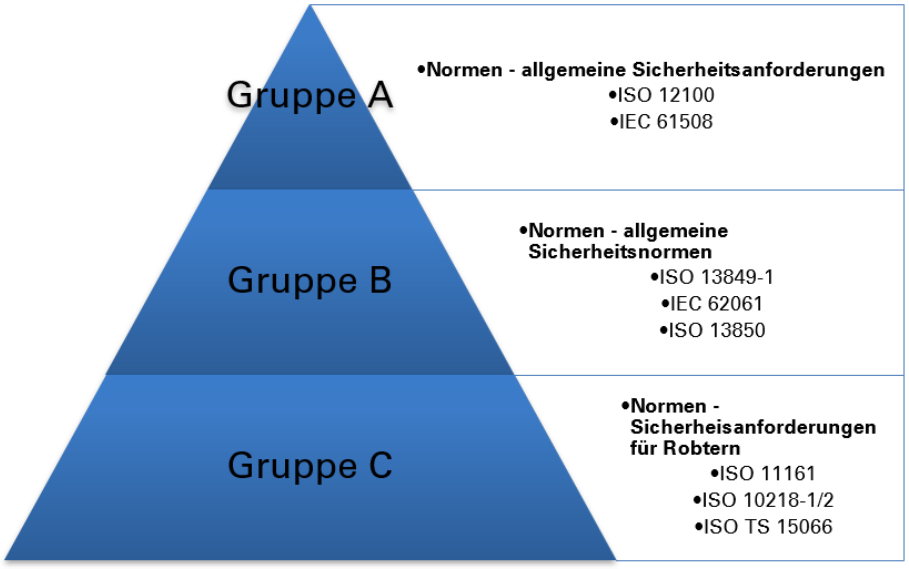
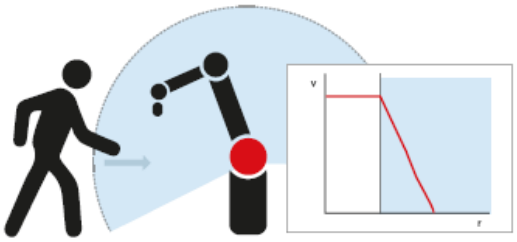


Abbildung 3.1: Sicherheitsnormen, die Relevant für Menschen Roboter Kollaboration sind.

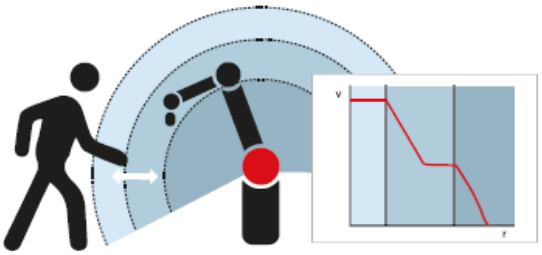
Normen in der Gruppe A befassen sich mit allgemeinen Sicherheitsanforderungen. Darunter fallen die Risikobeurteilung und Risikominderung. Während der Risikobeurteilung erfolgt die Definition der Elemente der Systems. Auf Basis dieser Definition erfolgt im nächsten Schritt eine Risikoanalyse. Danach erhält man ein Sicherheitskonzept und die Sicherheitsanforderungen einer Sicherheitssystem.IEC 61508 ist die Sicherheitsnorm für elektrische/elektronische/programmierbare elektronische Systeme, unabhängig von Anwendungen [28].

Normen der Gruppe B beschäftigen sich spezifisch mit der Sicherheit von Maschinen [33]. Die ersten zwei Normen (ISO 13849-1 und IEC 62061) spezifizieren spezifische Sicherheitsaspekte hinsichtlich des Designs und der Implementierung von sicherheitsbezogenen Steuerungssystemen. ISO 13850 Norm behandelt und spezifiziert die funktionalen Sicherheitsaspekte von Not-Halt-Vorrichtungen [56].

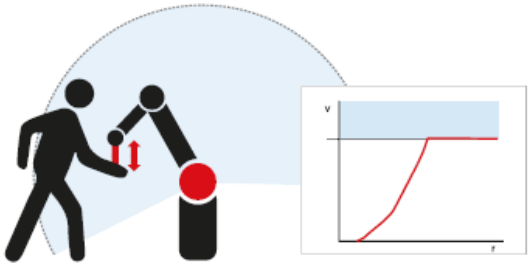


Sicherheitsbewerteter überwacher Halt In gemeinsamem Arbeitsraum wird Roboter im Stillstand gesetzt, wenn der Arbeiter eintritt. Und beim Verlassen wird Roboter weiter Arbeiten.

Handführung - Im gemeinsame Arbeitsraum wird Roboter im Stillstand gesetzt. Der Arbeiter führt den Roboter manuell



Geschwindigkeits- und Abstandsüberwachung Der Arbeiter hat Zugang zu Kollaborationsraum während (des Betriebs) der Roboter sich bewegt. Die Sicherheit wird durch den Abstand zu Roboter gewährleistet.



Leistungs- und Kraftbegrenzung - Der Arbeiter hat Zugang zu Kollaborationsraum, während während (des Betriebs) der Roboter sich bewegt. Eine Kollision zwischen beiden ist hier möglich. Die Kontaktkräfte müssen sichergestellt werden, dass zu keinen Verletzungen führen können [53].

Abbildung 3.2: Schutzprinzipien TS 15066 [53]

Normen der Gruppe C legen die Sicherheitsanforderungen an Industrie-Roboter und die Angaben für kollaborative Roboter (Teil 2) fest. ISO 10218 ist unter Maschinenrichtlinien gelistet. Teil 1 (z. B. Antriebsfunktionen) behandelt Roboter in der Fertigung. Der zweite Teil wurde im Jahr 2011 spezifiziert und deckt die Sicherheitsanforderungen für ein Robotersystem und dessen Integration ab [15]. Im Gegensatz zu "klassischen " Roboterapplikationen wird bei MRK eine Kollision möglich sein. Aber das darf nicht zu Verletzungen von Menschen führen. Voraussetzung dafür sind zulässige Steuerungen und intelligente, dynamische Sensoren am Roboter selbst

[33]. Die Gesetzliche Randbedingungen erlaubten die Kollaboration von Menschen und Robotern nach Anpassungen und Neuentwicklungen der Normen und Richtlinien [17]. Besonders wichtig für MRK ist die Technische Spezifikation ISO/TS 15066 der Anfang der 2016 veröffentlicht wurde. Um die Sicherheit des Menschen jeder Zeit sicherzustellen, sind bei ISO/TS 15066 vier Kollaborationsarten für den Schutz des Menschen bei der Kollaboration (Abbildung 3.1.1).

Eine Kombination aus diesen Kollaborationsformen ist bei der Umsetzung möglich. Bei einem schutzzaunlosen Betrieb kann es auch zu einer Kollision kommen. Wichtig ist, dass es bei einer unbeabsichtigten Berührung, bzw. bei einer Kraft oder Druckeinwirkung zu keinen Verletzungen kommt. Für die Massenträgheit von Schmerz-Welle in verschiedene Körperregionen geht ISO/TS 15066 detailliert in ihrem Anhang A (für alle Körperteile die biomechanischen Kollisionsgrenzwerte) ein.

3.2 Funktionale Sicherheit

Funktionale Sicherheit ist Bestandteil der Betriebssicherheit. Über Funktionale Sicherheit wird gesprochen, wenn die Sicherheit von korrekter Funktion des Sicherheitsfunktion abhängt. Funktionale Sicherheit befasst sich wesentlich mit Erkennung bestimmter gefährlicher Ausfälle, die zu schwerwiegenden Folgen (das Leben gefährden) führen können. Das Ziel ist solche mögliche Funktionsausfälle auf ein vertretbares oder adaptierbares Maß zu reduzieren [28]. Funktionale Sicherheit ist vorhanden, wenn eine Sicherheitsfunktion korrekt ausgeführt wird und im Fehlerfall entsteht kein unsicherer Zustand. Die Reduzierung des Risikos erfolgt durch geeignete Maßnahmen auf ein akzeptables Restrisiko. Risiko ist laut Sicherheitsnormen eine Kombination aus dem Ausmaß eines Schadens und der Wahrscheinlichkeit des Auftretens eines Schadens. Die Gewährleistung der funktionalen Sicherheit erfolgt nach Erfüllung der Anforderungen von Sicherheitsnorm (z. B. IEC 61508).

3.2.1 Basisnorm IEC 61508

Als Basisnorm der funktionalen Sicherheit ist die Internationale Norm IEC 61508 für sicherheitsbezogene, elektrische/elektronische/programmierbare elektronische Systeme E/E/PES und unabhängig von Anwendungen [28]. Die erste Revision diese Norm wurde 1998 publiziert und die zweite Revision erfolgte im Jahr 2010, wobei die Arbeit für die nächste geplante Revision (2020) bereits begonnen hat. Von dieser Basisnorm leiten sich andere Normen je nach spezifischer Anwendungen ab. Die Deutsche Version von IEC 61508 ist DIN EN 61508 und wurde Jahr 2001 übernommen. Die deutsche Norm gliedert sich gleich wie die internationale Norm. DIN EN 61508 besteht aus sieben Teilen:

1. Teil: Allgemeine Anforderungen

2. Teil: Anforderungen an sicherheitsbezogenen elektrischer/elektrionischer /programmierbarer elektronischer Systeme (E/E/PES)

3. Teil: Anforderungen an Software

4. Teil: Begriffe und Abkürzungen

5. Teil: Beispiele zur Ermittlung der Stufe der Sicherheitsintegrität

6. Teil: Anwendungsrichtlinie für Teil 2 und Teil 3

7. Teil: Anwendungshinweise über Verfahren und Maßnahmen

Sicherheitsziel von IEC 61508 ist, die Risikos, die von einem System ausgehen, auf ein akzeptables Restzisiko zu reduzieren [28]. Daher basiert auf DIN EN 61508 wird ein Safety Konzept erstellt. Die Verlaufslogik, auf der Konzept erstellt wurde, basiert auf der abgeleiteten Norm ISO 26262. Aber die spezifischen Begriffe werden aus dem IEC 61508 verwendet. Industrie allgemein und Industrie 4.0 werden durch verschiedene Standards der funktionalen Sicherheit auf der Grundlage der IEC 61508 gut abgesichert [29]

3.2.1.1 Sicherheitsintegritätslevel SIL

Die Stufe der Anforderungen an Sicherheit wird in der internationalen Sicherheitsnorm IEC 61508 als Sicherheitsintegritätslevel SIL (Safety Integrity Level) oder auch Sicherheitsstufe bezeichnet. SIL steht für einen Sicherheitslevel und ist ein Mittel, um die erforderliche Risikominderung darzustellen, die erforderlich ist, um das Risiko auf ein akzeptables Restrisiko zu reduzieren [29]. Nach Beurteilung des Schadensausmaßes und der Schadenswahrscheinlichkeit ist es möglich, die Risikostufe eines Systems festzulegen [43]. Wie das Verfahren beurteilt, dazu wird in der Abbildung ein sogenannter Risikograph präsentiert, der zur Verständigung des Verfahrens dient. Eine Beurteilung von SIL wird in der vorliegenden Arbeit nicht durchgeführt, da dies der detaillierten Beurteilung durch mehrere Experten bedarf.

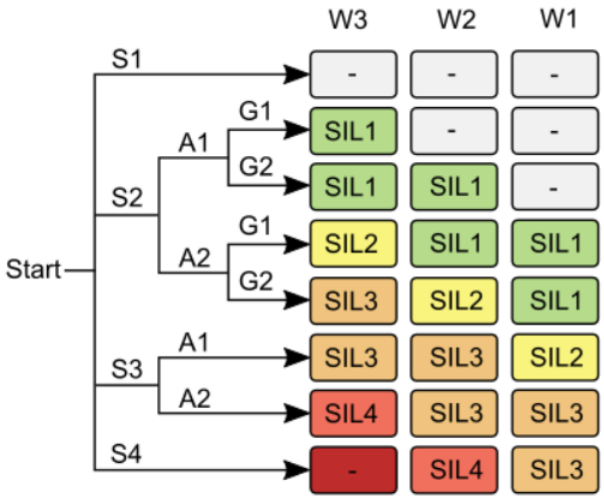


Abbildung 3.3: Risikograph von SIL - IEC 61508 [44].

S - S1 bis S4 präsentiert das Schadensausmaß oder den Schweregrad des Schadens. S1 bezeichnet die leichten Verletzungen, während S4 katastrophale Auswirkungen mit viele Toten bedeutet.

A - Häufigkeit und Dauer der Explosion im Gefahrenbereich.

G - Die Möglichkeit zur Vermeidung oder Begrenzung des Schadens.

W - Die Wahrscheinlichkeit des Auftretens des gefahrbringenden Ereignisses.

KAPITEL 4

SICHERHEITSANALYSE

In diesem Kapitel wird die Sicherheitsanalyse des Anwendeungszenarios durchgeführt. Hier werden nur einige Sicherheitsaspekte des Systems berücksichtigt.

4.1 Die Berücksichtigung des Top-Level-Sicherheitskonzepts und seine Anforderungen

4.1.1 Das Konzept

Ziel ist es, prototypisch zu versuchen bestehender Fail-Safe Operation in ein in einer dynamischen industriellen Umgebung zu adaptieren. Dafür wird eine experimentelles „Dynamic Safety Demonstaror“ aufgebaut. Die industrielle Umgebung ist in mehreren so genannten "Virtual Safety Areas " (VSA) geteilt. Jede davon umfasst einen Teil des kollaborativen Arbeitsbereichs. Das Schutzsystem besteht aus Not-Aus-Tastern, einer Kamera, einem Automatisierungssystem (Simatic-System) und aus einer Übertragungssystem. Zwei Not-Aus Taster sind in jeder VSA instantiiert, welche mit dem Automatisierungssystem (Simatic-System) verbunden sind. Die Bewegungen von Objekten (Roboter, Mensch) in den VSA wird durch eine Kamera überwacht. Durch den Einsatz von maschinellem Sehen (Computer Vision) wird festgestellt, welches Objekt sich gerade in welcher VSA befindet und wie sich die Objekte bewegen. Schutzsystem besteht aus eine Kamera. Und entsprechend werden die Objektskoordinaten durch Übertragungssystem (OPC UA) in die Simatic-System übertragen.

4.1.2 Overall Scope Definitionen

Um die Funktionalität besser zu verstehen wird das Schutzsystem in Sicherheitsfunktion und in sicherheitsbezogenes Sicherheitssystem (oder Sicherheitssystem) geteilt.
Sicherheitsfunktion - um Risiken von Gefährdungen zu minimieren, braucht man eine Sicherheitsfunktion, die die Funktionalität gewährleistet. Die Sicherheitsfunktion muss so gestaltet werden, dass das System (Anlage/Maschinen/Roboter) in Gefahrsituationen in einen sicheren Zustand zu bringen oder aufrechtzuerhalten ist. Die Hauptfunktion vom Schutzsystem ist: Hauptfunktion von dem Schutzsystem ist:

- Das System soll in Gefahrsituationen automatisch erkennen, auf welche Maschine/Roboter es einwirken soll.

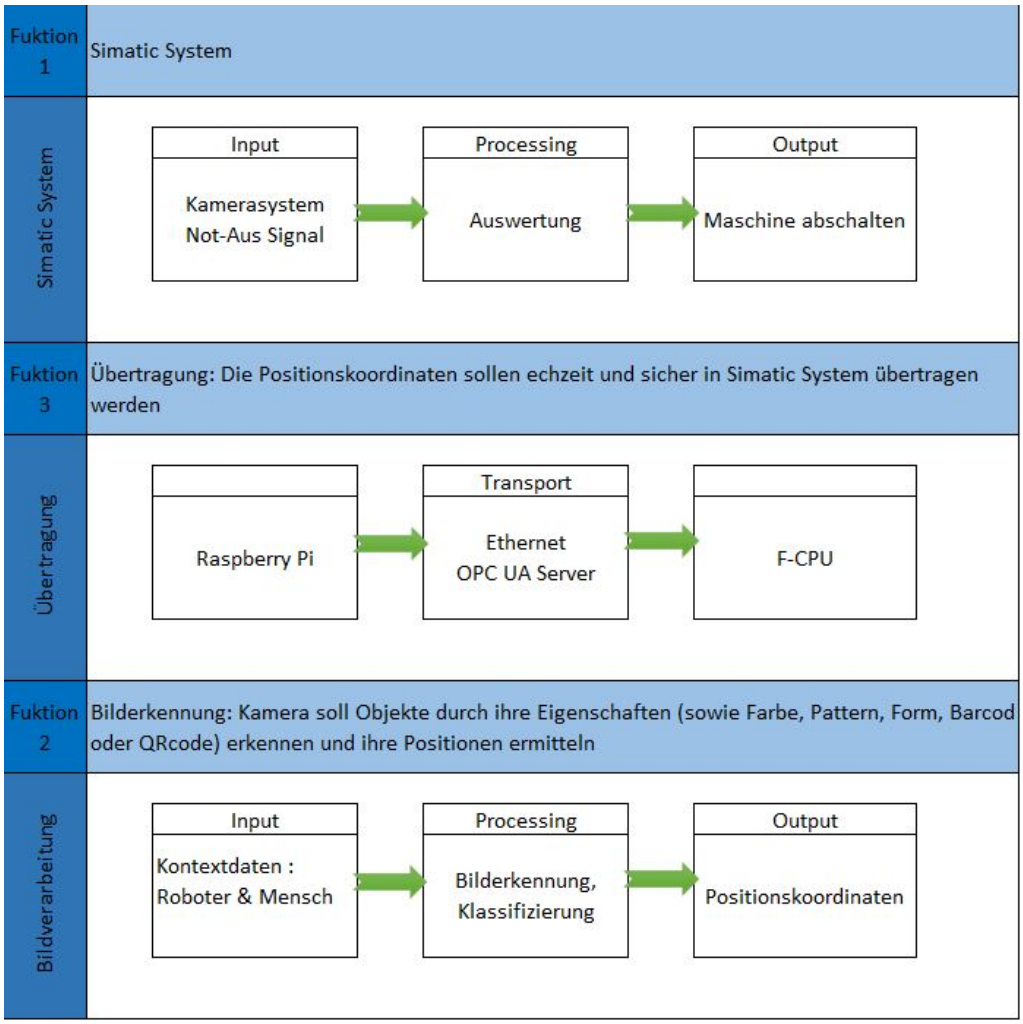


Abbildung 4.1: Fehlerarten

Sicherheitsbezogene Sicherheitssystem – besteht aus mehreren sicherheitsbezogenen

elektrisch/elektronisch/programmierbaren elektronischen Elementen, die erforderlich sind, um die Sicherheitsfunktion auszuführen. Laut DIN EN 61508 ist ein "sicherheitsbezogene Systeme " ein System, das ein oder mehrere Sicherheitsfunktionen ausführt.
Um ein Safety Concept zu schaffen, werden ein paar Schritte durchgeführt, die notwendig sind. Als Erstes werden die Funktionalität des Systems, die Architektur und die beteiligten Komponenten und deren Funktion erläutert. Dann kommt die Risikoanalyse (Hazard Analysis and Riskassessment). Von beiden Schritten leitet man das Safety-Konzept bzw. die Anforderungen ab.

In dieser Arbeit werden nur einige relevante Aspekte des Gesamtsicherheitskonzepts gezeigt. Hier werden nur die Top-Level-Safety-Anforderungen (eng. Requirements) und wesentliche architektonische Blöcke sowie der Block Channel betrachtet. In Abbildung 4.1 sind drei Hauptfunktionen, die wir betrachten werden. Deswegen werden 3 Arten von Fehlern auf der höchsten System-Ebene definiert:

1. Simatic-System-Fehler.
2. Übertragungsfunktions-Fehler.
3. Klassifizierung und Bildverarbeitung-Fehler.

Für jede Art des Fehlers werden eine Fehleranalyse und ein Safety-Konzept auf der höchsten System-Ebene für die Sicherheitsfunktion des Demonstrators entworfen

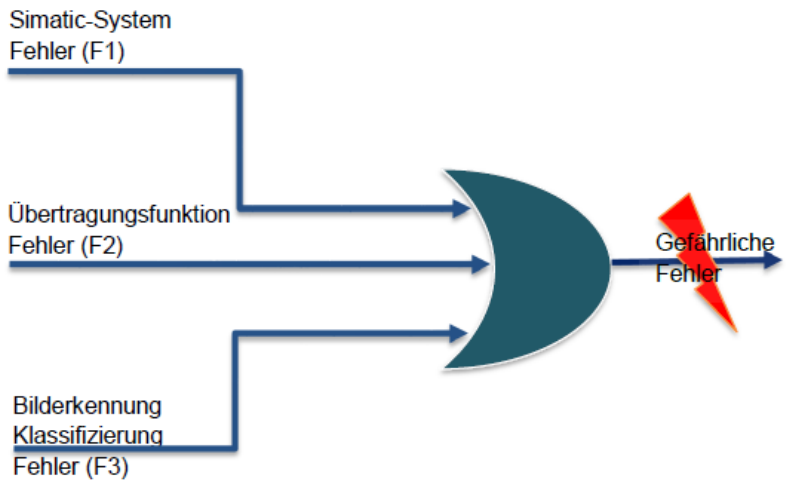


Abbildung 4.2: Fehlerarten des Sicherheitssystems.

4.1.3 "Hazards and Risks"- Analyse

In diesem Abschnitt werden Fehler analysiert, die während des eingeschalteten Sicherheitssystems auftreten können.

Type of Hazard and Risk analysis		
Main Funktions	**Hazard**	**Safety Relevant**
Simatic System	Rate ist bereit auf SIL3 redusiert	YES
Übertragung	Repetition - Unbeabsichtige Wiederholung	YES
	Deletion - Verlust der Daten	YES
	Insertion - Eingefühgt werden	YES
	Incorrect Sequence - Falsche Sequenz	YES
	Corruption	YES
	Delay - Inakzeptable Verzögerung der Daten	YES
	Masquerade - Maskieren	YES
Bildverarbeitung	Kein Bild	YES
	Stasiches Bild	YES
	Lichtverhältnissänderung	YES
	Position des Objekts falsch erkannt	YES

Abbildung 4.3: Gefährdungs- und Risikoanalyse

4.1.3.1 Simatic-System-Fehler (F1)

(**F1**) - Fehler, die aus Simatic-Systemen kommen können, werden in dieser Arbeit mit (**F1**) bezeichnet. Aufgrund der ausfallsicheren Architektur und der hohen Diagnoseabdeckung von Selbsttests ist die Ausfallrate bei SIL3 garantiert. Deswegen wird diese Art der Fehler nicht berücksichtigt.

4.1.3.2 Übertragungsfunktion Fehler (F2)

(**F2**) - Fehler, die aufgrund der Übertragung sowie von Timing und Datenintegrität auftreten können (das Bild kommt nicht, kommt spät, falsche Sequenz usw.):

- Die Daten kommen nicht an (Übertragungsfehler)
- Die Daten kommen zu spät an (Übertragungsverzögerung)
- Die Daten kommen nicht in der richtigen Reihenfolge an (Lieferung außerhalb der Bestellung)

Analysis - Hazards and Risks				
Funktion	**Hazard Description**	**Situation**	**Klassifizierung**	**Folgen**
		Anwesenheit	Anwesenheit detektiert	
Übertragung	Repetition - Unbeabsichtige Wiederholung	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet
	Deletion - Verlust der Daten	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet
	Insertion - Eingefühgt werden	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet
	Incorrect Sequence - Falsche Sequenz	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet
	Corruption	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet
	Delay - Inakzeptable Verzögerung der Daten	Objekt in VSA	JA	Ok - Keine Fehler
		Objekt in VSA	NEIN	Safety Funktion gefärdet

Abbildung 4.4: Analyse der Übertragungsgefährdungen.

4.1.3.3 Bildverarbeitungs-Fehler (F3)

(**F3**) - Fehler, die aufgrund von Funktionsfehlern der Kamera oder beim Bildklassifizierungssystem auftreten können. Es gibt zwei Fehlertypen, die auftreten können:

- Zufalls-Hardwarefehler
  Bildsensorprobleme können Fehler verursachen.

- Systematische Fehlern
  Probleme wie Rauschen im Bild, Verzerrung, Schatten, Blendung, Reflexion, niedrige Kontraste usw. können Fehler verursachen.

Ein Beispiel für Fehler, die bei der Klassifizierung auftauchen können, ist unten in der Tabelle dargestellt.

Analysis - Hazards and Risks				
Funktion	**Hazard Description**	**Situation**	**Klassifizierung**	**Folgen**
		Anwesenheit	**Anwesenheit detektiert**	
Bildverarbeitung	Kein Bild	Objekt in VSA	Keine Daten	Safety Funktion gefärdet
		Objekt nicht in VSA	Keine Daten	Safety Funktion gefärdet
	Statisches Bild	Objekt in VSA	Yes	Ok - Keine Fehler
		Objekt in VSA	No	Safety Funktion gefärdet
		Objekt nicht in VSA	Yes	Safety Funktion nicht gefärdet
		Objekt nicht in VSA	No	Ok - Keine Fehler
	Lichtverhältnissänderung	Objekt in VSA	Yes	Ok - Keine Fehler
		Objekt in VSA	No	Safety Funktion gefärdet
		Objekt nicht in VSA	Yes	Safety Funktion nicht gefärdet
		Objekt nicht in VSA	No	Safety Funktion nicht gefärdet
	Position des Objekts falsch erkannt	Objekt in VSA	Yes	Ok - Keine Fehler
		Objekt in VSA	No	Safety Funktion gefärdet
		Objekt nicht in VSA	Yes	Safety Funktion nicht gefärdet
		Objekt in VSA	No	Safety Funktion nicht gefärdet

Abbildung 4.5: Analyse der Fehler bei der Bildklassifizierung

4.1.4 Sicherheitsanforderungen (SR)

Aus der Analyse von Fehlern kommen die folgende Top-Level-Sicherheitskonzept (eng. Safety-Conzepts) und -Anforderungen (eng. Safety Requirements (SR)).

(SR1) - die Anforderungen sind schon für das Simatic-System (F-DI, F-DQ und F-CPU usw.) implementiert. Sie sind nicht wichtig weiter zu betrachten, aber wichtig im Gesamtpaket zu adressieren.

(SR2) - das Sicherheitssystem muss die typischen Kommunikationsfehler detektieren, die zur Verletzung der Sicherheitsfunktion führen können. Und das wären:

- Verlust von Nachrichten
- Wiederholung
- Korruption von Daten
- Falsche Sequenzierung
- Einfügen
- Adressieren

- Maskieren

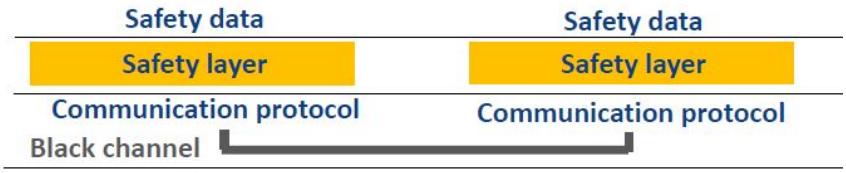
(SR3) - das Sicherheitssystem sollte den Sicherheitsstatus beibehalten, wenn während der Objekterkennung ein Fehler auftritt. Und es muss mit zwei Kategorien von Fehlern umgehen können:

- Zufalls-Hardwarefehler
- Systematische Fehler

4.1.5 Sicherheitsmaßnahmen (SM)

(SM1) - beim Simatic-System sind keine weiteren Sicherheitsmaßnahmen erforderlich, F-Geräte verfügen bereits über alle erforderlichen Maßnahmen.

(SM2) - um die Übertragung von Daten abzusichern, wird eine Black-Channel-Architektur vorgeschlagen werden. Daher können die kommunikationsbezogenen Fehler erkannt werden. Die Sicherheitsnorm IEC 61508-2:2010 enthält viele Anleitungen, in denen auch Black Channels bereitgestellt werden, die Unterweisung für den Benutzer wird auf IEC 61784-3 oder IEC 62280 (auch bekannt als EN 50159). Für das Black-Channel-Prinzip enthält EN 50159 eine Reihe von Bedrohungen und Erkennungsmechanismen gegen diese Bedrohungen [52], die in Abbildung 4.6 dargestellt sind.



Threat	Defenses			
	Sequence number	Time stamp	Time-out	Cryptographic techniques
Repetition	X	X		
Deletion	X			
Insertion	X			
Re-sequence	X	X		
Corruption				X
Delay		X	X	
Masquerade				X

Abbildung 4.6: Black-Channel-Prinzip

(SM3) - ohne weiteres ist das die größte Herausforderung für die Erweiterung des Sicherheitssystems. Wie erwähnt soll die Kamera fähig sein mit zwei Kategorien von Fehlern umgehen zu können. Dies sind Zufalls-Hardwarefehler [43] und systematische Fehler [59]. Was die relevanten Forschungsthemen sein könnten, wird im Kapitel 8 beschrieben.

KAPITEL 5

KOMPONENTEN DES DEMONSTRATOR SICHERHEITSSYSTEMS

Hier werden die Komponenten, mit denen das Sicherheitssystem realisiert wurde, beschrieben, im nächsten Kapitel sind die Implementierungsdetails zu finden.

5.1 Sps Simatic-System

5.1.1 Automatisieren mit speicherprogrammierbarer Steuerung

Unter dem Begriff Automatisieren in der Industrie versteht man das Steuern, Regeln und Überwachen von Anlagen und Maschinen. Durch Speicherprogrammierbare Steuerungen (SPS) können Automatisierungsfunktionen ausgeführt werden[57].

5.1.2 Aufbau der Simatic

Das Simatic-System ist das Automatisierungssystem von Siemens, das in den meisten Fällen modular aufgebaut ist und ein umfangreiches Baugruppenspektrum für die optimale Anpassung an die Automatisierungsaufgabe hat [36]. In dieser Arbeit werden Simatic-Komponenten verwendet, beispielsweise ein SPS vom Simatic-System, das zusammen mit den wichtigsten Bauteilen dargestellt wird.

5.1.2.1 Zentralbaugruppe eines SPS

1. **Stromversorgung** - versorgt das SPS Simatic-System mit Strom.

2. **Central Processing Unit (CPU)** - ist das 'Gehirn' oder die Zentraleinheit des SPS-Systems. CPU führt das Anwenderprogramm aus und arbeitet das SPS Programm sequentiell und zyklich ab [57]. Gleichzeitig besitzt es Kommunikationsschnittstellen, die z. B. eine

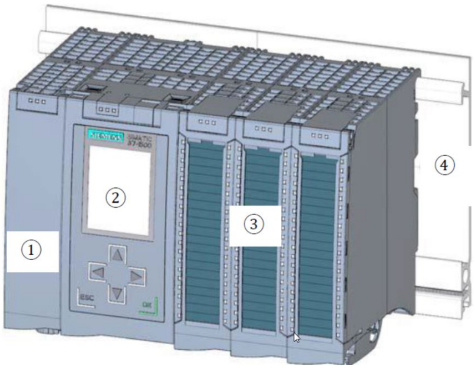


Abbildung 5.1: Zentralbaugruppe eines SPS.[71]

Anbindung zur Dezentralen Peripherie, mit dem Rechner oder OPC UA Server, ermöglichen. Als Speichermodul für CPU wird eine SD-Karte (Secure Digital Memory Carde) verwendet, worauf das Anwenderprogramm bei der Übertragung auf CPU geschrieben wird.

3. **Zentralepripherie Module** - dienen für Digitalein- bzw Digitalausgabe und Analogein- bzw. Analogausgabe.

4. **Profilschiene** - auf Profilschienen werden die Baugruppen montiert.

5.1.2.2 Dezentrale Peripherie

Dezentrale Peripherie - ist ein modulares dezentrales Peripheriesystem, bei dem es sich um eine Erweiterung der Zentralsteuerung handelt. Wegen großer Distanzen wird es oft benötigt, wenn die Signale vor Ort gesammelt werden müssen. Dafür ist die Dezentrale Peripherie sehr geeignet, die Signale vor Ort zu sammeln und über ein Bussystem oder Profinet (Process Field Network – LAN-system) in die Zentralbaugruppe CPU zu übertragen. Die Übertragung erfolgt über ein angeschlossenes LAN-Kabel zwischen Zentralbaugruppe CPU und Dezentralbaugruppe. Profinet ist das sogenannte industrielle Ethernet.

1. **Intermace Module** – ist die Schnittstelle für Datenaustausch zwischen Controller und Peripheriemodulen.

2. **Base Unit** - ist die Modulkomponente für elektrische und mechanische Verbindungs-I/O-Module.

3. **Base Unit**.

4. **Peripheriemodule**:

   DI - Digitaleingabe, bekommt 1-Bit Signale
   DQ - Digitalausgabe, sendet 1-Bit Signale

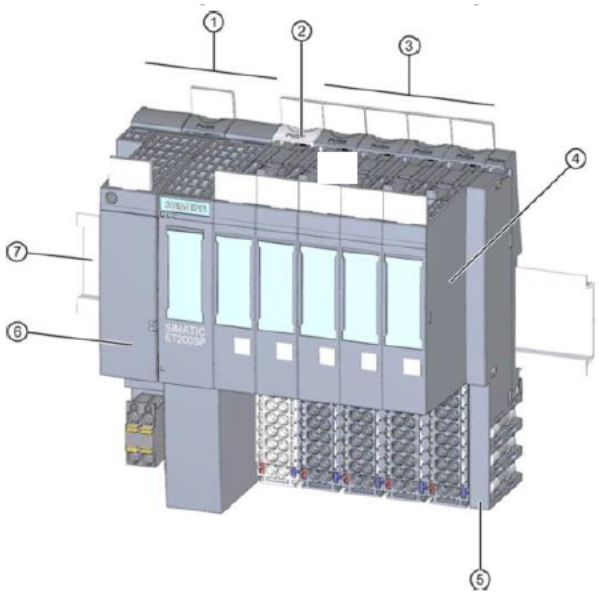


Abbildung 5.2: Dezentrale Peripherie [71]

AI - Analogeingabe, bekommt analoge Signale
AQ - Analogausgabe, sendet analoge Signale

5. **Servermodul** - das ist am Ende der Peripheriemodule und dient als Abschluss für die dDezentralen Peripheriemodule.

6. **Bus-Adapter** - für die Verbindung mit einer Zentralbaugruppe.

7. **Normalprofilschiene** - ist die Schiene, die am dDezentralen Peripheriesystem fixiert wird.

5.1.3 Fehlersichere Speicherprogrammierbare Steuerung

Für die Gewährleistung der funktionalen Sicherheit eines Sicherheissystem werden bei Maschinen oder Anlagen fehlersichere SPS-Systeme benötigt.

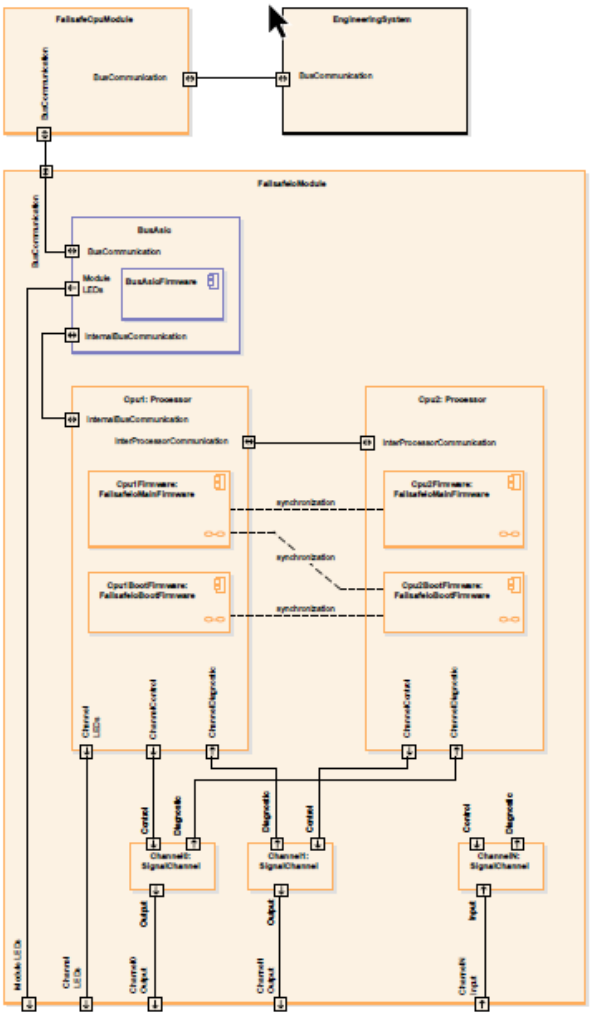


Abbildung 5.3: Interne Struktur eines fehlersicheren I/O-Moduls [89]

Die Hauptfunktionalität der Fail-Safe-Module (FSM) ist, fehlersichere physikalische Input- oder Output-Signale zu verarbeiten. Siemens AG (DF FA Graz) entwickelt Firmware für solche fehlersicheren Module. Ein FSM besteht aus fehlersicherer-Firmware bzw. -Hardware und verfügt über einen redundanten Aufbau. Die Redundanz sorgt dafür , dass im Fehlerfall die Ausfallwahrscheinlichkeit verringert wird, die zu einer nicht-korrekten Funktion des Sicherheitssystems führen kann [57], das ist so genannte 1oo2 Prinzip. In der folgenden Abbildung 5.3 wird ein Beispiel der internen Struktur fehlersicherer I/O-Module (FSM) dargestellt, in dem einige wichtige Merkmale der FSM visualisiert sind [47].

BusAsic - ein BusAsic wird in vielen Fällen für den Datenaustausch zwischen interner und externer Kommunikation verwendet.

Prozessoren - sind die zentrale logische Einheit einer FSM. Sie werden von einer fehlersicheren Fimware angesteuert und verwalten die Signalkanäle sowie die externe Kommunikation.

SignalChannel - Signalkanäle, die fähig sind elektronische Ausgangssignale zu erzeugen oder Eingangssignale zu 'spüren'.

5.1.4 Funktions- und Arbeitsprinzip der SPS

Das Anwenderprogramm besteht innerhalb von SPS aus einer Liste von Anweisungen, welche in einem Engineering-Tool programmiert wird. In dieser Arbeit wird das TIA Portal von Siemens verwendet, das ein Engineering-Tool ist. Die Programmiersprache, in der das Anwenderprogramm geschrieben wird, wird nicht direkt in die SPS übertragen.

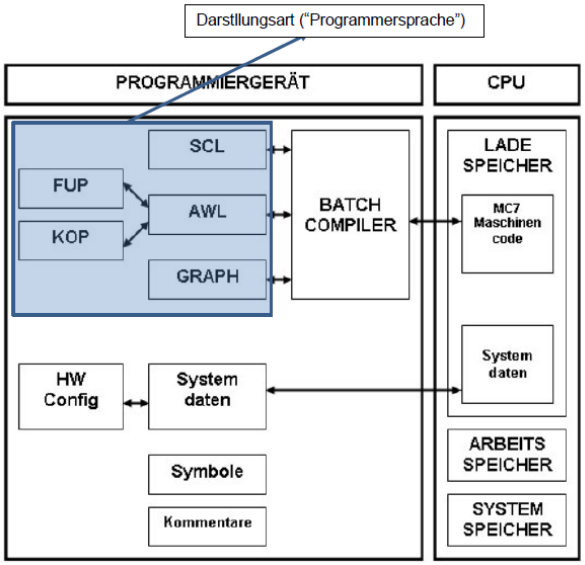


Abbildung 5.4: Funktionsweise zwischen Programmiergerät und CPU [49]

Die Übertragung erfolgt, wenn das Programm für einen Maschinencode generiert wird. Die Programmumwandlung in einen geeigneten Maschinencode wird durch einen Compiler realisiert. Das Anwendungsprogramm wird zyklisch und sequentiell vom CPU abgearbeitet. Zu Beginn des Zyklus werden aktuelle Zustände der Eingänge der Peripherie eingelesen. Am Ende werden die abgearbeiteten Anweisungen in die Ausgangsperipherie geschrieben und der Zyklus beginnt von vorne.

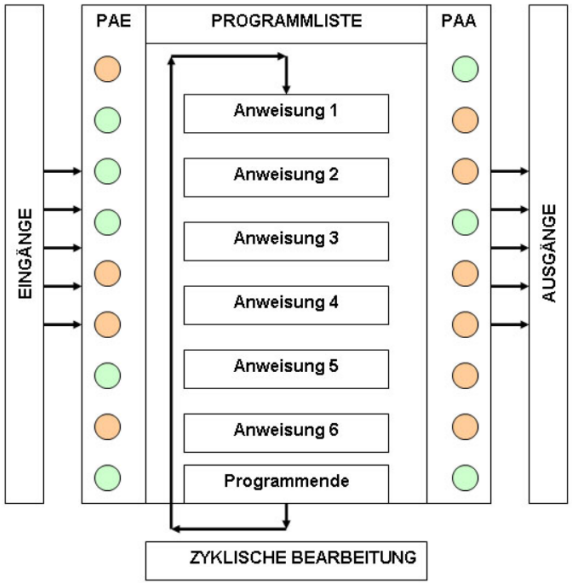


Abbildung 5.5: Zyklischer und sequentieller Ablauf eines SPS-Programms [49]

5.2 Übertragungsfunktion

Durch die zunehmende Digitalisierung wird die reibungslose Kommunikation zwischen verschiedenen Geräten und Maschinen (Abb. 5.6)) von diversen Herstellern immer wichtiger. OPC UA (Open Platform Communications Unified Architecture) etabliert sich dabei immer mehr als offener Industrie-4.0-Standard. OPC UA ist in IEC 62541 als offener Kommunikationsstandard spezifiziert worden und nicht als weitere industrielle Kommunikationslösung [13]. Dadurch wird der sichere und standardisierte Datenaustausch von unterschiedlichen Herstellern zwischen Steuerungen oder IT-Systemen möglich [19].

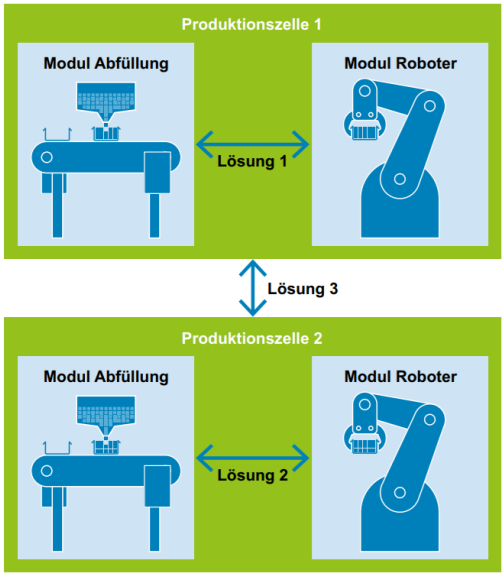


Abbildung 5.6: Industrielle Kommunikation von Industrie 4.0 [13]

Besonders geeignet ist es wegen seiner Plattformunabhängigkeit, Skalierbarkeit, Erweiterbarkeit und seines sicheren Übertragungsverfahrens. Bei der Implementierung werden Informationsmodelle gestellt, die nicht nur die Daten bereitstellen, sondern auch zusätzliche Informationen von den Daten (z. B. Datentypen) [7]. Diese semantischen Informationen bilden eine Struktur aus sogenannten Knoten bzw. Kanten, die als Informationsmodel bezeichnet werden. Darauf basiert der Adressraum eines OPC UA Servers.

5.2.1 OPC UA Server

OPC UA verfügt über ein System, das als Server arbeitet und die vorhandenen Informationen anderen Systemen (Clients) zur Verfügung stellt. Ein System kann zugleich Server oder Client sein [7]. Die Daten, die vom Server zur Verfügung gestellt werden, sind in Form von Knoten (eng. Nodes) angeordnet. Die Beziehung zwischen Knoten ist hierarchisch gestaltet. Die Knoten können ein Objekt, eine Methode oder eine Variable beinhalten. Das Netzwerk von Knoten ist so angeordnet, dass ausgehend von der Wurzel (eng. Root) alle gewünschten Knoten gangbar sind. Der Client kann auf die verfügbaren Daten, die im Server vorhanden sind, so zugreifen, dass er diese lesen und schreiben kann. Neben Variablen ist auch ein Methodenaufruf möglich. In dieser Arbeit wird ein S7 1500 CPU verwendet, das über einen OPC UA Server verfügt. Somit wird es möglich sein, die Daten von der Kamera auf S7 1500 CPU zu übertragen. In der folgenden Grafik wird gezeigt, wie eine Methode im Server (Server-Methode Coolïm S7 1500 OPC UA) von OPC UA Client aufgerufen wird. Die Konfiguration wird dann unter Kapitel 6.3 Übertragungsfunktion erklärt.

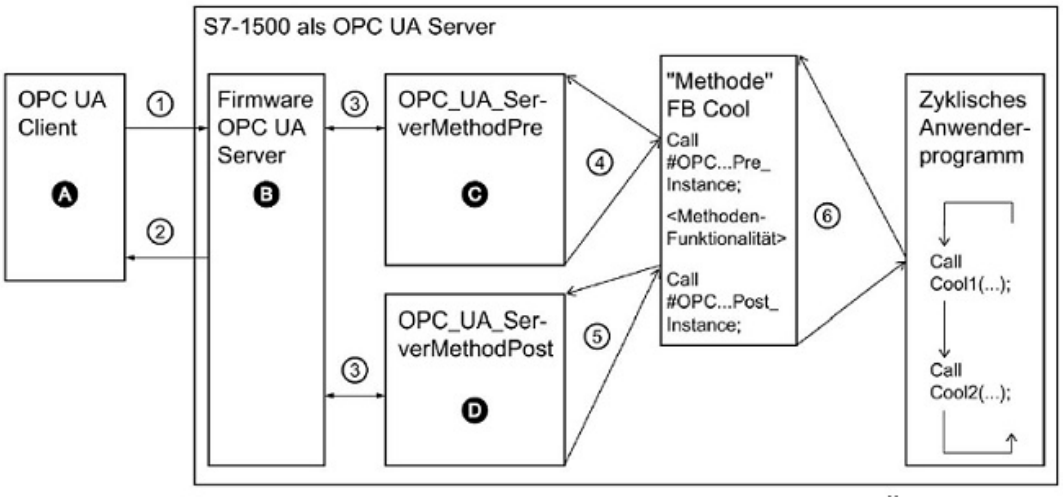


Abbildung 5.7: Server-Methode "Cool" wird von Client aufgerufen [7]

5.2.2 Black Channel

Die sichere Übertragung der Daten ist ein wichtiger Teil der gesamten Sicherheits-Kette. Eine sichere Übertragung ohne zusätzliche Verdrahtung ist schon durch Einsetzen vom Black-Channel-Prinzip möglich [18]. Dieses Prinzip verwendet das gleiche Übertragungsmedium (z. B. Standard-Industrielles-Ethernet) [4], um die Standard-Daten und Sicherheits-Signale zu übertragen.

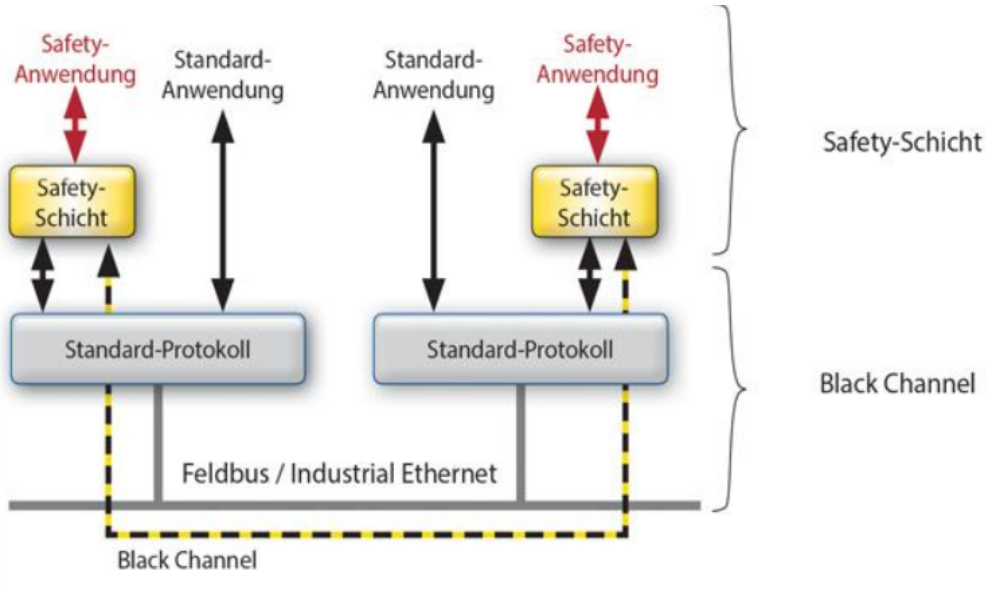


Abbildung 5.8: Black-Channel-Prinzip [6]

Wie auch in Abbildung 5.8 zu sehen ist, sind die Safety-Schicht (oder Safety Layer) oberhalb von Black Channel angeführt. Das ist eine Sicherheitsfunktion zur Kommunikation sicherheits-

relevanter Daten unter Verwendung von vorhandenen, nicht sicherheitsrelevanten Feldbusarchitekturen. Den Safety Layers ist der Kommunikationsmechanismus unbekannt und sie werden unter der Sicherheitsfunktion (Safety-Layer) als Black Channel (Black Box) betrachtet [10]. Die Prüfmechanismen sind im Safety-Layer hinterlegt und sorgen dafür, dass die Verletzung der funktional sicheren Kommunikation bei der Übertragung vermieden wird. PROFIsafe verwendet das gleiche Prinzip und ist seit 2007 internationale Norm IEC 61784-3. Die Implementierung des schon am Markt vorhandenen PROFIsafe würde die Übertragungsfunktion zur funktional sicheren Kommunikation machen. Die Sicherheitsnorm IEC 615018-2:2010 enthält viele Anleitungen, in denen auch Black Channels bereitgestellt werden, und die Verweisung für den Benutzer wird auf IEC 61784-3 oder IEC 62280 [52] gegeben.

5.3 Bildverarbeitung

Die Erfassung und Verarbeitung der Bilder kann als sehr wichtiges Element bei der Identifizierung bestimmter Informationen in Industrie 4.0 sein. Die Verarbeitung von Signalen, die Digitalbilder repräsentieren, wird als Bildverarbeitung verstanden. Die Digitalbilder werden vom Computer als Matrix oder multidimensionale Array repräsentiert. Das heißt, jedes Pixel hat eine Koordinate (0,0) und bei digitalen Farbbildern stehen noch die Farbraum-Werte sowie RGB (Rot Grün Blau) oder HSV (Hue-Farbwert, Saturation-Farbsättigung, Value-Hellwert) zur Verfügung [32].

5.3.1 OpenCV

OpenCV (Open Source Computer Vision Library) ist eine Open-Source-Bibliothek für Bildverarbeitung und maschinelles Sehen [3]. OpenCV wurde entwickelt um eine gemeinsame Infrastruktur für die Bildverarbeitungsanwendung bereitzustellen und die Verwendung der Maschinenwahrnehmung in kommerziellen Produkten zu beschleunigen. Die Bibliothek verfügt über 2500 optimierte Algorithmen, die einen umfassenden Satz von klassischen und modernen Algorithmen für Bildverarbeitung und maschinelles Sehen enthalten. Diese Algorithmen können verwendet werden, um Gesichter zu erkennen, Objekte zu identifizieren bzw. auch um bewegte Objekte zu verfolgen. OpenCV verfügt über C++, Python, Java und MATLAB-Schnittstellen und unterstützt Windows, Linux, Android und Mac OS.

5.3.2 Raspberry PI

Raspberry Pi ist ein Einplatinencomputer, der ursprünglich als Lernplatform entwickelt wurde [39] (eine Abbildung eines Raspberry Pi 1 Model B findet sich in Abbildung 5.9). Trotz seiner kleinen Abmessungen ist eine ziemlich starke Hardware auf der Platine verbaut. So ist es möglich ein Betriebssystem darauf zu installieren und das Ganze mit Tastatur, Maus und HDMI zu verbinden und als Rechner zu verwenden. Seine Einsatzmöglichkeiten sind äußerst

vielfältig und er findet in verschiedenen elektronischen Projekten (vor allem im Kontext von IoT) Anwendung [30]. In dieser Arbeit wird der Raspberry Pi verwendet, um die Bildverarbeitung und LED-Steuerung im Demonstrator zu realisieren.

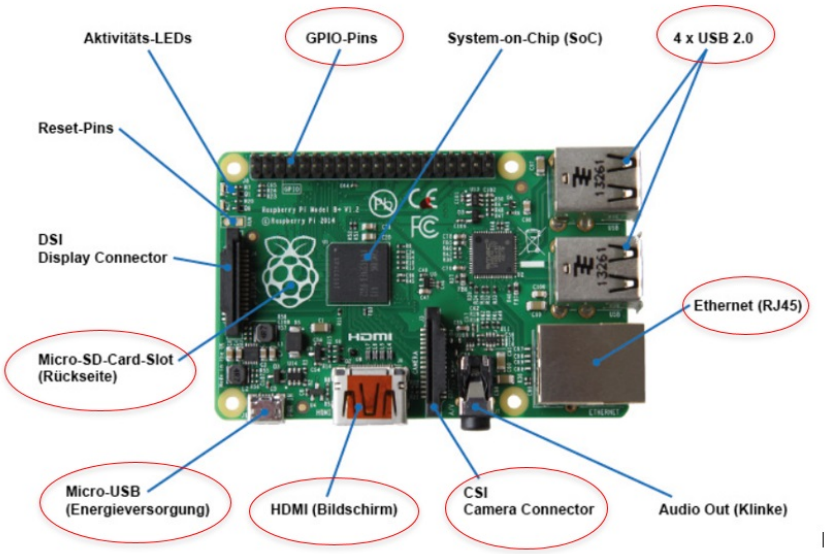


Abbildung 5.9: Ansicht von oben Raspberry Pi B+ [12]

Die Teile des Raspberry Pi, die in dem Demonstrator zum Einsatz kommen, sind in Abbildung 5.9 rot umrandet und bedürfen keiner weiteren Erklärung. Neben dem Raspberry Pi wurde für dieses Projekt eine Kamera (siehe Abbildung 5.10) verwendet, die mit Raspbery Pi über CSI Camera Connector (siehe 5.9) verbunden wird.

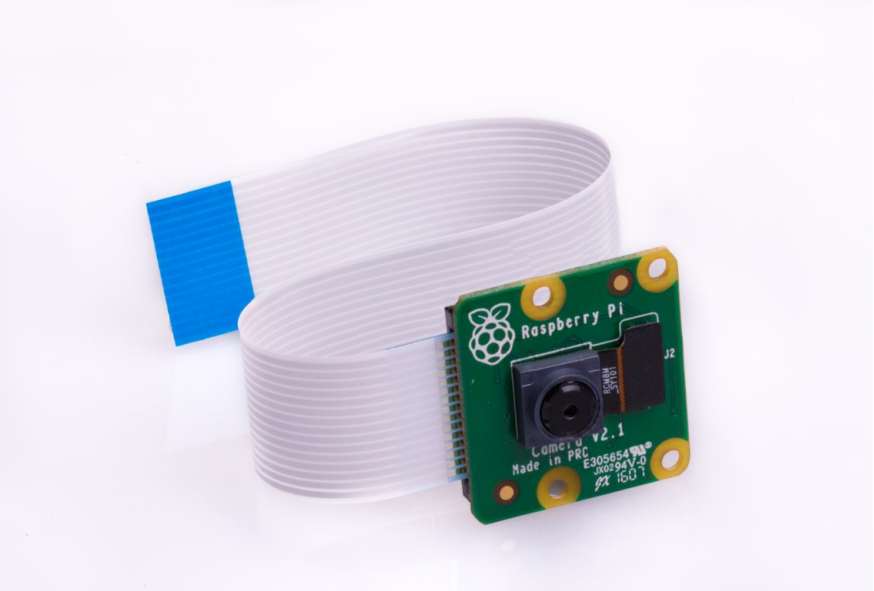


Abbildung 5.10: Kamera von Raspberry Pi [51]

KAPITEL 6

DYNAMIC SAFETY DEMONSTRATOR IMPLEMENTIERUNG

Dieses Kapitel befasst sich mit dem Thema der praktischen Implementierung des Demonstrators. Bei der Implementierung wird die simulierte industrielle Arbeitsumgebung mit dazugehörigen Bauteilen sowie die Software von SPS Simatic und das Kamerasystem beschrieben. Außerdem werden die wichtigsten Hardwarekonfigurationen erläutert. Die praktische Implementierung des Demonstrators erfolgte bei Siemens und bei Pro2Future auf der Techischen Universität Graz. Die simulierte industrielle Arbeitsumgebung und Simatic-System wurden bei Siemens aufgebaut und das Kamerasystem (Kamera, Raspbery Pi, Bildverarbeitung usw.) hauptsächlich bei Pro2Future.

6.1 Aufbau der simulierten industriellen Arbeitsumgebung

Die simulierte industrielle Arbeitsumgebung besteht aus einem so genannten "industriellen Basisfeld" und den dazugehörigen Tastern und LEDs. Als "industrielles Basisfeld" werden eine Holzplatte verwendet, die auf drei Latten steht. Diese sind unterhalb der Holzplatte befestigt, wodurch die ganze Verdrahtung des Aufbaus erfolgt. Seitlich sind die Taster und LED-Türme montiert. Wegen der Mobilität wurden geeignete Befestigungsbehälter für die Taster hergestellt, deren Herstellung bei Siemens Graz (durch 3D Drucker) erfolgte. Wie auch in der Abbildung 6.1 zu sehen ist, sind die meisten Bauteile seitlich montiert. Bei der Positionierung von Not-Aus Taster ist es sehr wichtig, dass sie schnell, einfach und gefahrlos erreichbar sind. Deswegen sind für jeden VSA zwei Not-Aus Taster eingebaut. Nur die RGB-LEDs liegen direkt auf dem Basisfeld, weil diese durch die Kamera getestet werden. Für eine detaillierte Erklärung siehe Kapitel 6.4.

Abbildung 6.1: Demo auf der EU ICT2018 Conference

6.1.1 Start-, Stop-Taster

- **Start-Taster** - ist mit der dezentralen Peripherie ET200SP DI Modul über Base Unit verbunden und sorgt dafür, den Demonstrator zu starten/aktivieren.

- **Stop Taster** - ist ebenso mit der dezentralen Peripherie ET200SP DI Modul über Base Unit verbunden und sorgt dafür den Demonstrator zu stoppen/abschalten.

Tabelle 6.1: Not-Aus, Start-, Stop-Taster

Beschreibung	Bauteilbezeichnung	Anzahl	MLFB
Taster	Not-Aus-Taster	6	3SU1851-0NB00-2AA2
	Start-Taster	1	3SU1801-0AB00-2AB1
	Stop-Taster	1	3SU1801-0AC00-2AB1

6.1.2 Not-Aus-Taster

Not-Aus-Taster werden verwendet, wie am Anfang erwähnt wurde, wenn im Laufe einer Zusammenarbeit Fehler auftreten. Dann ist die gewöhnliche Reaktion der Arbeiter diesen zu drücken, um weiteren Schaden für beteiligte Menschen und Systeme zu verhindern. Das heißt, wenn nun so ein Taster in einer VSA gedrückt wird, wird die Sicherheit nur für die Objekte, die sich vorübergehend in diesem Bereich befinden, gewährleistet. Für einen hohen Sicherheitslevel werden Not-Aus-Taster zwei-kanalig ausgelegt. Die Überwachung auf Diskrepanz [1] und Querschluss [2] erfolgt durch das SPS Simatic-System.

Abbildung 6.2: Not-Aus-Taster und Montage

[1] Wenn zwei Eingangskanäle unterschiedliche Werte liefern, die nicht im Toleranzbereich sind.
[2] Quer- bzw. Kurzschlüsse zwischen zwei Eingangskanälen.

6.1.3 Signalleuchten

Bei dem Demonstrator werden keine Roboter verwendet, sondern Objekte, die einen Roboter oder den Menschen (Arbeiter) repräsentieren. Das wird noch detaillierter unter Kapitel 6.5 beschrieben. Da keine echten Auswirkungen demonstriert werden können, werden Signalleuchten benutzt, um die Zustände bzw. Zustandsauswirkungen für die VSA (Virtuelle Safety Areas) zu visualisieren. Dafür verwendet man in diesem Aufbau die LED-Türme von Siemens.

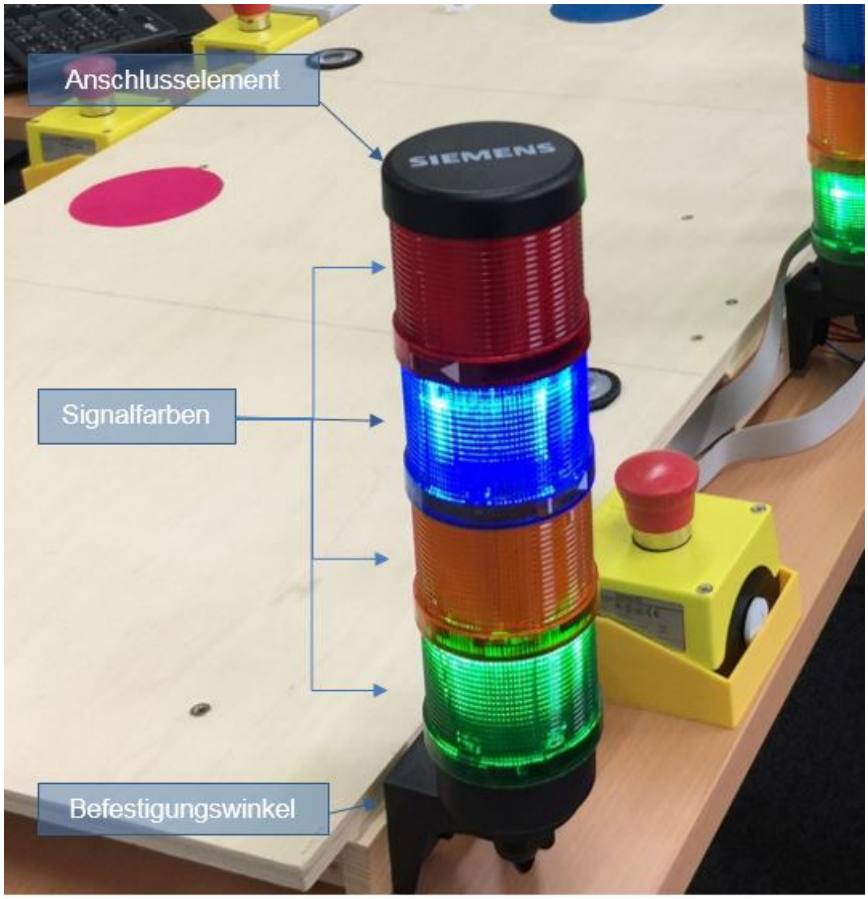


Abbildung 6.3: Signalleuchte - LED Türme

Wie auch bei der Abbildung 6.3 zu sehen ist, hat jeder LED-Turm (Tabelle 6.2) vier verschiedene LED-Farben (Grün, Gelb, Blau und Rot). Durch die Farben und die Farbenkombination kann jede VSA acht Zustände haben. Die Bedeutung und die Beschreibung aller Zuständen siehe die Abbildung 6.30.

Tabelle 6.2: Signalleuchten

Beschreibung	Bauteilbezeichnung	Anzahl	MLFB
LED-Türme	Licht grün	3	8WD4420-5AC
	Licht gelb	3	8WD4420-5AD
	Licht blau	3	8WD4420-5AF
	Licht rot	3	8WD4420-5AB
Winkel	Befestigungswinkel	3	8WD4408-0CD
	Anschlusselement	3	8WD4408-0AB

6.2 SPS Simatic-System Aufbau

Die nächsten und sehr wichtigen Punkte in diesem Kapitel sind die speicherprogrammierbaren SPS Simatic-Komponenten. Hierbei wird auf Hardware bzw. Hardwarekonfiguration und Software näher eingegangen.

Abbildung 6.4: CPU S7 1500 Simatic

6.2.1 Hardware

Die anspruchsvolle Industrie 4.0 erwartet durchgängige, skalierbare und zukunftsichere SPS-Automatisierungssysteme. Dafür braucht man Systeme, die für solche Herausforderungen intelligente Antworten geben. SPS Simatic ist die richtige Lösung für solche technologischen Aufgaben. In dieser Arbeit werden SPS Simatic-Systeme von Siemens verwendet. Simatic S7 1500 ist die CPU der Zentralbaugruppen. Wie auch unter Kapitel 4 Theoretische Grundlagen erläutert wurde, ist CPU die Zentraleinheit eines SPS-Systems und ermöglicht auch die Kommunikation mit anderen Automatisierungskomponenten. Die hohe leistungsfähige S7 1500 CPU und das Stromversorgungsmodul sind in Abbildung 6.4 zu sehen. Diese CPU verfügt über ein Display, auf dem Informationen z. B. zur Fehlermeldung, zum Firmwarestand oder zu angeschlossenen Modulen stehen. Durch das PROFINet-Kabel wird Dezentrale Peripherie der CPU zugeordnet.

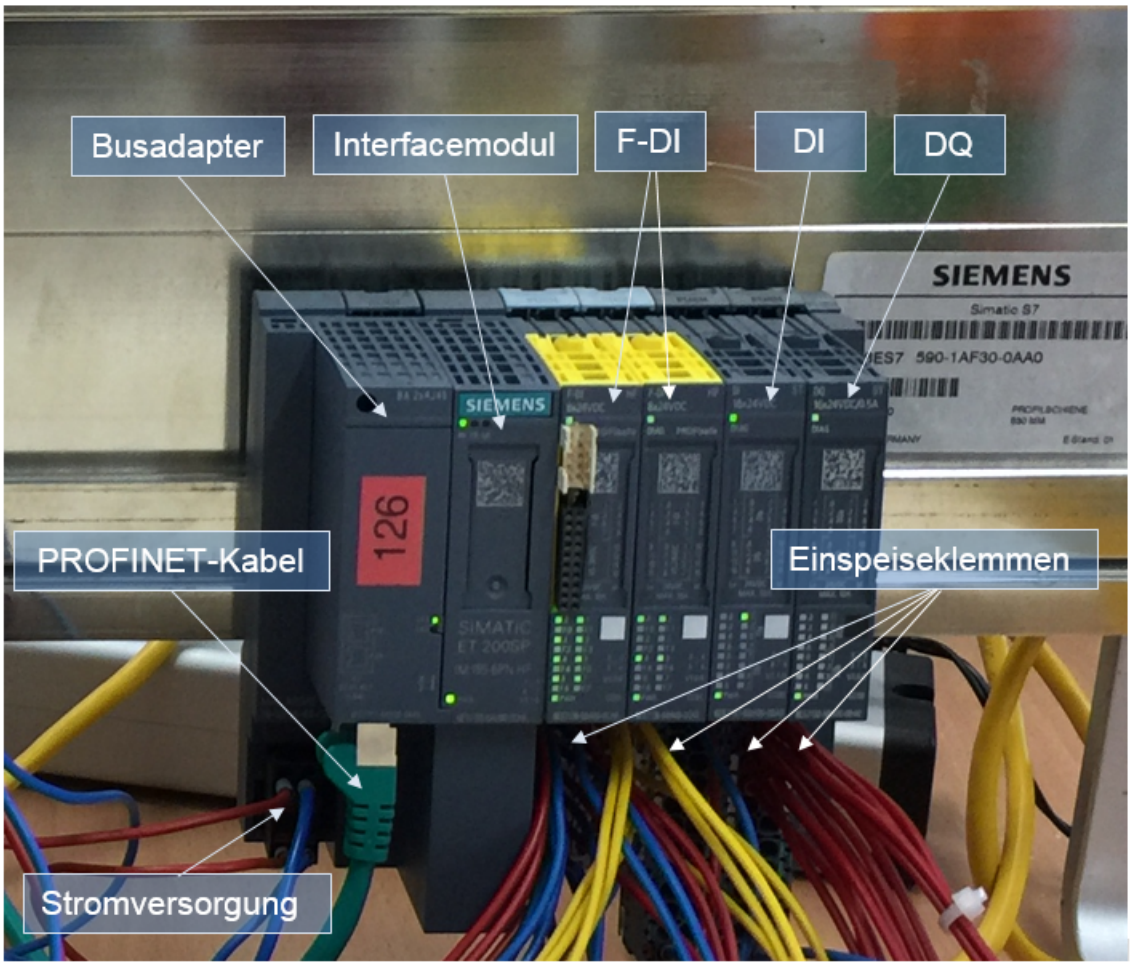


Abbildung 6.5: ET200SP Dezentrale Peripherie

Die Familienbaugruppe ET200SP gehört zur so genannten Dezentralen Peripherie und ist für den direkten Einsatz auf Maschinen, Sensorik bzw. Aktuatorik geeignet, dabei ist die Durch-

führung der erfassten bzw. ausgegebenen Signale vor Ort wichtig. Der modulare Aufbau erlaubt eine leichtere Erweiterung. Siemens Graz entwickelt fehlersichere Firmware für diese Familienbaugruppe. Dabei kann es um digitale und analoge Eingabemodule sowie Ausgabemodule gehen kann. Um leichter zu erkennen, steht darunter in Abbildung 6.5 die Dezentrale Peripherie. Wobei alle dazugehörigen Teile, Module und die Verbindung über PROFINET-Kabel mit CPU gekennzeichnet sind. Ein Switch wird verwendet,um mehrere Anschlüsse sowie den, der Dezentralen Peripherie und des OPC UA Server zum Kamerasystem mit dem CPU zu verbinden.

Wie zu sehen ist, sind in der Tabelle 6.3 die Bauteile aufgelistet, die in diesem Demonstrator verwendet werden. In der Tabelle sind die Beschreibung, Bauteilbezeichnung, die Anzahl und Maschinenlesbare Fabrikate-Bezeichnung (MLFB) angeführt.

Tabelle 6.3: Hardware-Komponenten

Beschreibung	Bauteilbezeichnung	Anzahl	MLFB
Schiene	Profilschiene	1	6ES7590-1AF30-0AA0
CPU	Simatic S7-1500F, CPU 1516F-3 PN/DP	1	6ES7516-3FN01-0AB0
Speicherkarte	SIMATIC S7, Memory Card, für S7-1x00 CPU/SINAMICS, 3, 3V Flash, 24 MByte	1	6ES7954-8LF03-0AA0
Stromversorgung	Simatic PM 1507 25 V/8 A	1	6EP1333-4BA00
Interface Modul	Simatic ET 200SP, Profinet Interface-Modul	1	6ES7155-6AU00-0CN0
Bus Adapter	SIMATIC ET 200SP,Busadapter BA 2xRJ45, 2 RJ45 Buchsen für PROFINET	1	6ES7193-6AR00-0AA0
F-DI	Simatic DP, Elektronikmodul für ET200SP, F-DI 8x14VDC HF	2	6ES7136-6BA00-0CA0
DI	Simatic DP, Elektronikmodul für ET200SP, DI 16x24V DC ST	1	6ES7131-6BH01-0BA0
DQ	Simatic DP, Elektronikmodul für ET200SP, DQ 16x24VDC/0.5A ST	1	6ES7132-6BH01-0BA0
Server-Modul	Simatic ET 200SP, Ersatzteil Server-Modul für ET200SP	1	6ES7193-6PA00-0AA0
Base Unit weiß	Simatic ET 200SP, Base-Unit BU15-P16+A0+2DP	1	6ES7193-6BP00-0DA0
Base Unit schwarz	Simatic ET 200SP, Base-Unit BU15-P16+A0+2B	2	6ES7193-6BP00-0BA0

Die Bedeutung und die Funktion der in Tabelle 6.3 zugeordneten Bauteile wird kurz beschrieben.

- Schiene - steht für die Befestigung der SPS-Komponenten.
- CPU - ist die Zentraleinheit der SPS-Komponenten und führt das Anwenderprogramm aus.

- Speicherkarte - damit CPU laufen kann, ist die Simatic Memory Card erforderlich. Das Anwenderprogramm wird auf die Simatic Memory Card geschrieben vor einer Übertragung in die CPU.
- Stromversorgung - dieses Modul versorgt das gesamte Simatic-System mit Strom.
- Interface Module - ist eine Schnittstelle für die Verbindung der Dezentral Peripherimodule mit CPU.
- Bus Adapter - ist eine Kommunikationsschnittstelle für PROFINET an Interface Modulen.
- F-DI - das fehlersichere Digitaleingabemodul nimmt auf und verarbeitet fehlersichere physikalische Eingangssignale, z. B. bei der Überwachung von Not-Aus-Tastern erkennt es sofort, wenn dieser gedrückt wird.
- DI - Digitaleingabemodul kann physikalische Signale aufnehmen und verarbeiten.
- DQ - steht für Digitalausgabemodul und kann digitale Signale ausgeben/senden. Z. B. LED-Türme werden damit ein- bzw. ausgeschaltet.
- Server-Modul - dient als Abschluss für die Dezentrale Peripherie, d. h. wird ans Ende der Peripheriemodule gesteckt.
- Base Unit - dient als Modulkomponente für elektrische und mechanische (I/O-Module) Verbindungen.

6.2.2 Software

In diesem Unterkapitel werden die Software des SPS Simatic-Systems, die notwendigen Hardwarekonfigurationen sowie die verwendeten Programmiersprachen beschrieben. Für die Programmierung bzw. Projektierung des Sps Simatic ist STEP 7 notwendig. STEP 7 ist eine Software, die diesen Zweck erfüllt. Dafür gibt es dieverse Software sowie WinCC für die Visualisierung oder CFC für graphische Projektierung usw.(ist für uns nicht so wichtig, um es genau zu wissen). Das TIA (Totally Integrated Automation)-Portal ist das Engineerin-Tool (oder Arbeitsumgebung),das beim Programmieren der SPS verwendet wird. Im Gegensatz zu STEP 7 handelt es sich beim TIA-Portal um eine Weiterentwicklung dieser (STEP7), welche mehr oder weniger die bisherige Software von STEP 7 und diverse zusätzliche Software integriert. TIAPortal bietet einen vollständigen Zugriff auf die gesamten Automatisierungsprojekte, bei denen die Konfiguration des Simatic-Systems bzw. Programmierung erfolgen wird. Für das Projekt wird TIA Portal V15 (Version) verwendet und noch andere Software-Komponenten, die in Tabelle 6.4 aufgelistet sind. Safety Advanced ermöglicht es das Sicherheitsprogramm zu implementieren. STEP 7 (TIAPortal) bietet verschiedene Programmiersprachen.

Tabelle 6.4: Software-Komponenten

Software	Anzahl	Artikelnummer	Hinweise
SIMATIC STEP 7 Professional	1		TIA V15
SIMATIC STEP7 Safety Advaned	1		TIA V15

In dieser Arbeit werden die grafischen Programmiersprachen FUP (Funktionsplan) und SCL (Structured Control Language) verwendet. FUP ist eine sogenannte graphische Sps-Programmiersprache und im Gegensatz zu anderen Programmiersprachen übersichtlicher und schnell nachvollziehbar.

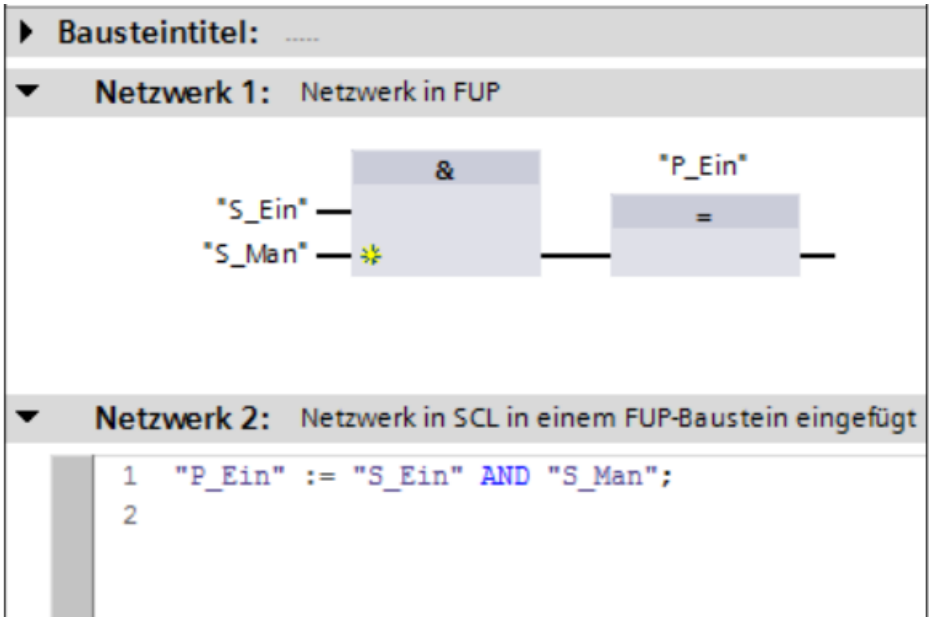


Abbildung 6.6: Beispiel mit FUP und SCL [50]

SCL ist mehr eine strukturierte SPS-Programmiersprache und hat Ähnlichkeit mit der Programmiersprache C Syntax. Als Nächstes werden die Konfiguration der Hardware sowie das Anwenderprogramm (SPS-programmierte Software) beschrieben. Wie schon erwähnt wird das TIA Portal dafür verwendet.

6.2.2.1 Hardwarekonfiguration

Im Folgenden ist im TIAdas Projekt (Dynamic Safety Demonstrator) im TIA Portal angelegt und unter Netzwerkansicht Abbildung 6.7 ist die hinzugefügte Hardware zu sehen. Wie in der Abbildung 6.7 dargestellt, befindet sich auf der linken Seite CPU und auf der rechten Seite die Dezentrale Peripherie.

Um die Dezentrale Peripherie zu verdeutlichen, wird in der Abbildung 6.8 die Konfiguration in TIA Portal gezeigt.

Es ist auch zu sehen, dass auf der Dezentralen Peripherie zwei fehlersichere Digitaleingabemodule (F-DI), ein Digitaleingabemodul (DI) und ein Digitalausgabemodul (DQ) gesteckt

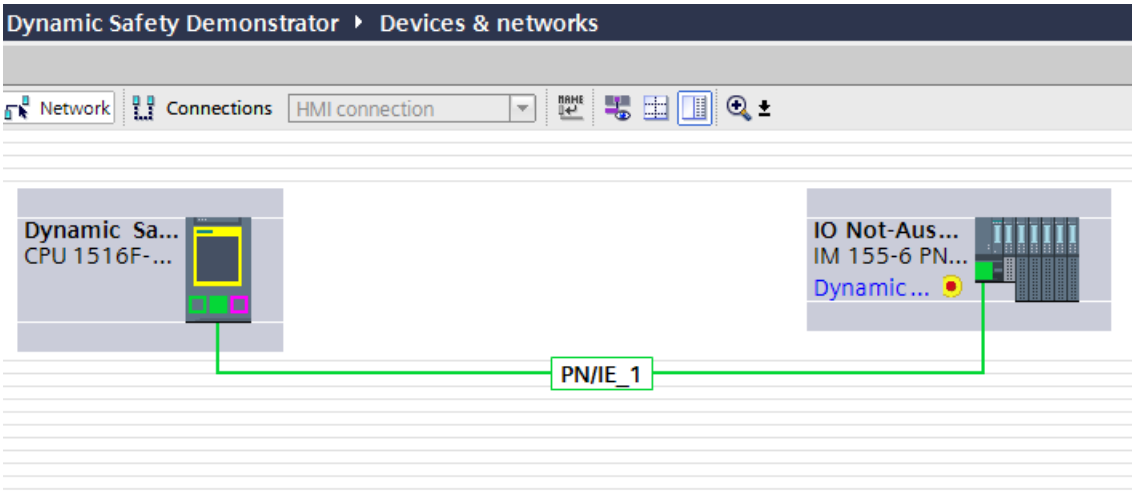


Abbildung 6.7: Grafische Netzwerkansicht

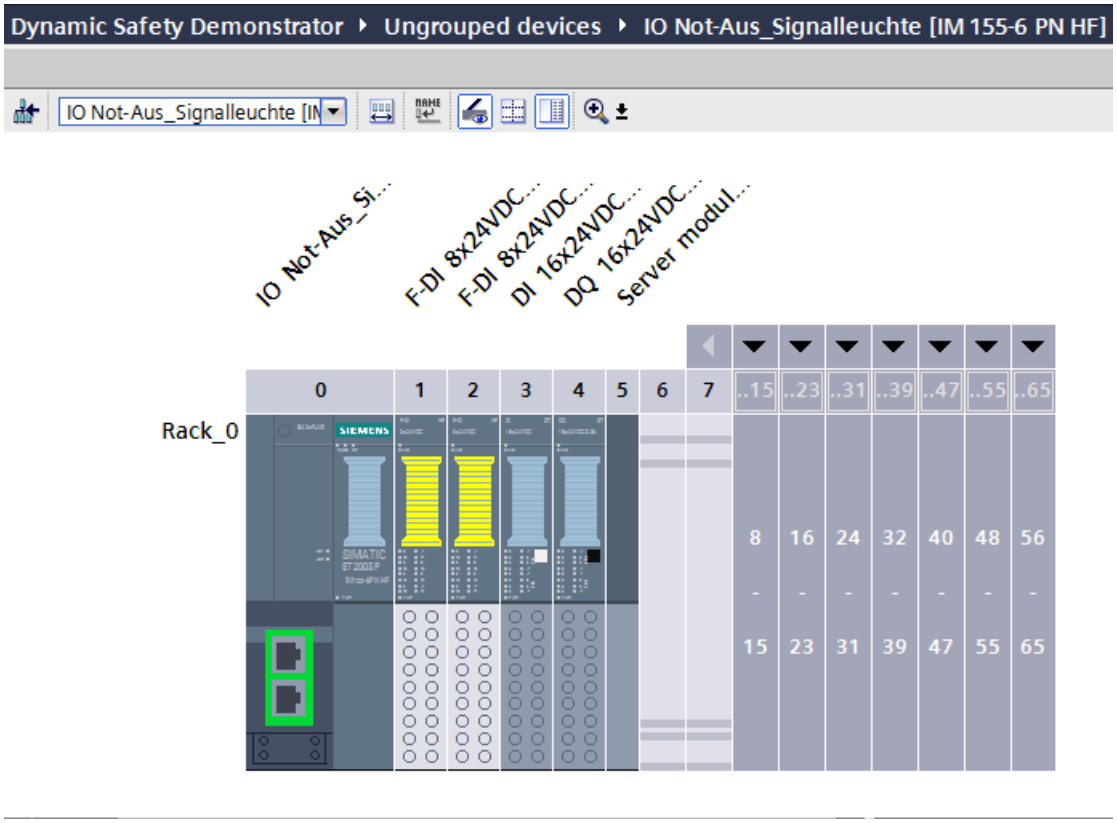


Abbildung 6.8: Konfiguration der Dezentralen Peripherie

sind. Dadurch können alle Taster und LED-Türme gesteuert werden. Dafür muss man vorher auch die erforderlichen Modulkonfigurationen durchführen. Durch Doppelklick auf das Modul werden die Eigenschaften des Moduls ersichtlich. Für dieses Projekt sind zwei Eigenschaften

besonders wichtig. Eine ist die Kanalparametrierung und die andere die Bekanntgabe von Ein- und Ausgängen. Die Kanalparametrierung findet unter "General"-> "Channel parameters ". Diese Konfiguration von Parametern, die in der folgenden Abbildung verdeutlicht sind, ermöglicht die redundante Verdrahtung (1oo2-Prinzip) und die Aktivierung der Sensorversorgung.

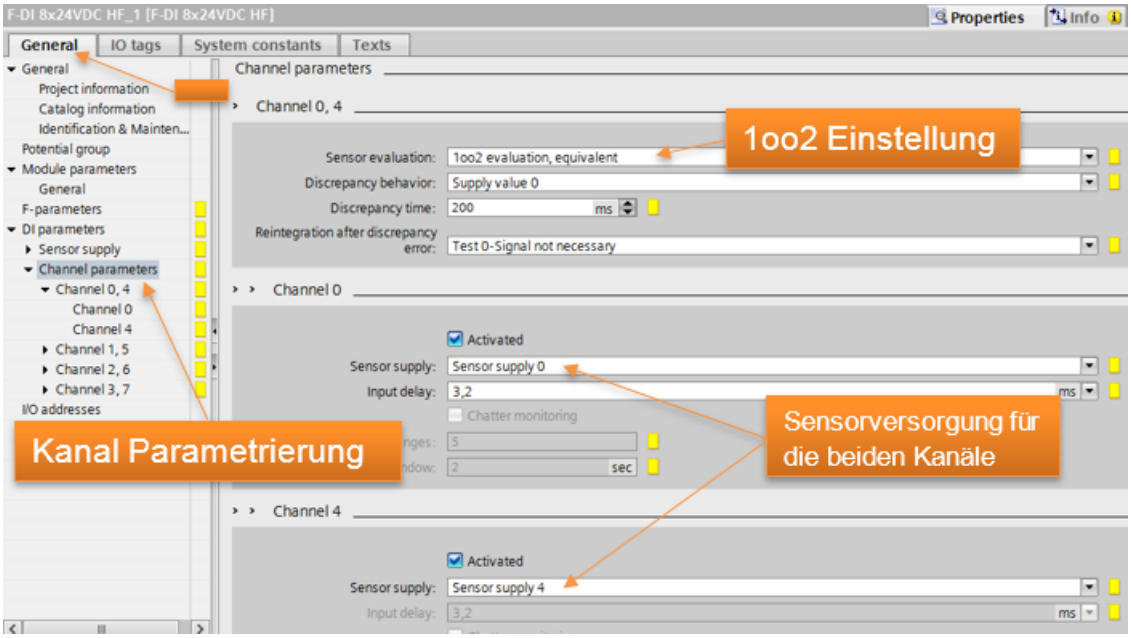


Abbildung 6.9: Konfiguration eines Moduls im TIA Portal.

Das 1oo2-Prinzip wird (im TIA Portal) als ein Kanal gesehen, aber mit einer Fehlertoleranz. Das heißt, wenn einer der beiden Schaltern versagt, ist der andere verfügbar. Anders gesagt, die funktionale Sicherheit wird nicht verletzt durch einen einzigen Fehler. Die Sensorversorgung "Sensor supply "ist eine spezielle Versorgungsspannung, die vom Modul über Base Unit zur Verfügung gestellt wird. Der Not-Aus-Taster wird von "Sensor supply "versorgt und ihr Zustand vom Eingang des Moduls (in unserem F-DI) gelesen. Das gleiche Prinzip wird bei allen Not-Aus-Tastern implementiert. Eine genaue und detaillierte Verschaltung des Not-Aus-Tasters zeigt die Abbildung 6.10. Außerdem ist es wichtig, jedem Ein- bzw. Ausgang einen aussagekräftigen Namen zu geben, damit später beim Programmieren ein eindeutiger Zugriff möglich ist. Die Benennung bzw. die Zuweisung erfolgen am einfachsten über 'IO tags' (neben General). Wie auch unter Abbildung 6.12 zu sehen ist, wird der erste Eingang den ersten Not-Aus-Tastern zugewiesen. Nachdem die Zugänge zugewiesen werden, lassen sie sich beim Anwenderprogramm einfach durch Anführungszeichen (z. B. Not-Aus-Taster) aufrufen. Jede Eingang ist auf 1oo2 Prinzip eingestellt.

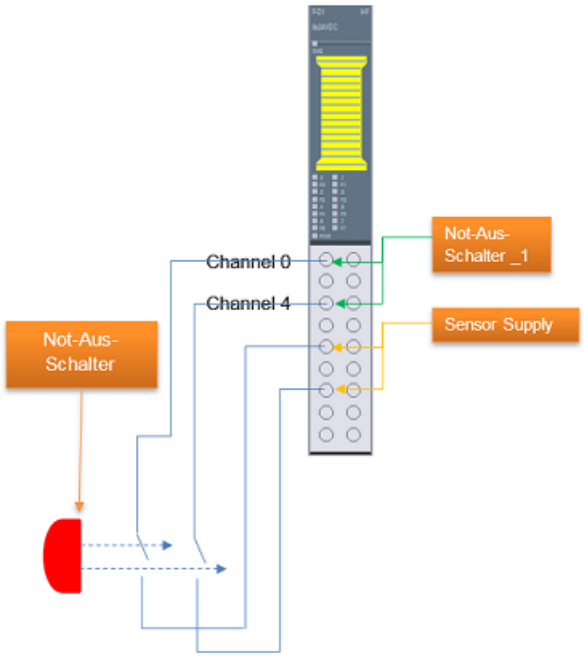


Abbildung 6.10: Verschaltung des Not-Aus-Taster mit F-DI

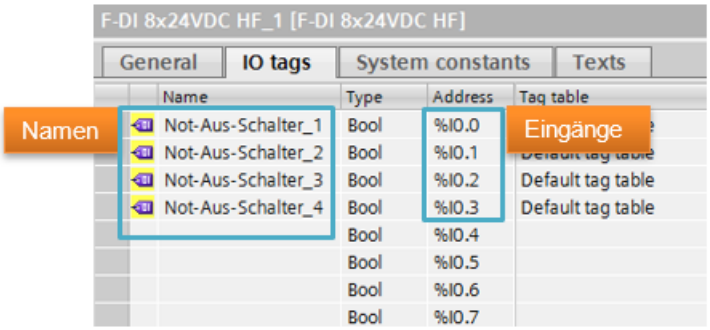


Abbildung 6.11: Zugänge Zuweisung

6.2.2.2 Bausteine von STEP7

Hier werden die Bausteine von STEP 7 beschrieben, auf die das Anwenderprogramm geschrieben wird. Wie wir schon wissen, ist das Simatic-System modular aufgebaut. STEP 7 unterstützt auch diesen Konzept. Das wird durch seine Bausteine erfolgen. Dadurch kann man komplexe Aufgaben in Teilaufgaben aufgliedern. Somit lassen sich die Teilaufgaben Bausteinen zuordnen. Die

Anweisungen werden in Bausteine geschrieben und linear (nach der Reihenfolge) bearbeitet. Der Aufruf der Programmbausteine erfolgt vom übergeordneten Baustein [24]. Es gibt verschiedene Arten von Bausteinen (OB, FC und FB), die in einem Anwenderprogramm verwendet werden können.

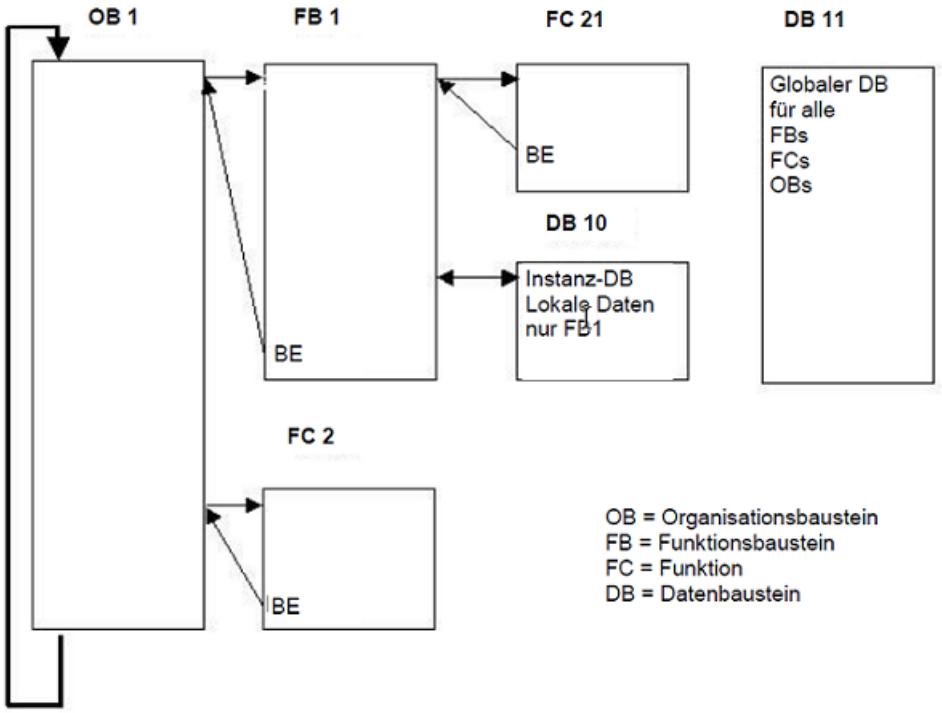


Abbildung 6.12: Bausteine Struktur [45]

- **OB** - Organisationsbaustein wird auch als Grundbaustein (Main [OB1]) genannt. Dieser dient als Schnittstelle zwischen Anwenderprogramm und Betriebssystem der CPU. OB1 wird vom Betriebssystem aufgerufen und zyklisch bearbeitet [24].

- **FC** - Funktion sind "Unterprogramme", die eine Teilaufgabe des Programms übernehmen können. Es kann aber nur lokale Variablen haben, weil es kein Speicherbereich zugeteilt wird [24].

- **FB** - Funktionsbausteine sind wie die FCs "Unterprogramme". Im Gegensatz zu FCs verfügen Funktionsbausteine (FB) über einen Speicherbereich in Form einer Instanz-DB. Das heißt bei jedem Aufruf der FB wird ein Datenbaustein als Instanz zugeordnet. Damit können Variablen auch nach der Bearbeitung der Funktion erhalten bleiben. FBs und FCs stellen ein paar Einträge (Arten von Variablen) zur Verfügung. Der wesentliche Unterschied ist die Static Variable. Das ist nur bei FBs erhalten. Dies dient dazu um die Zwischenwerte im Instanz-Datenbaustein zu speichern. Deswegen bleiben sie auch über mehrere Zyklen erhalten.

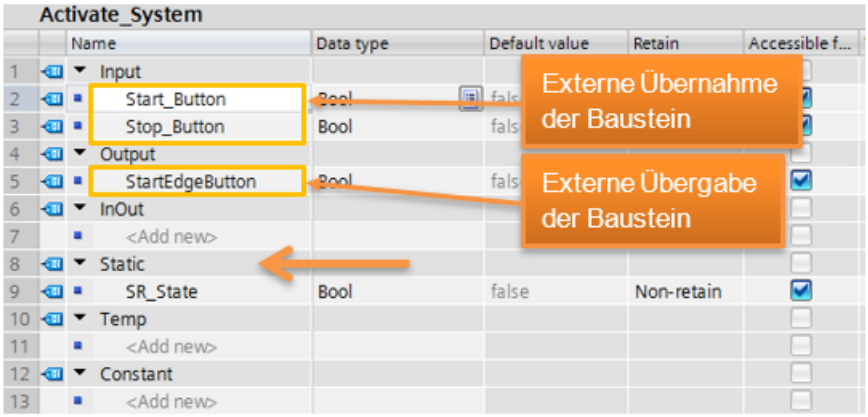


Abbildung 6.13: Vorgegebene Einträge im TIA Portal

- **Datenbausteine [DB]** - während der Bausteine (OB, FC und FB) Codeanweisungen haben können, sind DBs dafür ausgelegt, Datenvariablen zu speichern. Es gibt zwei verschiedene Datenbausteine. Datenbausteine, denen ein Funktionsbaustein [FB] zugewiesen ist, und Globale Datenbausteine, bei denen die Daten von allen Bausteinen gelesen werden können. Der Unterschied ist in Abbildung 6.14 ersichtlich.

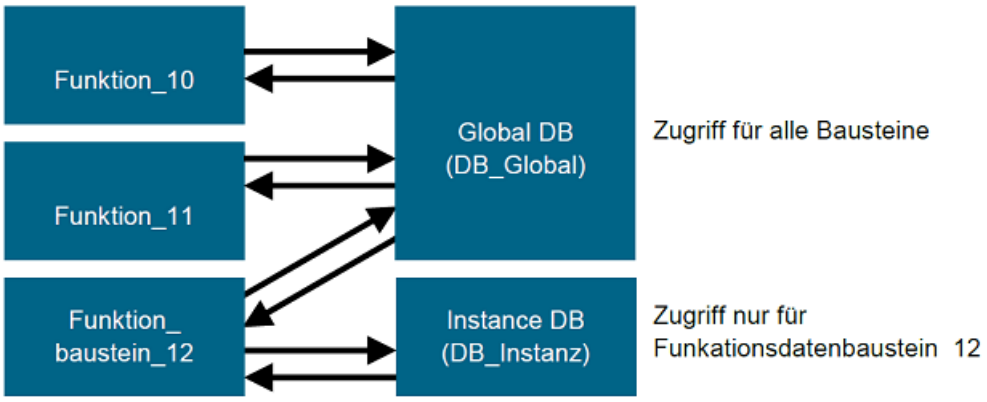


Abbildung 6.14: Globale- und Instanz-Datenbausteine [46].

6.2.2.3 Anwenderprogramm

Das Anwenderprogramm implementiert die Funktionalität des Simatic-Systems. Die Hauptfunktionen des Anwenderprogrammes sind die Überwachung von Not-Aus-Tastern, die Überprüfung der Objektkoordinaten (über OPC UA Server) und die Visualisierung der VSA-Zustände über die Signalleuchten (LED-Türme). STEP 7 ermöglicht uns komplexe Aufgaben durch die Bausteine in Teilaufgaben zu unterteilen. Deshalb ist das Programm in mehrere Programmbausteine unterteilt. Die Unterteilung erfolgt, so dass die Unterprogramme in die Bausteine (OB1, FC, FB)

geschrieben werden. Die Erstellung der neuen Bausteine erfolgt im TIA Portal unter "Project tree "=> "Program block ".Dort werden durch Doppelklick auf "Add new network " alle mögliche Bausteine zur Verfügung gestellt, die STEP7 anbietet. DDie Aufteilung der Bausteine im Programm ist in Abbildung 6.15 ersichtlich. Erste Baustein ist Organisationsbaustein Main [OB1], der durch die Farbe Violett zu erkennen ist. Als Nächstes kommen die FC- Blöcke, die grün gefärbt sind. FBs stehen in der Mitte und sind durch hellblaue Blöcke zu erkennen. DBs haben eine andere Blockform, die wie eine übliche Datenbankform aussieht und Farbe Dunkelblau hat. Am Ende stehen Safety-Bausteine, die für das Sicherheitsprogramm zuständig und durch die gelbe Farbe identifizierbar sind.

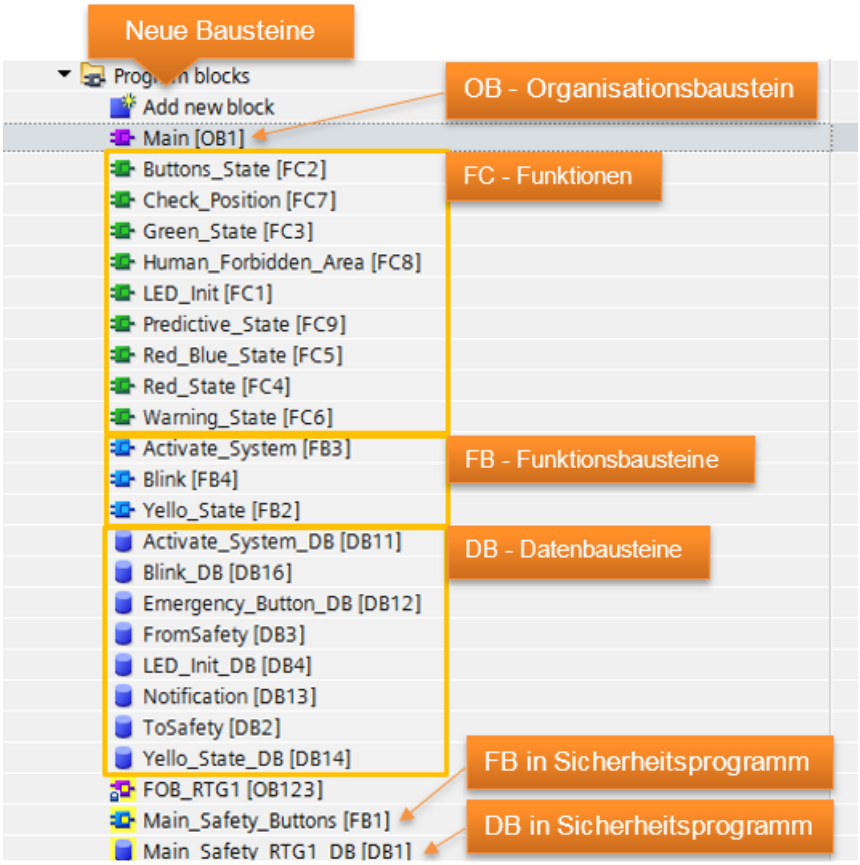


Abbildung 6.15: Projekt Baum der Bausteine im TIA Portal

Hinweis : *Die Programmbausteine werden nicht in der Reihenfolge aufgerufen, wie unter Abbildung 6.15 ersichtlich sind. Eine genaue Programmablauf wird später erklärt.*

1. **Main[OB1]** - der Organisationsbaustein ist der Grundbaustein und dient als Schnittstelle zwischen Anwenderprogramm und Betriebssystem. OB wird von CPU zyklisch aufgerufen. Der Programmablauf wird zyklisch abgearbeitet.. Im OB1 werden die FCs bzw. FBs aufgerufen, die in diesem Anwenderprogramm wirken.

```
"Activate_System_DB"(Start_Button:="Start",
                     Stop_Button:="Stop",
                     StartEdgeButton=>"ToSafety".StartSignal);
"Buttons_State"();
"LED_Init"();
"Green_State"();
"Red_State"();
"Red_Blue_State"();
"Human_Forbidden_Area"();
"Yello_State_DB"();

// Zusatzliche- bzw. Zukuntige-Optionen
//"Predictive_State"();
```

Abbildung 6.16: Die aufrufenden Bausteine im OB1

2. **FB Funktionsbaustein**

- **Activate_System_DB(...)** - ist ein Funktionsbaustein und ist in FUP, einer graphischen Programmiersprache, geschrieben. Durch diesen FB-Funktionsbaustein wird der Demonstrator aktiviert/gestartet bzw. deaktiviert/abgeschaltet. Die Aktivierung erfolgt durch eine positive Flanke am Eingang über den Start-Taster. Das Startsignal wird zurückgestellt, wenn der Stop-Taster betätigt wird. Somit wird die Anlage (Demonstrator) abgeschaltet. Beide Schalter sind als externer Eingang auf FB zugewiesen. Um nach Betätigung der Taster die Zustände zu speichern, wird ein SR Flip-Flop verwendet und eine Static-Variable "SR_State " zugewiesen, damit der Zustand über mehrere Zyklen erhalten bleibt.

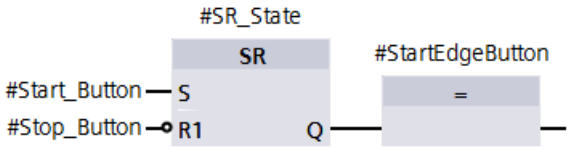


Abbildung 6.17: Set/Reset Flip-Flop

Außerdem wird durch diese FB in den globalen Datenbaustein "ToSafety "auf Varibale "StartSignal "geschrieben. "StartSignal"wird in Safety (sicherheit) Programm "Main_Safety_Buttons "ausgewertet. Dort wird das Safety-Programm für die Auswertung von Not-Aus-Tastern ablaufen. Wie unter Abbildung 6.2 ersichtlich, wird die Überwachung der Not-Aus-Taster-Signale in die ESTOP1-Anweisung implementiert [48]. Die ESTOP1-Anweisung wird von STEP 7 Safety Advanced zur Verfügung gestellt. Wenn der Notausschalter betätigt ist, zeigt die Anweisung auf Q-Ausgang

FALSE. In diesem Fall, wenn der Not-Aus-Taster_1 betätigt und die Anlage (Demonstrator) aktiviert wird, dann sollte auf dem Q-Ausgang FALSE stehen. Ansonsten wird immer TRUE. Ausgang Q in einen globalen Datenbaustein "FromSafety"geschrieben. Das gleiche Auswertungsprinzip wird für alle Not-Aus-Taster ausgelegt.

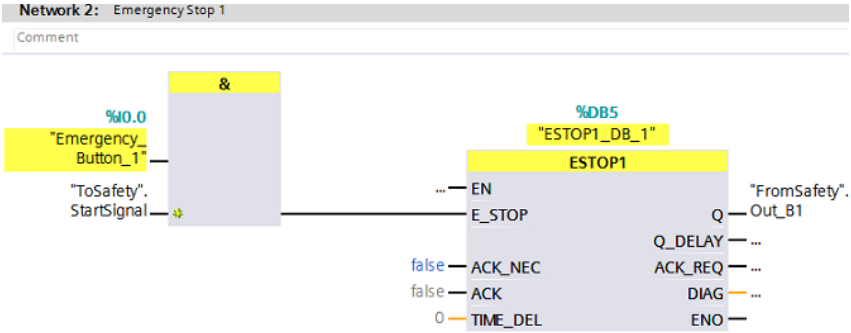


Abbildung 6.18: ESTOP Anweisung.

- **Yellow_State()** - ist ein spezielle Fall, weil es sich um mehr als zwei VSA handelt. Dieser Funktionsbaustein kommt in Frage, wenn es sich zwei oder alle drei VSAs in Safe State (in sicheren Zustand) befinden. Das ist dafür gedacht, wenn in mindestens zwei VSAs eine gefährliche Situation eintritt und sie in sicherem Zustand sind oder wenn während Kameraverifikation ein Fehler auftretet, dann sollte der Arbeiter bzw. Roboter, die gerade im dritten VSA (im grünen Zustand) weiterarbeiten, eine Meldung bekommen, dass die zwei restlichen VSAs sich in Safe State befinden. Während der anderen Meldungen werden die LED-Türme leuchten, hier wird der graue LED blinken. Dafür werden zwei so bezeichnete "Timer operations " Anweisungen("TON " und nächste "TOF ") verwendet die in der Bastein "Blink " programmiert sind. Hier ist das Teilprogramm in FUP programmiert. Die erste Operation "TON "eine Einschaltverzögerung erzeugen. Das heißt, dass wir die Setzung des Ausganges verzögern (ref. aus TIA Portal Hilfe) können, nach der Zeitdauer PT. Die nächste Anweisung ist "TOF " und bedeutet, dass eine Ausschalte-Verzögerung erzeugt wird. Genau gesagt, wird die Rücksetzung des Ausgangs um eine vorgegebene Zeitdauer PT verzögert.

3. **FC Funktionen** -

- **Buttons_State()** - durch diese Funktion (FC) erfolgt die Zuweisung der Not-Aus-Taster-Zustände auf einen zweidimensionalen Array, der in einem globalen Datenbaustein deklariert ist. Zweidimensionale Arrays wurden absichtlich verwendet, damit die Verteilung der Not-Aus-Taster in den jeweiligen VSAs verdeutlicht wird. Einfacher gesagt löst das die Frage: Welche Not-Aus-Taster gehören zu welchen VSAs (Virtuell Safety Area). Wie bereits ausgeführt, werden Not-Aus-Taster in Safety Program

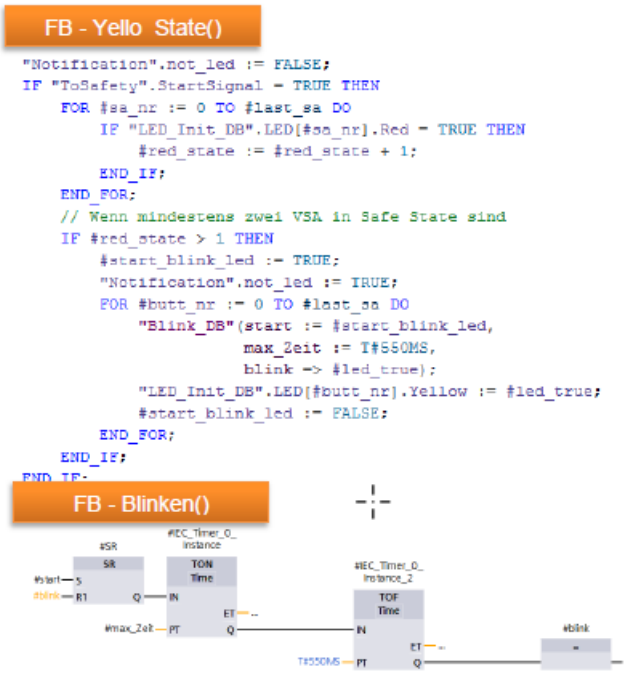


Abbildung 6.19: Warnungszustand durch die gelbe Farbe

(Main_Safety_Buttons) ausgewertet und der Zugriff erfolgt über "FromSatey ". Für ein besseres Verständnis der Zuweisung siehe Abbildung 6.20

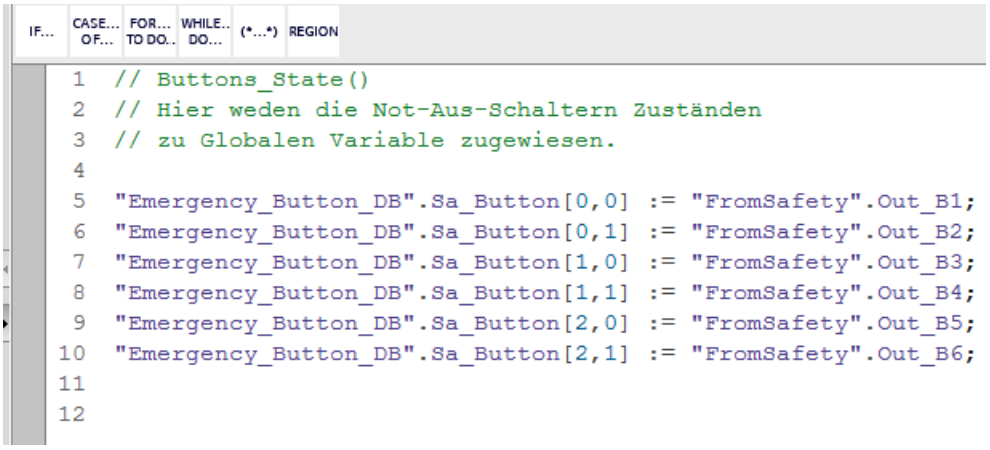


Abbildung 6.20: Not-Aus-Taster-Zuweisung zu globalem Datenbaustein Array

- **LED_Init()** - die Zuweisung und Initialisierung der LED Türmen passiert in diesem Funktion (FC). Nach Bekanntgabe (durch Doppelklich auf Modul und unter IO tags) der auf DQ (Digitalausgabemodule)-Ausgänge, werden LED-Zustände zugewiesen. Die zugewiesene Zustände sind in einen globalen Datenbaustein gespeichert "LED_Init_DB ". Dort werden so genannte anwenderdefinierten Datentypen "SA_LED

"definiert. Diesen Datentypen entsprechen eine Struktur, die in diesem Projekt für die LED-Farben verwendet wird. Das erleichtert die Arbeit bei der Wiederverwendung gleicher Farben in mehreren VSAs.

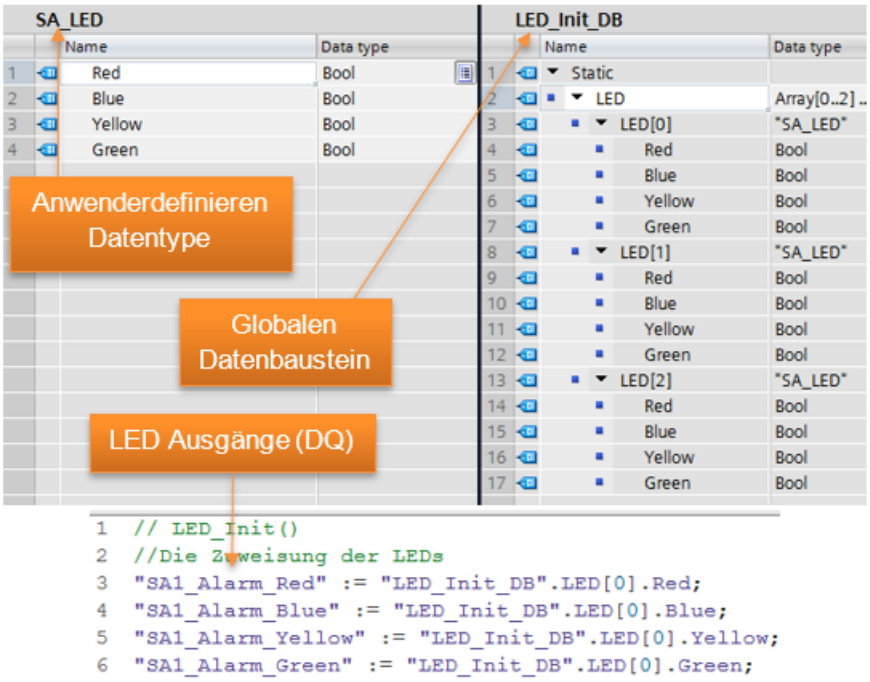


Abbildung 6.21: Die Zuweisung bzw. Initialisierung der LED-Türme

- **Green_State()** - im grünen Zustand befindet sich ein VSA, wenn keiner von beiden Not- Aus-Tastern dieses VSA gedrückt ist. Durch dieses FC erfolgt die Überprüfung aller VSAs mit den jeweiligen Not-Aus-Tastern. . Die erste Bedingung für die Ausführung dieser Funktion ist, dass die Anlage aktiviert sein muss. Danach wird jeder LED-Turm der drei VSAs grün leuchten, wenn keiner der beiden Not-Aus-Taster gedrückt ist. Wenn der LED-Turm grün leuchtet, sind die Maschinen (z. B. Roboter) in diesem Bereich im normalen Arbeitsmodus. Das heißt, in diesem Arbeitsbereich (Virtual Safety Area) besteht keine Gefahr. Hierbei passiert keine Objektkoordinaten-Überprüfung, weil es nur der (grüne) Anfangszustand ist.
- **Red_State()** - in diesem Baustein (Abbildung 6.22) ist der rote Zustand einer oder mehrere VSAs (Virtual Safety Area) programmiert. Dieser Zustand tritt ein, wenn einer von den beiden Not-Aus-Tastern derselben Virtual Safety Area gedrückt wird und sich gerade eine Maschine (z. B. Roboter) dort befindet. Wenn der LED-Turm rot leuchtet, sind die betroffenen Maschinen in diesem Virtual Safety Area im sicheren Zustand. Der Arbeitsfluss ist in diesem Bereich unterbrochen. Dieser Baustein wird gleichfalls wie beim vorherigen Baustein am Anfang geprüft, wenn die Anlage aktiviert ist. Danach werden die ausgewertete Not-Aus-Tastern im Safety Program

"Main_Safety_Buttons"über globalen Datenbaustein "Emergency_Button_DB "geholt und überprüft.. Wenn ein Not-Aus-Taster gedrückt ist, wird ein anderer Baustein "Check_Position() "aufgerufen.Durch diesen FC-Baustein werden die Objektkoordinaten von Robotern und Menschen geprüft. Die Koordinaten werden von der Kamera erkannt und über OPC UA Server an S7 1500 CPU gesendet (5.2.1 OPC UA). Unter PLC[3] tags sind Integer-Variable deklariert, die die Werte von Objektkoordinaten erhalten. Die Werte werden über OPC UA Server übertragen und aktualisiert, um ständig die Positionen der Objekte zu bekommen. Check_Position() teilt die Arbeitsbereiche virtuell auf und deswegen heißen sie auch Virtual Safety Area. Für die Aufteilung wird eine Switch Case Anweisung [4] verwendet. Nach dem Schlüsselwort "CASE "wird der Ausdruck "Sa_Nr "geschrieben, welche wir auswerten wollen. Dieser Ausdruck ist als Input (Eingabe) der Baustein und weist aus, in welcher VSA ist ein Not-Aus-Taster gedrückt ist. Danach folgen die Fälle (1, 2 oder 3), die der VSA entsprechen. Bei Übereinstimmung wird dort geschaut (durch die Koordinaten-Auswertung), ob es sich gerade ein Roboter oder ein Mensch in entsprechende VSA befindet. Wenn es der Fall wäre, dann ergibt als Wahr (TRUE) auf Output (Ausgabe) dieser Baustein. Nach der Bearbeitung der Check_Position() kehrt das Programm wieder in Red_State() zurück und abarbeitet weiter die Folgende Anweisungen. Wie wir schon wiessen, bedeutet Rot sichere Zustand (en. Safe State). In ein Safe-State kann die Anlage gebracht werden, wenn ein Fehler beim Bilderkennung passiert und erkannt wird. Dafür gibt es eine Variable "Safe_State "benannt ist. Sie kommt auch vom Kamerasystem über OPC UA und wenn ein Fehler erkannt wird, dann werden dadurch alle VSAs in den Safe State gebracht. Das ist auch in Abbildung 6.22 zu sehen.

- **Red_Blue_State()** - ist das gleiche Baustein wie Red_State(). Der einzige Unterschied ist, dass hier noch ein Blue LED leuchtet. Das passiert, wenn bei beiden Not-Aus-Tastern der gleiche VSA gedrückt wird.
- **Humman_Forbidden_Area()** - durch diesen FC-Baustein wird den Arbeitern die erste VSA verboten. Sobald ein Arbeiter die erste VSA betritt, wird diese VSA in Safe State gehen. In Zukunft ist gedacht, Audio-Meldungen zu installieren, sodass es bei Versehen auch eine Audio-Warnung gibt.
- **Predictive_State()** - wurde als eine weitere Funktionalität dieses Demonstrators gebaut. Die Idee dahinter ist, dass wenn sich eine VSA in einem Safe-State befindet und in die Richtung diese VSA die Bewegungstrajektorien (Bewegungsverlauf) eines Manschens bzw. Roboters zeigt (Mensch oder Roboter bewegen in Richtung Safe-State VSA), dann wird ein Warnsignal gegeben, wenn die Anzahl der vordefinierten Schritte überschritten wird. Wie man weiß, wird die Position von einem

[3]SPS engl. PLC (Programmable Logic Controller)
[4]Swich Case Anweisung ist eine bedingte Anweisung

```
IF "ToSafety".StartSignal = TRUE THEN

    FOR #sa_nr := 0 TO 2 DO
        FOR #butt_nr := 0 TO 1 DO
            //If one of both Buttons is pressed, than send Red signal
            IF "Emergency_Button_DB".Sa_Button[#sa_nr, #butt_nr] = FALSE THEN
                // Here is checked if on of Robots in this SA currently located
                #sa_nr_pos := #sa_nr + 1;
                //Check the Positions of Robots
                "Check_Position"(Sa_Nr:=#sa_nr_pos,Robot_Loca=>#Rob_located,
                                 Human_Loca=>#Hum_located);
                //If the Robot in this SA currently located
                IF #Rob_located = TRUE THEN
                   "LED_Init_DB".LED[#sa_nr].Red := TRUE;
                   "LED_Init_DB".LED[#sa_nr].Green := FALSE;
               ELSE
                   "LED_Init_DB".LED[#sa_nr].Green := TRUE;
               END_IF;
            ELSE
                #sa_nr_pos := 0;
            END_IF;
            "Notification".Red_Area := #Rob_located;
            //Failure in image Processing - Camera error
            IF "Safe_State" = TRUE THEN
                "LED_Init_DB".LED[#sa_nr].Red := TRUE;
                "LED_Init_DB".LED[#sa_nr].Green := FALSE;
            ELSE
                "Safe_State" := FALSE;
            END_IF;
        END_FOR;
    END_FOR;
END_IF;
```

Abbildung 6.22: Sicherer (roter) Zustand einer VSA - FC Red_State()

Objekt durch die Koordinaten (X,Y) dargestellt. Dadurch wird auch die Anzahl der Schritte berechnet. Nach fünf Schritten in die Richtung der Safe-State VSA wird ein Warnsignal (Prävention-Meldung) geben. Das ist eine Prädiktiv-Sicherheitsmeldung bzw. Prävention von sicherheitskritischen Bewegungen.

- **Warning_State()** - ist nicht relevant für dieses Programm.

4. **DB Datenbasteine** - eine weitere und detaillierte Erklärung der Datenbausteine ist nicht nötig. Wie erwähnt, sind DBs dafür ausgelegt Daten (Variable) zu speichern. Es gibt zwei Arten der DB, Globale Datenbausteine und Datenbausteine, die einer FB zugewiesen sind.

6.3 Übertragungsfunktion

Ein wesentlicher Punkt von Industrie 4.0 ist die Vernetzung zwischen Geräten unterschiedlicher Hersteller. Das ermöglicht einen Datenaustausch zwischen einzelnen Automatisierungsgeräten. Dieses Kommunikationskonzept wurde auch in dieser Arbeit implementiert und "Übertragungsfunktion" benannt. Die Übertragungsfunktion überträgt die Daten vom Kamerasystem nach Simatic CPU über OPC UA Server. OPC UA (Open Platform Communications Unified Archi-

tecture).Das ist ein Kommunikationsstandard von Industrie 4.0. und verfügt über ein System, das als Server arbeitet und die vorhandenen Informationen anderen Systemen (Clients) zur Verfügung stellt. Ein System kann zugleich Server oder Client sein [7].]. Der Client kann lesend oder schreibend auf die Daten, die im Server vorhanden sind, zugreifen. Neben Variablen ist auch ein Methodenaufruf möglich.

6.3.1 OPC UA konfiguration

S7 1500 CPU (ab. FW 2.0) verfügt über OPC UA Server und ermöglicht einen Datenaustausch mit RPI. Für diese Kommunikation sind einige Konfigurationen unter TIA Portal erforderlich sowie die Programmierung der Client-Seite auf RPI (Python Source). Am Anfang muss der OPC UA Server aktiviert werden. Bevor der Server aktiviert wird, wird eine Lizenz "Runtime-Licenses " vorausgesetzt.Diese Lizenz ist nötig für den Betrieb der Server. Je nach CPU-Leistung gibt es drei Arten von Lizenzen:

- SIMATIC OPC UA S7-1500-small
- SIMATIC OPC UA S7-1500-medium
- SIMATIC OPC UA S7-1500-large.

Für unserem CPU wird "medium" gefordert (Abbildung 6.23 Punkt 2). Durch Doppelklick auf CPU kommt man zu Eigenschaften "General ". Unter **OPC UA / Server / Accessibility of the Server** wird der Server aktiviert und die Bestätigung der Sicherheitshinweise folgt [7]. Danach wird die geforderte Lizenz unter "Runtime licenses " gewählt. Nach Bestätigung der Sicherheitshinweise wird ein Zertifikat automatisch erzeugt und im lokalen Zertifikat-Verzeichnis der CPU gespeichert. Damit kann eine sichere Verbindung zum Client erstellt werden. Im aktuellen Zustand des Projektes wird keine Verbindung mit dem lokalen Zertifikat erfolgen.

6.3.1.1 OPC UA Server-Adresse

Die PROFINET-Ports der S7 1500 CPU ermöglichen den Zugang zum OPC UA Server. Das PROFINET-Kabel wird direkt an die CPU PROFINET-Schnittstellen gesteckt und über Switch an RPI angeschlossen. Außerdem gibt es für den Verbindungsaufbau noch die Server-Adressen (Abbildung 6.23 Punkt 3) der CPU. Dadurch ist der Server von Client erreichbar. Die Adresse ist in drei Teile gegliedert. Erste Teil ist *Protokollkennung* **"opc.tcp://** **"**. Der nächste ist *IP-Adresse* **"192.168.0.1"**, um den Server über Ethernet-Subnet zu erreichen. Und der letzte Teil ist *TCP-Port* **"4840"** (ist ein Standard Port, kann aber geändert werden) [7].

Client wird schreibend auf die Daten, die im Server vorhanden sind, zugreifen. Das sind die Koordinaten von Objekten.

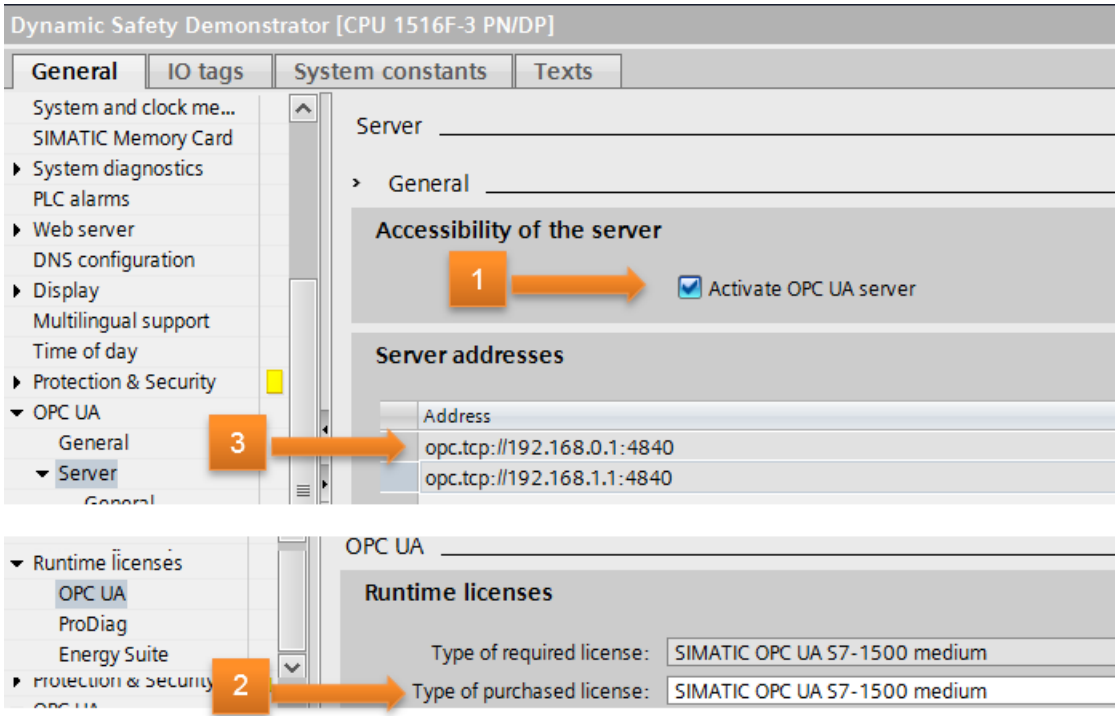


Abbildung 6.23: OPC UA Server Konfiguration.

6.3.1.2 OPC UA Client

DDie Client-Seite erfolgt in RPI und wird in der gleichen Source-Datei mit einem Bildverarbeitungs-Algorithmus programmiert. Das heißt, die Client-Seite wird in Python programmiert. Dafür sind am Anfang einige Installationen vorzunehmen, z. B. python-opcua Biblothek [40]. Danach sind die notwendigen Client-Pakete importiert.

```
from opcua import Client
from opcua import ua
```

Nach dem die wichtigsten Paketen importiert sind, kann eine Verbindung zur Server erstellt werden. Die Server-Adresse (Kapitel 6.3.1.1) ist für die Erkennung und Adressierung des Servers. Wie auch unten in der Anweisung zu sehen ist, erhält Client-Anweisung die Server-Adresse.

```
Client = Client("opc.tcp://192.168.0.1:4840")
```

Die Erstellung der Verbindung wird versucht durch folgenden Anweisungen:

```
try:
    print("Tryint to connect to OPC Server")
    client.connect()
    time.sleep(1)
```

```
        print("Device connected to OPC Server")
        connected = True;
    except:
        print("Unexpected error while conncting to OPC Server")
        connected = False;
```

Wenn die Kommunikation zum Server erfolgreich erstellt worden ist, ist es möglich auf die Server-Daten lesend oder schreibend zuzugreifen. Wie bekannt ist, stellt der Server die Daten in Form von Knoten (eng. Nodes) zu Verfügung. Die Beziehung zwischen den Knoten ist hierarchisch gestaltet. Die Knoten können ein Objekt, eine Methode oder eine Variable beinhalten. Das Netzwerk von Knoten ist so angeordnet, dass ausgehend von den Wurzelknoten (eng. Rootnode) alle gewünschten Knoten gangbar sind. Client stellt die Methode

```
root = get_root_node()
```

zur Verfügung. Dadurch können wir, ausgehend von der Wurzel, auf die Knoten zugreifen. In unserem Fall geht es um die Variablen, auf die wir die Positionskoordinaten schreiben werden. Nachdem die Positionskoordinaten ermittelt wurden, müssen wir die Variablen lesen und schreiben. Die Beschreibung, wie die Positionskoordinaten ermittelt werden, ist im 6.4.4 zu lesen. Um die Variable lesen zu können, müssen wir ein Objekt der Variable generieren. Das ist durch die Methode "getChild() " möglich.

```
// Objekt der Variable generiert - Roboter 1
opc_r1x = root.get_child(["0:Objects","3:SA_PLC_Button","3:Memory","3:R1_x"])
opc_r1y = root.get_child(["0:Objects","3:SA_PLC_Button","3:Memory","3:R1_y"])
```

Das Schreiben auf entsprechende Variable ist auch ganz einfach. Dafür wird eine Setter-Method verwendet.

```
// Auf OPC Server Variablen schreiben - Roboter 1
opc_r1x.set_data_value(int(xa1), uaVariantType.Int16)
opc_r1y.set_data_value(int(ya1), uaVariantType.Int16)
```

Für alle Objektkoordinaten wird das gleiche Verfahren in Python implementiert.

6.4 Kamerasystem Aufbau

In diesem Teil der Implementierung wird die Bilderkennung und die Verarbeitung beschrieben. Dafür wird eine Kamera und Raspberry PI verwendet. Die beiden sind keine sicheren Hardwarkomponente [5], was in unserem Fall sehr wichtig wäre. Um ein sicherheits Funktion zu schaffen,

[5] Sind nicht für die Funktionale Sicherheit nach eine bestimmte Sicherheitsnorm zertifiziert.

muss die gesamte sicherheits Kette eines Schutzsystems sicher sein. Der Grund warum es keine Safety Kamera verwendet wird, ist, weil es noch nicht im Markt solche geeignete Safety Kamera gibt. Die Kameras, die eine höheres Sicherheits-Level haben, waren zu teuer für dieses Projekt.

6.4.1 Raspberry Pi Camera v2

Die Bilderkennung erfolgt über eine Kamera und die Bearbeitung in Raspberry PI. Für diesen Aufbau wird eine Kamera eingesetzt, die kombinierbar mit RPI (offizielle RPI-Kamera) ist. Raspberry Pi Camera v2 ist mit allen RPI-Modellen kompatibel. Diese Kamera verfügt über 8 Megapixel Auflösung, hochwertige Bilder und einen Bildsensor von Sony (IMX219PQ). Durch diese Kamera werden die Bewegungen verfolgt und die Positionen von Objekten erkannt. Zur Bildanalyse wird die OpenCV Biblothek verwendet, welche im folgenden Abschnitt erklärt wird. Der Anschluss der Kamera an RPI erfolgt über ein Flachbandkabel und wird direkt an CSI (Camera Connector sieh Abbildung 5.9) serielle Schnittstelle von RPI gesteckt. Dies wird oberhalb von Basisfeld positioniert und befestigt (Abbildung 6.24). Es wurde eine Metallstange mit Winkeln für die Positionierung montiert. Für die Befestigung der Kamera wurde ein Halterungsgehäuse mit Hilfe vom 3D-Drucker im Siemens-Labor Graz hergestellt.

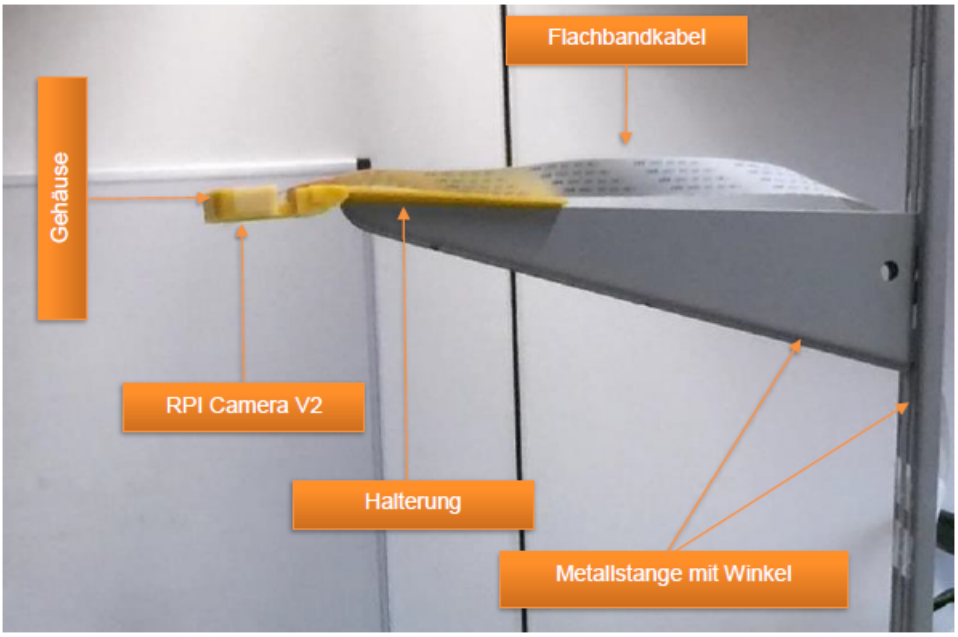


Abbildung 6.24: Raspberry Pi Camera v2

6.4.2 Raspberry PI 3

Für die Bildbearbeitung und die Übertragung der Objektkoordinaten auf das Simatic-System wurde ein Raspberry Pi Version 3 verwendet. Die Wahl fiel auf das System, weil es preisgünstig

und kompatibel zu RPi und Übertragungsfunktion ist

6.4.2.1 Betriebsystem auf RPI installieren

Um mit Raspberry Pi zu beginnen, benötigt man ein Betriebssystem. NOOBS (New Out of Box Software) ist ein einfacher Installationsmanager für das Raspberry Pi. Zunächst muss das Betriebssystem NOOBS von der Raspberry PI-Website heruntergeladen werden. Danach muss es auf SD-Karte extrahiert werden. Von Raspberry Pi wird 8 GB empfohlen, aber aus Erfahrung ist es besser mindestens 16 GB zu verwenden. Bevor das Betriebssystem extrahiert wird, ist es besser, die SD-Karte zu formatieren. Wenn das Extrahierungsverfahren fertig ist, wird die SD-Karte sicher entfernt und in die Raspberry Pi gesteckt. Danach werden Maus, Tastatur und Monitorkabel (HDMI-Kabel) sowie USB-Stromkabel an RPI angeschlossen. Raspberry Pi wird gestartet, und ein Fenster mit einer Liste verschiedener Betriebssysteme wird angezeigt, die man installieren kann. Es wird die Verwendung von Raspbian empfohlen. Wenn der Installationsprozess erfolgreich abgeschlossen ist, können wichtige Konfigurationen (z. B. Zeit zum Aktivieren der Kamera usw.) vorgenommen werden.

6.4.2.2 OpenCV Installieren

Wie bereits erläutert wurde, wird die OpenCV-Bibliothek benötigt, um das Bild verarbeiten zu können. Die Installation von OpenCV auf einem Raspberry Pi kann sehr zeitaufwändig sein, da viele Abhängigkeiten und Voraussetzungen installiert werden müssen. Das gesamte Verfahren findet man im Internet [31].

6.4.3 Aufbau

Hier wird der Aufbau von RPI gezeigt. Raspberry PI ist an der Metallstange festgemacht und steht zwischen Kamera und Simatic-System. RPI benötigt 5 V (Spannungsversorgung) und 1 A (Strom) wird empfohlen, aber in dieser Arbeit werden wir HDMI-Port verwenden, die ca. 500 mA braucht und mit anderen USB-Ports kommen wir zusammen auf über 1000 mA. Dafür wird ein Ladegerät mit 2 mA verwendet. Für die Verbindung zum Simatic-System wird ein Ethernet-Kabel verwendet. Das wird nicht direkt an CPU (S7 1500) gesteckt, sondern über einen Switch. Wie unter Abbildung 6.25 zu sehen ist, sind die meisten Teile schon bekannt.

6.4.3.1 RGB LEDs

Da keine sichere (safety) Kamera verwendet wird, können Fehler auftreten, die zu Fehlfunktionen in der Kamera oder bei der Bildklassifizierung führen. Das würde die funktionale Sicherheit (eng. Functional Safety) des Systems verletzen. Bei der funktionalen Sicherheit muss jedes Teilsystem sicher sein. Denn wenn ein Teilsystem ausfallen würde, führt das zu einem Ausfall der gesamten Sicherheitsfunktion (DIN EN 62061). Damit die Zuverlässigkeit an die Kamera erhöht wird,

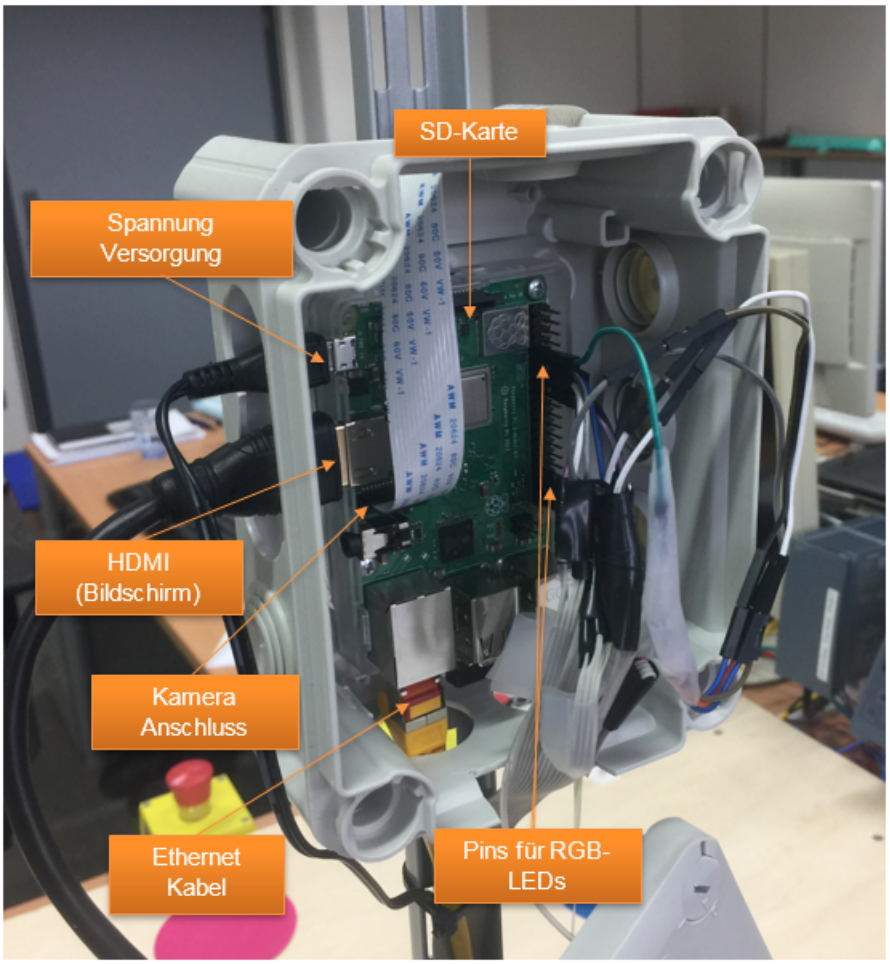


Abbildung 6.25: Raspberry Pi Aufbau

wurde eine Idee erfunden. Die Idee ist, die Kamera mittels RGB-LEDs zu testen. Ziel dessen ist, die Kamera vor der Zustandsbewertung der VSAs zu testen. Durch diesen Test können wir beim jeden Zyklus-beginn schon wissen, ob die Kamera funktioniert. Das erfolgt durch die zufällige Umschaltung der RGB-LEDs, so dass die RGB-LEDs zufällig umgeschaltet werden und die Kamera wird gefragt, wenn sie auch das gleiche sieht. BBei Übereinstimmung werden Bildaufnahme und Bildverarbeitung fortgeführt. Wenn das aber nicht der Fall ist, dann geht das gesamte Sicherheitssystem in Safe State. Dieses Verfahren nennen wir Kameraverifikation und es ist im Quelltext "LED_Check.py " implementiert. Damit die RGB-LEDs ersichtlich für die Kamera sind, wurden die RGB-LEDs direkt auf dem Basisfeld eingebaut. Das ist in der Abbildung 6.26 zu sehen.

6.4.4 Bildverarbeitung mit Python

In diesem Teil wird der Bildverarbeitung-Algorithmus (Haupt-Algorithmus) beschrieben. Das erfolgt auf RPI und wird in Python unter "ImageProccessing.py"-Datei programmiert. Die Ka-

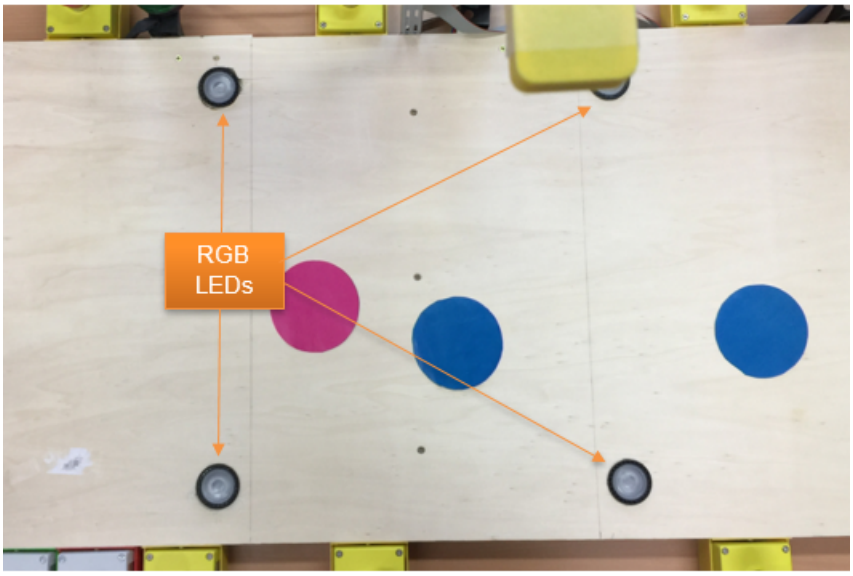


Abbildung 6.26: RGB-LEDs Aufbau.

mera ist die Eingabe (Input) des Algorithmus und stellt die Digitalbilder von der industriellen Umgebung zur Verfügung. Die Bilder werden bearbeitet und als Ergebnis sind die aktuellen Positionskoordinaten von Objekten. Am Anfang werden die notwendigen Pakete importiert. Dann werden die Treiber

```
import os
import time
os.system("sudo modprobe bcm2835-v4l2")

// Import the necessary Packages
from umutils.video import VideoStream
import numpy as np
import cv2
import imutils
import sys
```

für die Kamera geladen. Diese Treiber müssen geladen werden, bevor die Kamera benutzt wird. Für die Objekterkennung wird der HSV-Farbraum verwendet. Der Standard-Farbraum von OpenCV ist der so genannte RGB (Rot, Grün und Blau), aber hier werden bestimmte Farben gefiltert und dafür ist HSV geeigneter [32]. Diese Farbraum wird durch drei Koordinaten definiert: **H**ue (Farbwert), **S**aturation (Sättigung) und **V**alue (den Hellwert) [5]. Die Farbfilterung des Bildes erfolgt durch die Farbgrenzen. Als Nächstes folgt die Definierung der untere und obere Grenze für die "pink" und "blaue" Farbe von Objekten.

```
// Lower and Upper boundaries of the "Pink" and "Blue"
// Boundaries = (H, S, V)
blueLower = (169, 100, 100)
blueUpper = (189, 255, 255)
pinkLower = (36, 40, 40)
pinkUpper = (86, 255, 255)
```

Die Farben werden im HSV-System [34] definiert. Wie wir schon wissen, wird die Rolle der Ersatz-Objekte durch seine Farbe definiert. In unserem Fall ist das blaue Objekt ein Stellvertreter der Roboter und das pinke Objekt ersetzt den Arbeiter. Die geometrische Figur eines Objektes ist ein Kreis mit einem konstanten Radius für alle Objekte. Diese werden durch ihre Eigenschaften erkannt (Abbildung 6.27).

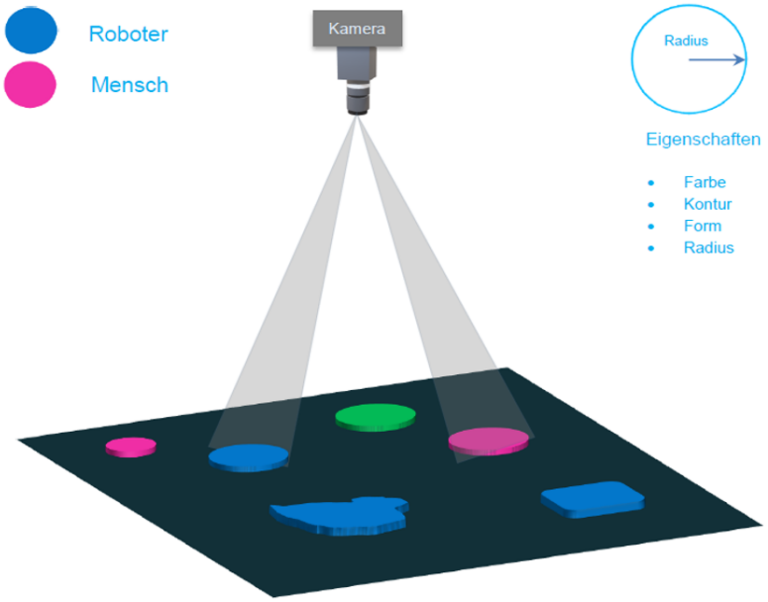


Abbildung 6.27: Objekterkennungsprinzip.

Die Definition der Farbgrenzen hängt von den Lichtstärken ab. Das bedeutet, wenn sich die Lichtverhältnisse ändern, wird der Algorithmus die Farben nicht mehr richtig erkennen. Das kommt häufig vor, wenn der Demonstrator in anderen Umgebungen (z. B. andere Räume) gebraucht wird, wo die Lichtstärke anders ist. Damit er überall (in jeder Umgebung) an die Lichtverhältnisse angepasst werden kann, wurde ein zusätzlicher Algorithmus programmiert. Dieser Algorithmus kalibriert HSV-Color und ist extra zum Ausführen und wird dafür verwendet, bei unterschiedlichen Lichtverhältnissen die geeigneten unteren bzw. oberen Farbgrenzen zu finden. Die gefundenen Farbgrenzen werden in den Haupt-Algorithmus eingetragen. Somit wird die Farberkennung der Objekte ohne Probleme funktionieren. Dieses Verfahren wird "Farbkalibrierung " genannt und später ausführlich erklärt. Es gibt noch einen anderen wichtigen Algorithmus

für die Kalibrierung der Positionen von RGB-LEDs. Der wird auch separat ausgeführt, um die Positionen von RGB-LEDs zu ermitteln.

6.4.4.1 Bildfilterung

Hier werden alle Schritte bzw. Funktionen beschrieben, wodurch das Bild (das Frame) gefiltert wird. Im nächsten Schritt wird durch

```
vs = VideoStream(src=0).start()
```

Statement ein Videostream gestartet und "**vs**" zugewiesen. Dann kommt die Verbindung zu OPC UA Server, das ist in Abschnitt 6.3.1.2 beschrieben. Das Frame (Bild/Rahmen) wird über die Funktion/Methode

```
// read()
frame = vs.read()
```

gelesen. Für eine bessere Filterung wird, die Größe des Bildes (Frame) geändert. Eine solche Änderung des Bildes erfolgt durch

```
// resize()
frame = imutils.resize(frame, width = 400)
```

Funktion. Für die Glättung oder das Weichzeichnen wird der Gauß-Filter benutzt. Der Verwendungszweck ist das Bildrauschen zu vermindern:

```
// GaussianBlur()
blurred = cv2.GaussianBlur(frame, (11, 11), 0)
```

Danach kommt die Konvertierung des Frames in den HSV-Farbraum. Der HSV-Farbraum wird gewählt, weil er geeigneter für die Filterung nach bestimmten Farben ist und das ist der Fall in diesem Algorithmus.

```
// Convert to HSV- Colorspace
hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
```

Es werden zwei Masken konstruiert. Unterhalb sehen wir den Code, wie die Maske für die Blaufarbe aussieht. "blue_lower " entspricht für Blau den HSV-Werten für die untere Grenze, "blue_upper " ist die obere Grenze. Somit wird eine Maske erstellt, die alle Objekte, deren Farbe innerhalb dieses Grenzwertes liegt, erkennt. Die nächsten Statements (Anweisungen) führen eine Reihe von Erweiterungen durch, um alle in der Maske verbleibenden kleinen Flecken zu entfernen.

```
mask_b = cv2.inRange(hsv, blue_lower, blue_upper)
mask_b = cv2.erode(mask_b, None, iterations = 2)
mask_b = cv2.dilate(mask_b, None, iterations = 2)
```

Das Gleiche wird auch bei der pinken Farbe implementiert. Wenn aber die Grenzen der HSV-Werte nicht genau definiert sind, kann das zu Erkennungsproblemen führen. Das bedeutet, dass andere Objekte oder Flächen, die wir nicht analysieren wollen, wahrgenommen werden können. Es ist sehr wichtig, die Grenzen der HSV-Werte so genau wie möglich zu definieren. Das ist möglich durch die Verwendung des Farbkalibrierungs-Algorithmus, bevor der Haupt-Algorithmus ausgeführt wird. Die Erkennung der Objekte nur durch die Farbe wäre nicht ausreichend, weil vielleicht andere Bauteile oder Komponenten die gleiche oder ähnliche Farbe haben könnten. Um die Objekte exakter zu identifizieren, werden die Konturen untersucht. OpenCV stellt zahlreiche Funktionen zur Verfügung, um die Konturenerkennung zu vereinfachen [38].

6.4.4.2 Kreiskonturen

In diesem Abschnitt wird das konturbasierte Trackingverfahren der Objekte erklärt. Eine Kontur bezeichnet die Form eines Objektes. Die geometrische Figur unserer Objekte ist ein Kreis mit einem konstanten Radius für alle Objekte. Der Kreis wird auch in OpenCV durch seinen Radius und Mittelpunkt beschrieben. Zum Auffinden von Konturen wird die Methode

```
cnts = cv2.findContours(mask_b, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
```

verwendet. Die gefundenen Konturen stellen nicht immer vollständige rundliche Formen dar. Aber für jede z. B. blaue Farbe, die die Kamera erkennt, wird ein Kreis darum (rund um die Farbe) gezeichnet und man bekommt einen Radius dieses Kreises. Daher muss der Radius der Markierungen größer als die gefundene Kontur sein. In Abbildung 6.28 wird das durch ein Beispiel illustriert.

Die Methode, die verwendet wird, schließt alle Punkten mit minimale Fläche ein:

```
((x_t, y_t), radiusl_t) = cv2minEnclosingCircle(cnst[c])
```

Alle gefundenen Kreise werden durch ihren Radius verglichen. Die Kreise, die den minimalen vorbestimmten Radius haben, werden weiter analysiert. Wenn die kreisförmigen Konturen ähnlich wie die wirklichen Objekte sind, dann werden die Koordinaten zu X und Y zugewiesen. Mittels Koordinaten wird der Mittelpunkt des Objekts ermittelt. Daher wird es möglich sein, zu eruieren, wo die Objekte sich gerade befinden.

6.4.4.3 Die Farbkalibrierung

In diesem Abschnitt wird der Farbkalibrierungs-Algorithmus beschrieben. Bei der Verwendung des HSV-Farbraums ist es einfacher die Objekte durch ihre Farbe zu erkennen. Es ist notwendig

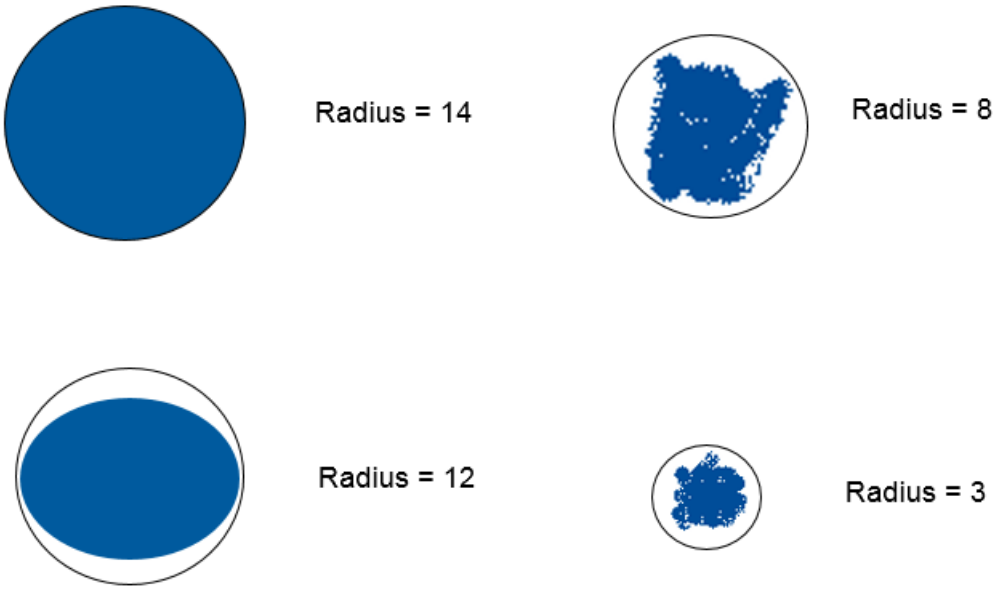


Abbildung 6.28: Kreiserkennung mit größten Radien

für jede Objektfarbe einen unteren und oberen HSV-Wert (Grenzwerte) zu definieren. Für manche Farben ist es möglich, durch Internetrecherche die HSV-Werte zu finden. Das kann nur funktionieren, wenn zufällig die Lichtverhältnisse und die gewählte Farbe ziemlich gleich sind. Von der Erfahrung her ist das kaum möglich. Außerdem war die Idee, dass der Demonstrator nicht nur für ein bestimmtes Lichtverhältnis geeignet sein kann. Deswegen entstand die Idee einen Farbkalibrierungs-Algorithmus zu kreieren. Ziel dieses Algorithmus ist es ein Fenster mit Trackbars zu haben, das alle HSV-Werte (obere bzw. untere Grenzwerte) setzen kann. Das ermöglicht die HSV-Werte vor Ort anzupassen. Hier brauchen wir keine Wiederholung von allen Schritten und Paketen, die wir brauchen, weil sie im Haupt-Algorithmus 6.4.4 beschrieben sind. Als erstes wird das Trackbarfenster erstellt und genannt:

```
cv2.namedWindow("Kalibrierungsbar")
```

Für die einzelnen Kanäle der HSV-Farbraums:

```
// Lower Range of H
cv2.createTrackbar("Low_H", "Kalibrierungsbar", minWert_H, maxWert_H)

// Upper Range of H
cv2.createTrackbar("Up_H", "Kalibrierungsbar", minWert_H, maxWert_H)
```

So wird weiter für alle Kanäle die gleiche Logik implementiert. Dadurch wird das Trackbarfenster oder Kalibrierungsfenster erstellt. Für die Funktionalität sind noch ein paar Schritte wichtig.

Die Konvertierung zum HSV-Farbraum und zum Abrufen der aktuellen Positionen von allen Trackbars erfolgt so:

```
H_Low = cv2.getTrackbarPos("Low_H", "Kalibrierungsbar")
S_Low = cv2.getTrackbarPos("Low_S", "Kalibrierungsbar")
V_Low = cv2.getTrackbarPos("Low_V", "Kalibrierungsbar")

H_Up = cv2.getTrackbarPos("Up_H", "Kalibrierungsbar")
S_Up = cv2.getTrackbarPos("Up_S", "Kalibrierungsbar")
V_Up = cv2.getTrackbarPos("Up_V", "Kalibrierungsbar")
```

Nachher kommt noch die gleiche Maske wie im Haupt-Algorithmus. Das ist in der Abbildung 6.29 ersichtlich.

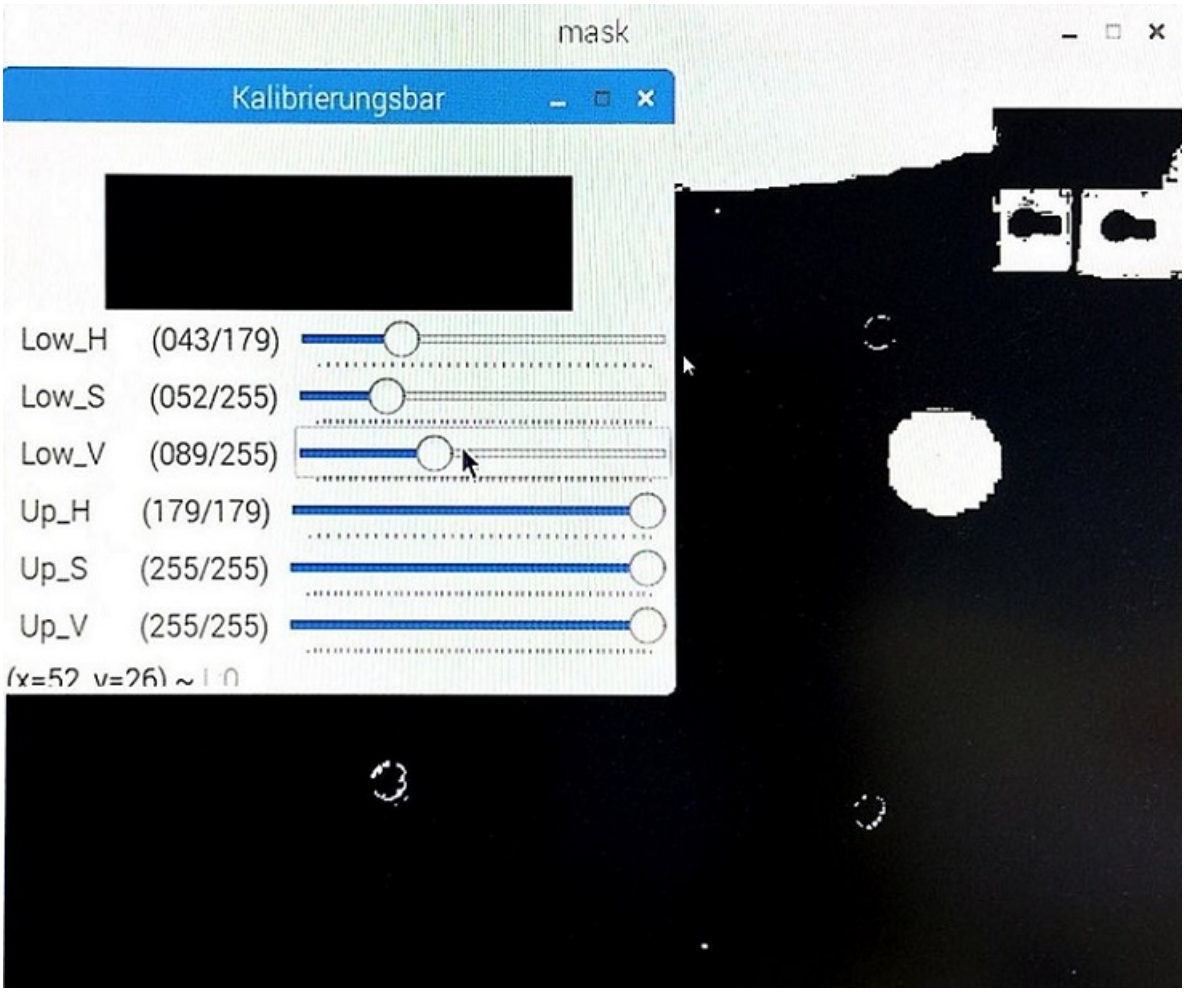


Abbildung 6.29: Die Kalibrierung der HSV-Werte.

6.4.4.4 Kameraverifikation

Wie bereits im Kapitel 6.26 angeschnitten, ist der so genannte Kameraverifikations-Algorithmus dafür gebaut, um die Kamera zu testen. Ziel ist, die Kamera vor der Zustandsbewertung der VSAs zu testen. Das bedeutet, bevor wir die Positionen der Objekte von der Kamera bekommen, wird die Kamera getestet. Durch diesen Test können wir bei jedem Zyklusbeginn schon vorher wissen, ob die Kamera funktioniert. Das erhöht die Sicherheit der Kamerafunktion. Aber wir erreichen nicht das gewünschte Sicherheits-Level. RGB-LEDs werden von RPI über I/O-Pins angesteuert. I/O-Pins sind die digitalen Pins von Raspberry PI. Damit man die Pins ansprechen bzw. programmieren kann, muss die GPIO-Bibliothek importiert werden.

```
import RPI.GPIO as GPIO
```

Zwei Methoden spielen die Zentrale Rolle:

```
Camera.Check_LED()
Camera.Check(frame)
```

Die erste Methode startet die Umschaltung der RGB-LEDs. Die Umschaltung passier zufällig. Das ermöglicht der Zufallsgenerator. Dafür importieren wir die geeignete Biblothek.

```
import random
```

Weiterhin müssen wir die Pins als Ausgabe (Output) definieren.

```
// LED 1
GPIO.setup(16, GPIO.OUT, initial = GPIO.LOW) // Blue
GPIO.setup(20, GPIO.OUT, initial = GPIO.LOW) // Green
GPIO.setup(21, GPIO.OUT, initial = GPIO.LOW) // Red
```

Die Output-Funktionen stehen dann uns zur Verfügung. "LOW " bedeutet den Pin ausschalten. "HIGH " heißt einschalten. Nun haben wird die Pins definiert und auf "LOW " initialisiert [12]. Zunächst werden die Pins zufällig umgeschaltet.

```
r = random.choice([der Pinszahl])
GPIO.ouput(r, GPIO.HIGH)
```

Nun kommt die zweite Methode, die für den Kameratest verantwortlich ist. Hier werden ein paar Schritte übersprungen, weil sie schon erklärt worden sind, wie die HSV-Grenzwerte, Bildfilterung, die Maske und so weiter. Die Positionen von RGB-LEDs werden mittels Positionskalibrierung ermittelt und dann auf "pos[] "Array eingetragen, womit wir die genauen Positionen von LEDs kennen. Wenn die Umschaltung und die Position der LEDs übereinstimmt, wissen wir, dass die Kamera richtig funktioniert. Aber das heißt noch immer nicht, dass die Kamera fehlersicher ist. Welche Fehler kommen können, wird bei der Analyse der Fehlerarten beschrieben.

6.4.4.5 Die Positionskalibrierung

Hauptaufgabe dieses Algorithmus ist die Erkennung der Positionen von RGB-LEDs. Die Positionen sind wichtig für die Kameraverifikation. Das heißt, vor der Bildaufnahme wird die Kamera getestet. Das geschieht mit Hilfe von RGB-LEDs und deswegen ist es bedeutungsvoll die genauen Positionen zu ermitteln. Erwähnenswert ist es auch, dass, wenn sich die Kamera aus irgendeinem Grund von ihrer Befestigungsposition bewegt, die von ihr gespeicherten Positionen der RGB-LEDs nicht richtig sein können. Dies kann am häufigsten sein, wenn sie anderswohin transportiert wird. Der Algorithmus ist im Python-Quelltext "calculatePos.py " geschrieben.

6.5 Demonstration

Die Demonstration von User Case wird hier beschrieben. Damit es verständlicher wird, werden alle möglichen Zustände, die ein VSA haben kann, beschrieben. Wie auch schon bei der Implementierung zu sehen war, besteht der Demonstrator aus drei VSAs. Die Zustände einer VSA werden durch LED-Türme repräsentiert und die Rolle der Ersatzobjekte wird durch seine Farbe definiert. In unserem Fall ist das blaue Objekt ein Stellvertreter der Roboter und das pinke Objekt ersetzt den Arbeiter. Mobile Roboter und Arbeiter können sich in VSA2 und VSA3 bewegen und miteinander arbeiten. VSA1 ist nicht für die Arbeiter erlaubt. Jede VSA verfügt über zwei Notausschalter. Die Positionen von Robotern und von Menschen werden von der Kamera erkannt und durch eine Übertragungsfunktion (über OPC UA) auf PLS(Programmable Logic Controller)-Simatic S7 1500 übertragen. Somit ist es möglich die Sicherheit für die Anlage dynamisch bzw. selektiv zu gewährleisten. Die Repräsentierung der Zuständen wird durch die vier unterschiedlichen Statuslampen (LED-Turm-Farben) und aus deren Kombination dargestellt.

Jede VSA kann aktuell bis zu neun Zustände haben. Die Zustände stellen wir durch die Statuslampe dar:

- **Rot** - Wenn der LED-Turm rot leuchtet, sind die betroffenen Maschinen in diesem Arbeitsbereich (VSA) im sicheren Zustand. Der Arbeitsfluss ist in diesem Bereich unterbrochen. Dieser Zustand tritt ein, wenn einer von den beiden Not-Aus-Tastern desselben VSA gedrückt wird und sich gerade eine oder mehrere Maschinen dort befinden.

- **Blau** - wenn der LED-Turm blau leuchtet, sind die beiden Not-Aus-Tastern desselben VSA gedrückt.

- **Gelb** - das bedeutet, dass sich die zwei anderen Arbeitsbereichen (VSAs) gerade im sicheren Zustand befinden und nicht betreten werden können. Oder wenn die gesamte Anlage in Safe-State gesetzt ist

- **Grün** - wenn der LED-Turm grün leuchtet, sind die Maschinen (z. B. Roboter) in diesem Bereich im normalen Arbeitsmodus. Das bedeutet, in diesem Arbeitsbereich der VSA

besteht keine Gefahr. Dieser Zustand tritt ein, wenn die Anlage aktiviert und keiner von den beiden Not-Aus-Tastern von dieser VSA gedrückt ist.

- **Rot-Blau** - wenn der LED-Turm rot und blau leuchtet, sind die betroffenen Maschinen in diesem VSA im sicheren Zustand. Der Arbeitsfluss ist in diesem Bereich unterbrochen. Dieser Zustand ein, wenn beide Not-Aus-Taster derselben VSA gedrückt werden.
- **Rot-Gelb** - diese VSA und mindestens noch eine andere VSA sind im sicheren Zustand.
- **Grün-Blau** - beide Not-Aus-Taster sind gedrückt, aber der Zustand in diesem VSA ist im normalen Arbeitsmodus.
- **Grün-Gelb** - normaler Arbeitsmodus in diesem VSA, aber die restlichen VSAs befinden sich in sicherem Zustand.
- **Grün-Gelb-Blau** - normaler Arbeitsmodus in diesem VSA, aber die restlichen VSAs sind im sicheren Zustand und es sind gleichzeitig beide Not-Aus-Taster gedrückt.

Nr.	Zustände	Beschreibung der Signalzustände
1	Rot	Diese VSA ist im sicheren Zustand. Arbeitsfluß is in diesem Bereich unterbrochen.
2	Blau	In dieser VSA sind beide Not-Aus Taster gedrückt.
3	Gelb	Das heißt es, dass zwei andere Arbeitsbereichen (VSA) in sicheren Zustand gerade befindet und können nicht betrettet werden.
4	Grün	Die Roboter sind in diesem VSA im normalen Arbeitsmodus. Das heißt in diesem Arbeitsbereich (VSA) besteht kein Gefahr.
5	Rot Blue	Beide Not-Aus Taster gedrückt und Arbeitsfluß is in diesem Bereich unterbrochen.
6	Rot Gelb	Das bedeutet, dass alle VSAs im sicheren Zustand befinden.
7	Grün Blau	Beide Not-Aus Taster gedrückt aber der Zustand in diesem VSA im normalen Arbeitsmodus.
8	Grün Gelb	Noramle Arbeitsmodus in diesem VSA aber die anderen restlichen VSAs befinden sich in sichern Zustand.
9	Grün Gelb Blau	Noramle Arbeitsmodus in diesem VSA aber die anderen restlichen VSAs sind im sichern Zustand und gleichzeitg sind beide Not-Aus Taster gedrückt.

Abbildung 6.30: Signalzustände von VSA.

KAPITEL 7

ERGEBNISSE

Nachfolgend erfolgt die Bewertung von einzelnen Systemen sowie deren Zykluszeiten. Zum Schluss werden die Ergebnisse zusammengefasst ,welche experimentell demonstriert wurden sowie die Schwachstellen von Sicherheitssystemen aufgezeigt

7.1 Bewertung des Simatic-Systems

Da das Simatic Automatisierungssystem schon lange am Markt ist und eine bewährte Lösung für die Automatisierung bietet, stellte der Demonstrator Aufbau kein Problem dar. TIA Portal bietet einen Zugriff auf das gesamte Automatisierungsprojekt und darüber hinaus ein anwenderfreundliches Interface. Im Unterschied zum Simatic-System ist das TIA Portal noch nicht so lange am Markt und hat noch ein paar Schwachstellen. So hat das Herunterladen des Anwenderprgramms auf Simatic-Systems oft viel Zeit in Anspruch genommen. Es ist selten passiert, dass das Programm abgestürzt ist. Mit dem TIA Portal ist es relativ einfach ein Projekt anzulegen sowie Hardwarekonfiguration und Parametrisierung durchzuführen. Außerdem wird für das Anwenderprogramm eine Vielzahl an Programmiersprachen angeboten, die einfach zu erlernen sind.

7.2 Bewertung des Kamerasystems

Für die Bildbearbeitung und die Übertragung der Objektkoordinaten auf das Simatic System wurde ein Raspberry Pi Version 3 verwendet. Neben dem Raspberry Pi wurde für dieses Projekt eine Kamera verwendet, die mit Raspberry Pi über den CSI Camera Connector verbunden wird. Raspberry Pi ist ein Einplatinencomputer, der ursprünglich als Lernplattform entwickelt wurde. Trotz seiner kleinen Abmessungen ist eine ziemlich starke Hardware auf der Platine

verbaut. Die Schwachstellen sind hierbei, dass nach längerer Zeit in Betrieb die Temperatur gestiegen ist. Dies hatte einen negativen Einfluss auf die Geschwindigkeit. Bei den Änderungen von Lichtverhältnissen war eine kurze Verzögerung an der Verarbeitung zu merken.

7.3 Zykluszeit

Das Simatic-System erlaubt die manuelle Einstellung von Mindest- und maximale Zykluszeit. Unter dem Begriff Zykluszeit (in Simatic-System) versteht man die Zeit, die die CPU braucht, um die Ein und Ausgänge zu aktualisieren, die Bearbeitung des zyklischen Programms und das Warten bis die Mindestzykluszeit erreicht ist. Außerdem kann man die kürzeste, aktuelle und längste Zykluszeiten online während des Betriebs über das TIA Portal sehen. Beim Kamerasystem wurden nicht die Zykluszeit von RPi gemessen, sondern die Zeit, die der Haupt-Algorithmus für die Bearbeitung benötigt. Dies kann unterschiedliche Laufzeiten haben. Um die Zeitdauer herauszufinden und eine entsprechende Zykluszeit zu definieren, wurde ein erstes Parametrisierungsexperiment durchgeführt. Dazu wurde die Zykluszeit der Kamerafunktionalität im laufenden Betrieb ermittelt. Da die Verarbeitungszeit seitens des Raspberry Pi eine gewisse Varianz aufweist, wurde die Zykluszeit 500-mal erfasst. Basierend auf den gemessenen Zeiten konnte der Minimal-, Mittel- und Maximalwert der Zeit des Algorithmus ermittelt werden:

- Minimalwert : 141,235 (ms)
- Mittelwert : 248,782331 (ms)
- Maximalwert : 327,337 (ms)

Bei dieser Zeit handelt es sich um die Laufzeit, die der Haupt-Algorithmus benötigt um die Objekte zu erkennen und deren Koordinaten in das Simatic-System zu übertragen.

7.4 Demonstrator

Das Ergebnis dieser Arbeit ist ein experimenteller Demonstrator, mit dem man eine schutzzaunlose Arbeitsumgebung simulieren kann. Das erfolgte durch das Kamera- und Simatic-System, die die Arbeitsbereiche in sogenannte Safety Areas virtuell aufteilen. Um die Sicherheit für Menschen und beteiligte Maschinen zu gewährleisten, wurde eine Sicherheitsfunktion erstellt. Die Sicherheitsfunktion wird vom Sicherheitssystem ausgeführt. Das Sicherheitssystem besteht aus dem Simatic System der Übertragung sowie dem Kamerasystem. Die Fehlerrate ist bis auf SIL 3 reduziert. Die Übertragungsfunktion erfolgt über eine Standardkommunikation (Ethernet). Durch die Implementierung des Black Channel Prinzips wirdder Übertragungsweg abgesichert. Dies ist seit mehreren Jahren in Verwendung und ist sehr stabil. Das Kamerasystem ist nicht so sicher wie die vorigen Systeme aber um das Sicherheitsniveau zu steigern, wurde die Idee mit den

RGB-LEDs (Kameraverifikation) umgesetzt. Es sind 4 RGB-LEDs auf dem Basisfeld eingebaut, welche auf der Kamera sichtbar sind. Das Ziel ist, die Kamera vor der Zustandsbewertung der VSAs zu testen. Durch diesen Test kann man bei Zyklusbeginn schon wissen, ob die Kamera funktioniert.

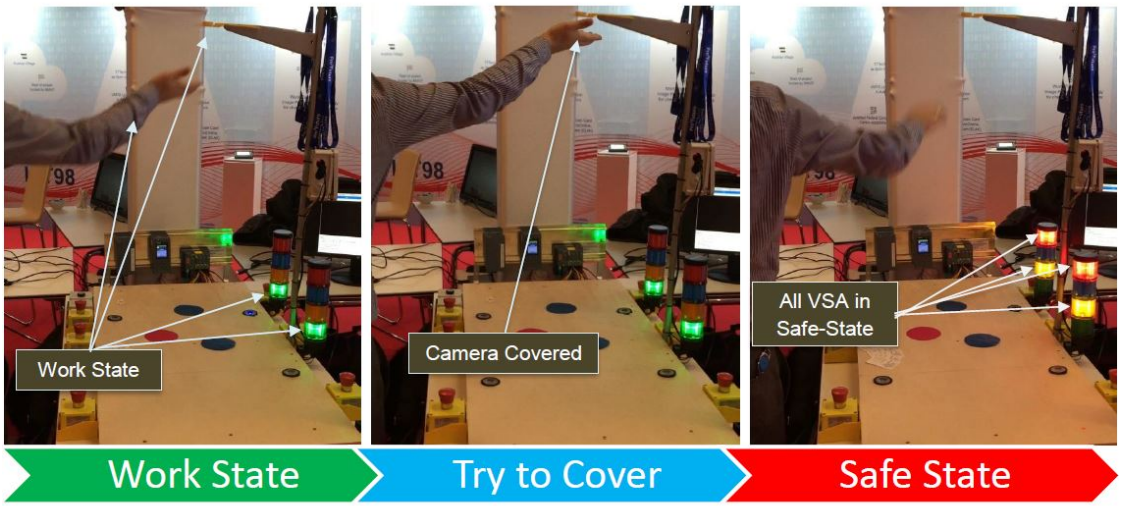


Abbildung 7.1: Safe State.

Dadurch ist es möglich einige Fehler zu erkennen, die die Sicherheitsfunktion verletzen können. Die Gefahren/Fehler, die vom Kamerasystem erkannt werden können, sind:

- Bedeckung von Kamera
- Kein Bild
- Statisches Bild
- Änderung der Lichtverhältnisse [1]
- Farb-, Kontur-, Positionserkennung

Bei jedem zyklischen Durchlauf des Algorithmus wird am Anfang die Kamera verifiziert. Durch einen Zufallsgenerator werden die RGB-LEDs umgeschaltet. Die Kamera überprüft die Farbe und die Position der LEDs. Dadurch können die oben genannten Fehler erkannt werden. Der dynamische Ansatz für die Sicherheit war eines der Ziele dieser Arbeit. Das wurde dadurch erreicht, dass die Sicherheit in Gefahrensituationen nur auf die entsprechende VSA einwirkt. Das System schafft somit:

- Selektive Einwirkung von Not-Aus-Taster

[1] Die Änderungen, die bei RGB-LEDs Erkennung Einwirkung haben!

- Safe-Sate für die Betretung von verbotene Bereiche
- Bewegungsverlauf für Erkennung/Prävention sicherheitskritischer Bewegungen

Selektive Einwirkung von Not-Aus Taster bedeutet, dass bei der Betätigung des Schalters nur jene Objekte angesteuert werden, welche sich vorübergehend oder statisch in der Safe Area befinden. Dies wurde mit unterschiedlichen Szenarios getestet. Die Testfälle sind in der Abbildung 7.2 dargestellt.

Testfall (TF)	Selektive Einwirkung auf VSAs					
	VSA	Not-Aus Taster Zustand	Situation	VSA Zustand Ist	VSA Zustand Soll	Folgen
TF1	VSA1	NG	R1 Da	Grün	Grün	OK
TF2	VSA1	G	R1 Da	Rot	Rot	OK
TF3	VSA1	NG	R1 nicht Da	Grün	Grün	OK
TF4	VSA1	G	R1 nicht Da	Grün	Grün	OK
TF5	VSA2	NG	R2 Da	Grün	Grün	OK
TF6	VSA2	G	R2 Da	Rot	Rot	OK
TF7	VSA2	NG	R2 nicht Da	Grün	Grün	OK
TF8	VSA2	G	R2 nicht Da	Grün	Grün	OK
TF9	VSA3	NG	R1 Da	Grün	Grün	OK
TF10	VSA3	G	R1 Da	Grün	Rot	Fehler
TF11	VSA3	NG	R1 nicht Da	Grün	Grün	OK
TF12	VSA3	G	R1 nicht Da	Grün	Grün	OK
TF13	VSA1	NG	R1 & R2 Da	Grün	Grün	OK
TF14	VSA1	G	R1 & R2 Da	Rot	Rot	OK
TF15	VSA2	G	R2 Da - Abgedeckt	Grün	Rot	Fehler

Not-Aus Taster Zustand	
NG -	Nicht Gedrückt
G -	Gedrückt

Roboter	
R1 -	Erste Roboter
R2 -	Zweite Roboter

Abbildung 7.2: Test-Szenario für die selektive Einwirkung.

In der Abbildung 7.2 sind 15 Testfälle zu sehen. 13-mal wurde die Selektivität garantiert. Das bedeutet, dass das System korrekt reagieren konnte. In zwei Fällen konnte das System nicht richtig reagieren. Testfall (TF10) zeigt, obwohl der Not-Aus Schalter der VSA3 gedrückt wurde und sich gleichzeitig ein Roboter (R1) dort befand, dass die VSA3 nicht in sicheren Zustand gesetzt wurde. Der Grund hierfür war, dass in diesem Bereich keine gleichmäßigen Lichtverhältnisse herrschten. Daher konnte die Kamera den Roboter nicht erkennen. Im anderen Fall konnte der Roboter nicht erkannt werden, weil die Kamera abgedeckt war. Als Folge konnte diese VSA nicht in den Safe-State gehen.

Safe-Sate für die Betretung von verbotenen Bereichen bedeutet, dass wenn ein Mensch eine VSA betritt, der für Menschen verboten ist, er in den Safe- State übergeht. Dasselbe gilt für Roboter. VSA1 ist für die Menschen verboten ("Kollaboration 1 "). Wenn ein Mensch diese VSA betritt, dann wird sie in den Safe-State gehen. Dasselbe macht die VSA 3 für die Roboter. Auch hierzu wurden wieder mehrere Experimente durchgeführt. 10-mal wurde die VSA1 getestet, ob das System das Eindringen von Menschen erkennt. Das System konnte immer erkennen, wenn

der Mensch die VSA1 betritt und setzte diesen Bereich in den Safe-State. Für die VSA3 wurden ebenfalls 10 Versuche gemacht. Auch hier konnte das System immer erkennen, wenn ein Roboter diesen Bereich betrat.

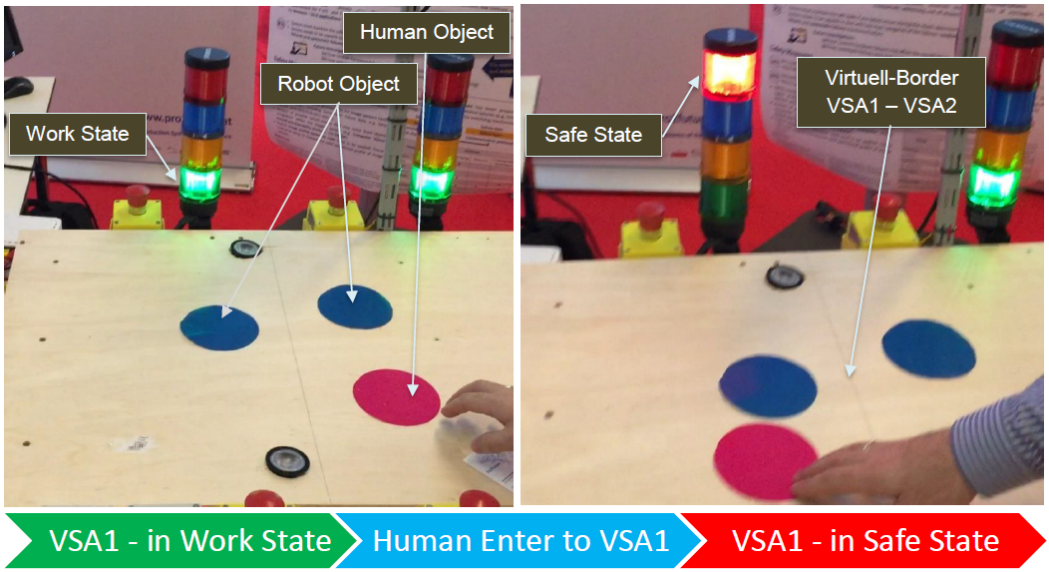


Abbildung 7.3: VSA1 verbotene Bereich für Menschen.

Testfall (TF)	Sichere Zustand für die Betretung von verbotene Bereiche			
	VSA	Situation	VSA Zustand Ist	VSA Zustand Soll
	VSA1 ist für Menschen Verboten			
TF1	VSA1	H Da	Rot	Rot
TF2	VSA1	H nicht Da	Grün	Grün
	VSA3 ist für Roboter Verboten			
TF3	VSA3	R1 Da	Rot	Rot
TF4	VSA3	R1 nicht Da	Grün	Grün

Human	
H -	Human

Roboter	
R1	Roboter 1

Abbildung 7.4: Test-Szenario für die Betretung von verbotene Bereiche.

Bewegungsverlauf für Erkennung/Prävention sicherheitskritischer Bewegungen wurde als eine weitere Funktionalität dieses Demonstrators gebaut. Die Idee dahinter ist, dass wenn sich eine VSA in einem Safe-State befindet und in die Richtung diese VSA die Bewegungstrajektorien (Bewegungsverlauf) eines Manschens bzw. Roboters zeigt (Mensch oder Roboter bewegen in Richtung Safe-State VSA), dann wird ein Warnsignal gegeben, wenn die Anzahl der vordefinierten Schritte überschritten wird. Wie man weiß, wird die Position von einem Objekt durch die Koordinaten (X,Y) dargestellt. Dadurch wird auch die Anzahl der

Schritte berechnet. Um zu bestimmen wie "groß ëin Schritt in dem Demonstrator sein kann, wurden verschiedene Experimenten gemacht. Daraus wurde festgestellt, dass es in unserem Fall optimal ist, wenn "Xüm 5 (Einheiten) größer und "Ynicht kleiner wird, dann wird das als ein Schritt gerechnet. Nach fünf Schritten in die Richtung der VSA3 (sieh Abbildung 7.5) wird ein Warnsignal (Prävention-Meldung) geben. Das ist eine Prädiktiv-Sicherheitsmeldung bzw. Prävention von sicherheitskritischen Bewegungen.

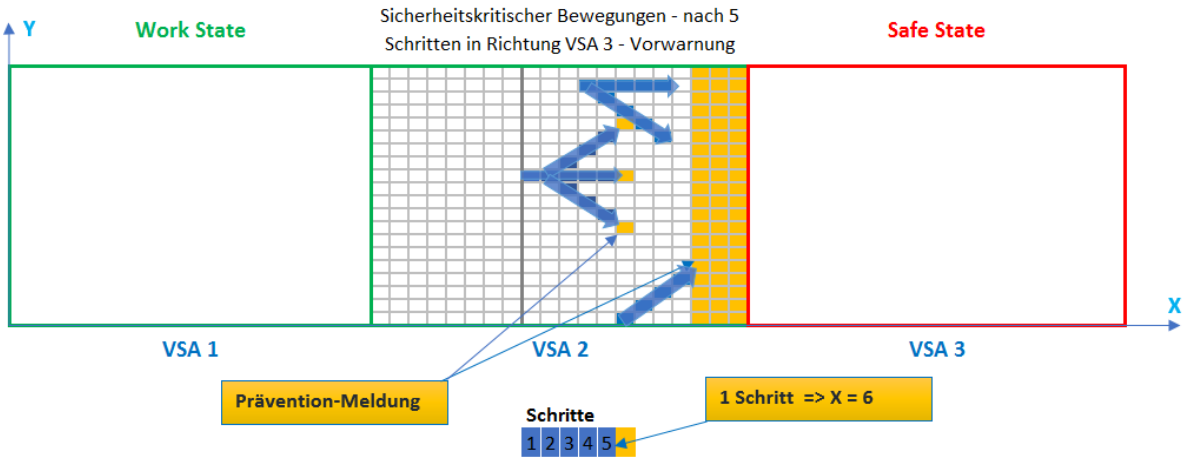


Abbildung 7.5: Bewegungsverlauf für die Erkennung/Prävention sicherheitskritischer Bewegungen.

Um das System auf die Korrektheit bei der Erkennung der sicherheitskritischen Bewegungen zu testen, wurden einige Experimente durchgeführt. Wie auch in der Abbildung ersichtlich, wurden die Experimente von unterschiedlichen Punkten mit unterschiedlichen Bewegungsrichtungen durchgeführt. Von 20 Test-Szenarios wurden die sicherheitskritischen Bewegungen nur 8-mal richtig erkannt. Der Grund dafür war, die Geschwindigkeit von Bewegungen schnell bzw. nicht konstant war. Das führte dazu, das die Bewegungen oft als nicht sicherheitskritisch bewertet wurden. Die Ursache dieses Problem sind die langsamen Bearbeitungszykluszeiten des Kamerasystems. Deshalb konnten die schnelleren Bewegungen nicht rechtzeitig erkannt werden

KAPITEL 8

ZUSAMMENFASSUNG UND AUSBLICK

Hier erfolgt eine Zusammenfassung der Arbeit sowie ein Ausblick auf mögliche Erweiterungen des Demonstrators

In dieser Arbeit wurde untersucht wie der traditionelle, statische Ansatz für Fail-Safe Operationen für die Anwendung in zukünftigen, dynamischen Produktionsszenarien angepasst werden kann. Somit wurde ein Szenario entworfen, welches die Interaktion von Menschen und Maschine in zukünftigen Arbeitsumgebungen skizziert. Die Arbeitsumgebung wurde virtuell in drei VSAs geteilt. Jede VSA umfasst einen Teil des kollaborativen Arbeitsraums. In jeder VSA sind zwei Not-Aus-Taster instantiiert , welche mit dem Simatic-System verbunden sind. Die Bewegungen von Objekten (also Roboter oder Menschen) in den VSAs werden durch die Kamera überwacht. Dadurch kann festgestellt werden, welches Objekt sich gerade in welcher VSA befindet und wie sich die Objekte bewegen. Sofern sich eine VSA in einer Gefahrensituation befindet und einer der beiden Not-Aus Taster gedrückt sind, ist es dem System möglich die spezifische VSA in einen sicheren Zustand zu bringen, wobei andere VSAs unbeeinflusst bleiben. Es konnten auch verbotene Bereiche für Mensch und Maschine realisiert werden. Wenn Mensch oder Maschine dagegen verstoßen, dann setzt das System die entsprechende VSA in einen sicheren Zustand. Außerdem konnte eine Prävention von sicherheitskritischen Bewegungen (siehe Kapitel 7) realisiert werden. Der Demonstrator wurde mit Standard Automatisierungs- und Sicherheitskomponenten der Simatic Automatisierungsfamilie realisiert. Für den Not-Halt sind Not-Aus-Taster mit 1oo2 Verdrahtungen eingesetzt. Raspberry Pi und die Kamera sind für die Überwachung der Objektbewegungen und Bildverarbeitung eingesetzt. LED-Türme repräsentieren die Zustände der VSAs. Auf Basis des Szenarios wurde eine Sicherheitsanalyse durchgeführt. Um die Analyse zu machen,

wurde das Sicherheitssystem in drei Teilsysteme bzw. drei Fehlerarten geteilt. Diese sind die Fehler, die aus dem Simatic-System während der Übertragung und aus der Bildverarbeitung kommen können.

- **Simatic-System** - beim Simatic-System sind keine weiteren Sicherheitsmaßnahmen erforderlich, F-Geräte verfügen bereits über alle erforderlichen Maßnahmen.
- **Übertragungsfunktion** - die Datenübertragung wird in solchen Szenarien immer wichtiger, daher braucht man eine zuverlässige Übertragung von Daten. Um die Übertragung von Daten abzusichern, wird eine Black Channel Architektur vorgeschlagen. Dadurch können die kommunikationsbezogenen Fehler erkannt und eine sichere Übertragung ohne zusätzliche Verdrahtung realisiert werden.
- **Bildverarbeitung** - ohne weiteres das ist dies die größte Herausforderung für die Erweiterung des Sicherheitssystems. Da in dieser Arbeit kein Safety Kamerasystem verwendet wurde, war nicht möglich das gewünschte Sicherheitslevel zu erreichen. Im Folgenden wird beschrieben wie die Sicherheit der Kamera verbessert werden kann.

Die Sicherheitsanalyse hat ergeben, dass das Kamerasystem bzw. die Bildverarbeitung die größte Schwachstelle des Sicherheitssystems ist. Um das Vertrauen in die Kamera zu steigern, wurde die Idee mit RGB-LED implementiert. Der Grund hierfür war die Kamera bei jeden Zyklusbeginn (vom Programm) zu verifizieren. Dadurch wurde erreicht, dass das Kamerasystem die Gefahren/Fehler, die im Kapitel 7 erwähnt sind, erkennen kann. Das machte das Kamerasystem sicherer, aber nicht so sicher wie es erforderlich ist. Durch diese Arbeit war es aber möglich einige Ergebnisse zu erreichen, wodurch die Anpassung an eine dynamische und industrielle Umgebung demonstriert werden konnte. Das System ist fähig eine selektive Einwirkung von Not-Aus Schaltern, Safe-State für die Betretung von verbotene Bereiche und Bewegungsverlauf für die Erkennung/Prävention sicherheitskritischer Bewegungen zu schaffen.

8.0.1 Ausblick für weitere Entwicklungen

Hier werden einige Aspekte betrachtet, wodurch das Kamerasystem performanter werden könnte. Ein wichtiger Aspekt bei einem Sicherheitssystem ist auch die Reaktionszeit des Systems. Dies umfasst in unserem Fall die Erfassung von Kontextdaten, Verarbeitung und die Reaktion des Systems. Um die Reaktionszeit des Kamerasystems zu verbessern, sollte man die Taktrate erhöhen. Anderes gesagt, RPi übertakten. Das würde die Zykluszeiten verkürzen. Dabei ist die Temperatur zu berücksichtigen. Ein performanterer Algorithmus würde auch einen großen Einfluss auf die Bearbeitungszeit des Programms haben. Ein anderer wesentlicher Faktor ist die Programmiersprache, in der es programmiert wird. In dieser Arbeit wurde Python verwendet aber eine andere, ßchnellere Programmiersprache wie etwa C oder C++ würde noch kürzere Zykluszeiten schaffen. Ein weiteres Erweiterungspotential ist die eindeutige Identifizierung der

Objekte. Das wäre sehr bemerkenswert, damit dem System im Fehlerfall bewusst wird, welche Roboter zu stoppen ist. Die Idee für eine solche Erweiterung ist die Verwendung von QR-Codes. Dies ist ein zweidimensionaler Code, der auf den Objekten (Roboter, Mensch) angebracht werden könnte. Dadurch kann eine Identitätsfeststellung möglich sein.

8.0.2 Vorschläge für künftige sichere Kamera

Die Fehler, die aufgrund von Funktionsfehlern der Kamera oder beim Klassifizierungssystems auftreten können, sind Bildsensorprobleme oder Probleme sowie Rausch im Bild, Verzerrung, Schatten, Blendung, Reflexion, niedrige Kontrast usw. Das bedeutet, dass ein zukünftiges Kamerasystem in der Lage sein soll mit zwei Arten von Fehlern, den zufälligen Hardwarefehlern sowie den systematischen Fehlern, umgehen zu können,. Um solche Fehler in einem Kamerasystem zu überwinden, werden einige relevante Forschungsthemen vorgeschlagen:

- Sicherheitsanalyse von Hardwarearchitekturen für Bildsensoren (mit höheren SIL / ASIL / einschließlich Komponentenfehlerdaten , z. B. Sony IMX324 mit ISO 22626: 2011 ASIL B).
- Analyse möglicher Fehlermodi, die aus der Objekterkennung innerhalb des Bildgebungssystems stammen können (z. B. Computer Vision Hazard and Operability Analysis - CV-HAZOP - Identifizierung von möglichst viele Gefahren).
- Es müssen explizite Regeln zur Erkennung der Bildqualität angewendet werden. Diese Regeln dienen zur Charakterisierung eines gültigen Betriebsprofils von Bildsensoren.

LITERATURVERZEICHNIS

[1] V. ALCÁCER AND V. CRUZ-MACHADO, *Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems*, Engineering Science and Technology, an International Journal, (2019).

[2] T. BAUERNHANSL, *Die vierte industrielle Revolution–Der Weg in ein wertschaffendes Produktionsparadigma*, in Handbuch Industrie 4.0 Bd. 4, Springer, 2017, pp. 1–31.

[3] G. BRADSKI AND A. KAEHLER, *Learning OpenCV: Computer vision with the OpenCV library*, Ö'Reilly Media, Inc.", 2008.

[4] G. CREECH, *Black channel communication: What is it and how does it work?*, Measurement and Control, 40 (2007), pp. 304–309.

[5] R. CUCCHIARA, C. GRANA, M. PICCARDI, A. PRATI, AND S. SIROTTI, *Improving shadow suppression in moving object detection with hsv color information*, in ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585), IEEE, 2001, pp. 334–339.

[6] DESIGN UND ELEKTRONIK VON MICHAEL VOLZ, *Prinzip Black Channel*, 2014. https://www.elektroniknet.de/design-elektronik/safety-generisch-geloest-114773-Seite-2.html Seite am 28.02.19 aufgerufen.

[7] DIVISION DIGITAL FACTOR, *Simatic S7-1500, ET-200MP, ET 200SP, ET 200AL, ET 200pro Kommunikation*, Industry Online Support, 2018.

[8] M. DOHI, K. OKADA, I. MAEDA, S. FUJITANI, AND T. FUJITA, *Proposal of Collaboration Safety in a Coexistence Environment of Human and Robots*, in 2018 IEEE International Conference on Robotics and Automation (ICRA), May 2018, pp. 1924–1930.

[9] U. DOMBROWSKI, C. RIECHEL, AND M. EVERS, *Industrie 4.0 – die Rolle des Menschen in der vierten industriellen Revolution*, Industrie, 4 (2014), pp. 129–153.

[10] P. DR. PETER WENZEL AND P. INTERNATIONAL, *Profisafe Black Channel*, 2012. https://www.industr.com/de/safety-first-230179 Seite am 15.03.19 aufgerufen.

[11] F. EBELN, S. IDLER, G. PREDE, AND D. SCHOLZ, *Grundlagen der Automatisierungstechnik*, Festo Didactic GmbH und Co. KG, 2008.

[12] ELEKTRONIK KOPENDIUM, *Raspberry pi: Grundlagen der Energieversorgung / Stromversorgung*, 2015.
http://www.elektronik-kompendium.de/sites/raspberry-pi/1912111.htm Seite am 12.01.19 aufgerufen.

[13] P. FLORIAN, S. SEBASTIAN, M. ALEXANDER, O. JENS, W. STEFAN, B. BJÖRN, N. OLIVER, AND J. JÜRGEN, *Industrie 4.0 – Kommunikation mit OPC UA*, VDMA Verlag GmbH, 2017.

[14] J. FRANKE, *Handbuch Mensch-Roboter-Kollaboration*, Carl Hanser Verlag GmbH Co KG, 2019.

[15] J. FRYMAN AND B. MATTHIAS, *Safety of industrial robots: From conventional to collaborative applications*, in ROBOTIK 2012; 7th German Conference on Robotics, VDE, 2012, pp. 1–5.

[16] H. GALL, *Functional safety iec 61508 / iec 61511 the impact to certification and the user*, in 2008 IEEE/ACS International Conference on Computer Systems and Applications, March 2008, pp. 1027–1031.

[17] P. GLOGOWSKI, K. LEMMERZ, H. A, AND B. KUHLENKÖTTER, *Menschzentrierte simulation mit adaptiver kollisionsfreier roboterbahnplanung in der mensch-roboter-kollaboration*, 2018.
https://www.researchgate.net/publication.

[18] H.-T. HANNEN, *Beitrag zur Analyse sicherer Kommunikationsprotokolle im industriellen Einsatz*, PhD thesis, 2012.

[19] R. HENSSEN AND M. SCHLEIPEN, *Online-Kommunikation mittels OPC-UA vs. Engineering-Daten (offline) in Automationml - Eine Möglichkeit der Integration und Kombination*, 07 2014.

[20] H. HIRSCH-KREINSEN AND M. TEN HOMPEL, *Digitalisierung industrieller Arbeit: Entwicklungsperspektiven und Gestaltungsansätze*, Handbuch Industrie 4.0: Produktion, Automatisierung und Logistik, (2016), pp. 1–20.

[21] H. HIRSCH-KREINSEN, J. WEYER, AND M. WILKESMANN, *Industrie 4.0 als Technologieversprechen*, Soziologisches Arbeitspapier, (2016).

[22] INDUSTRIE DE, *Zukunft von Industrie 4.0*, 2018.
https://industrie.de/top/6637/ Seite am 24.02.19 aufgerufen.

[23] H. KAGERMANN, W. WAHLSTER, AND J. HELBIG, *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*, Abschlussbericht des Arbeitskreises Industrie, 4 (2013).

[24] C. KARAALI, *Speicherprogrammierbare (sps)-steuerungen*, in Grundlagen der Steuerungstechnik, Springer, 2018, pp. 157–179.

[25] D.-I. B. KASPER, *Weiterentwicklung sicherheitstechnischer Analyse-und Bewertungsmethoden für die industrie 4.0 Development of safety related security methods in the field of Industry 4.0*.

[26] T. KAUFMANN, *Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge: der Weg vom Anspruch in die Wirklichkeit*, Springer-Verlag, 2015.

[27] KUKA ROBOTIC, *Mobile robotik kmr iiwa*, 2017.
https://www.kuka.comhttps://www.kuka.com Seite am 03.04.19 aufgerufen.

[28] P. LÖW, R. PABST, AND E. PETRY, *Funktionale Sicherheit in der Praxis: Anwendung von DIN EN 61508 und ISO/DIS 26262 bei der Entwicklung von Serienprodukten*, dpunkt. verlag, 2011.

[29] T. MEANY, *Functional safety and industrie 4.0*, in 2017 28th Irish Signals and Systems Conference (ISSC), June 2017, pp. 1–7.

[30] H. MERCHANT AND D. AHIRE, *Industrial Automation using IoT with Raspberry Pi*, International Journal of Computer Applications, 168 (2017).

[31] A. MORDVINTSEV AND K. ABID, *Opencv-python tutorials documentation*, Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf, (2014).

[32] O. MOSER, *Einführung in computer vision mit opencv und python*, Codecentric AG, 2016.
https://blog.codecentric.de/2017/06/einfuehrung-in-computer-vision-mit-opencv-und-python/ Seite am 08.02.19 aufgerufen.

[33] PILZ GMBH AND CO, *Sichere mensch-roboter-kollaboration*, 2018.

[34] K. N. PLATANIOTIS AND A. N. VENETSANOPOULOS, *Color image processing and applications*, Springer Science & Business Media, 2013.

[35] T. RADEMACHER, *Industriekameras in Industrie 4.0 für eine effiziente Produktion*, Blaser AG, (2016).
https://www.baslerweb.com/de/vertrieb-support/downloads/downloads-dokumente/industrie-4-0/ Seite am 11.03.19 aufgerufen.

[36] H. RADSZUWEIT, J. KRUNKOWSKI, J. PFLÜGER, AND M. TISCHER, *Ein simatic basiertes kontrollsystem fuer die undulatoren des tesla röntgenlasers*, 2000.

[37] C. RAMSAUER, *Industrie 4.0–die produktion der zukunft*, vol. 3, 2013, pp. 6–12. http://www.forschungsnetzwerk.at.

[38] C. REUL, *Evaluation von methoden zur bildverarbeitung für objekterkennung am beispiel der klassifikation von bäumen*, 2015.

[39] M. RICHARDSON AND S. WALLACE, *Getting started with raspberry PI*, Ö'Reilly Media, Inc.", 2012.

[40] M. ROSSI, *Pure Python OPC UA / IEC 62541 Client and Server Python 2, 3 and pypy*, 2018. https://github.com/FreeOpcUa/python-opcua/blob/master/README.md Seite am 14.04.19 aufgerufen.

[41] A.-W. SCHEER, *Industrie 4.0: Wie sehen Produktionsprozesse im Jahr 2020 aus*, IMC AG, (2013).

[42] SCHULPHYSIKWIKI, THE FREE ENCYCLOPEDIA, *Dampfmaschine*, 2013. http://schulphysikwiki.de/index.php/Datei:Dampfmaschine2.gif Seite am 29.01.19 aufgerufen.

[43] SEMICONDUCTOR COMPONENTS INDUSTRIES, *Evaluating Functional Safety in automotive image sensors*, (2017). https://www.onsemi.com/pub/Collateral/TND6233-D.PDF, Seite am 06.05.19 aufgerufen.

[44] M. SICHERHEIT, *Risikograph für Bewertung des SIL*. http://www.maschinen-sicherheit.net/07-seiten/0351-risikograph-SIL.phpSeite am 27.02.19 aufgerufen.

[45] SIEMENS AUTOMATION COOPERATES WITH EDUCATION, *TIA Portal Modul 010-020 Bausteinarten bei SIMATIC S7-1200*, 2012.

[46] ——, *Globale Datenbausteine bei SIMATIC S7-1500*, 2017.

[47] SIEMENS DF FA AS, *System Architecture*, 2017. Siemens Intern.

[48] SIEMENS INDUSTRY ONLINE SUPPORT, *Emergency Stop up to PL e / SIL 3 with a Fail-Safe S7-1500 Controller*, 2017.

[49] SPS LEHRGANG, *Cpu: Central Processing Unit der SPS*, 05/2017. https://www.sps-lehrgang.de/sps/https://www.sps-lehrgang.de/sps/ Seite am 04.04.19 aufgerufen.

[50] SPSHAUS GMBH, *Neuerungen im SCL ab V14 SP1*, 2017.

[51] THE RASPBERRY PI FOUNDATION, *Camera module v2*, 2016.
https://www.raspberrypi.org/products/camera-module-v2/ Seite am 17.03.19 aufgerufen.

[52] M. TOM, *Functional Safety and Networking*, (2018).
https://ez.analog.com/b/engineerzone-spotlight/posts/functional-safety-and-networkingSeite am 06.05.19 aufgerufen.

[53] TÜV AUSTRIA AND FRAUNHOFER AUSTRIA, *Safety in Human-Robot Collaboration*, Oktober, 2016.
https://www.tuv.at/en/solutions/industry-energy/industry-40/ Seite am 24.04.19 aufgerufen.

[54] V. M. UND ANLAGENBAU, *Arbeit 4.0 – im Zentrum steht der Mensch*, 2017.
https://mensch-maschine-fortschritt.de/reportage/arbeit-4-0-im-zentrum-steht-der-mensch/ Seite am 05.03.19 aufgerufen.

[55] UNIVERSAL ROBOTS, *Der Zustimmtaster Notwendigkeit in der Mensch-Roboter-Kollaboration (MRK)*, 2017.
https://docplayer.org/60495748-Whitepaper-der-zustimmtaster-notwendigkeit-in-der-mensch-roboter-kollaboration-mrk.html Seite am 19.03.19 aufgerufen.

[56] V. VILLANI, F. PINI, F. LEALI, AND C. SECCHI, *Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications*, Mechatronics, 55 (2018), pp. 248–266.

[57] G. WELLENREUTHER AND D. ZASTROW, *Automatisieren mit SPS: Theorie und Praxis*, Springer-Verlag, 2005.

[58] S. WEYER, M. SCHMITT, M. OHMER, AND D. GORECKY, *Towards Industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems*, Ifac-Papersonline, 48 (2015), pp. 579–584.

[59] O. ZENDEL, M. MURSCHITZ, M. HUMENBERGER, AND W. HERZNER, *Cv-hazop: Introducing test data validation for computer vision*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2066–2074.