
BACHELORARBEIT

cand.ing. Yunbin Xu

**Spezifikation und Implementierung
eines Fahrzeugumfeld-
modells im Rahmen des
ATLAS-Projektes**

Mittweida, 2019

BACHELORARBEIT

Spezifikation und Implementierung eines Fahrzeugumfeldmodells im Rahmen des ATLAS-Projektes

Autor:
cand.ing. Yunbin Xu

Studiengang:
Elektro- und Informationstechnik

Seminargruppe:
EI15wA-BC

Erstprüfer:
Prof. Dr.-Ing. Jan Thomanek

Zweitprüfer:
Prof. Dr.-Ing. Christian Schulz

Einreichung:
Mittweida, 04.12.2019

Verteidigung/Bewertung:
Mittweida, 2019

BACHELOR THESIS

Specification and Implementation of a Vehicle Environment Model for the ATLAS project

author:
cand.ing. Yunbin Xu

course of studies:
Electrical and Information Technology

seminar group:
EI15wA-BC

first examiner:
Prof. Dr.-Ing. Jan Thomanek

second examiner:
Prof. Dr.-Ing. Christian Schulz

submission:
Mittweida, 04.12.2019

defence/ evaluation:
Mittweida, 2019

Bibliografische Beschreibung:

Xu, Yunbin:

Spezifikation und Implementierung eines Fahrzeugumfeldmodells im Rahmen des ATLAS-Projektes, 82 Seiten, Mittweida, Hochschule Mittweida, Fakultät Ingenieurwissenschaften, Bachelorarbeit, 2019

Referat:

In der Bachelorarbeit wird Spezifikation und Implementierung eines Fahrzeugumfeldmodells im Rahmen des ATLAS-Projektes beschrieben. Hierzu wird das Umfeldmodell definiert und erläutert, und der Aufbau des Umfeldmodells mit einer Programmiersprache abgeschlossen. Die Bachelorarbeit enthält auch Vorschläge zur Verbesserung der Generierung von CSV-Dateien und zum Speichern der Standortinformationen des Fahrzeugs. Außerdem wird der Zusammenhang zwischen Umfeldmodell und autonomem Fahren kurz beschrieben.

Vorwort

Die folgende Bachelorarbeit wurden im Rahmen eines Elektrotechnikstudiums an der Hochschule Mittweida angefertigt. Die Idee eines fahrerlosen Transportsystems auf der Schiene, entstand auf Grundlage der „Gedanken zur grundlegenden technischen Umgestaltung der Bahn“[1] von Prof. Dr.-Ing. Christian Schulz.

Die Aufgabe dieser Arbeit ist es, das Umfeldmodell des Zuges durch Software aufzubauen.

An dieser Stelle möchte ich bei allen bedanken, die mich in meiner Arbeit unterstützt haben und mir eine Hilfe bei Lösung von Problemen waren.

Besonderen Dank gilt Prof. Dr.-Ing. Jan Thomanek, der mir das Thema zur Verfügung gestellt und mich in allen Belangen unterstützt hat.

Inhalt

Vorwort	1
Inhalt	2
Abbildungsverzeichnis	4
Tabellenverzeichnis	9
Abkürzungsverzeichnis	11
1. Einleitung	12
1.1 Hintergrund	12
1.2 Motivation	13
2. Autonome Systeme für die Schiene	14
2.1 Sensor	15
2.2 V2X	16
2.3 Datenakquise / Fusion	17
2.4 Umfeldmodell	18
2.5 Aktor	20
3. Das ATLAS-System	21
3.1 Ziele und Anforderung	21
3.2 Systemarchitektur	22
3.3 Fahrzeuge	23
4. Spezifikation eines Umfeldmodells	24
4.1 Grundsätzlicher Struktur	24
4.2 Beschreibung des Matrixelement	25
4.2.1 Möglichkeit 1	26
4.2.2 Möglichkeit 2	27
4.2.3 Vergleich der zwei Möglichkeiten	28
4.3 Beispielumgebung	29
4.4 Import und Aufbau des Gleisplans	30
4.5 Längenangepasbare Gleismodule	31
4.6 Visualisierung des Umfeldmodells	33
4.7 Programmierung und Programmumgebung	34

4.7.1	Integrierte Entwicklungsumgebung.....	34
4.7.2	Modulare Programmierung	35
4.7.3	Flussdiagramm.....	38
4.7.4	CSV-,CFG-Datei einlesen und Datenbearbeitung.....	38
4.7.5	Matrix bauen und Visualisierung.....	40
4.8	Effiziente Datenspeicherung.....	43
4.8.1	Zweidimensionaler Array	43
4.8.2	Verkettete Liste.....	44
4.9	Lokalisierung der eigenen Position	46
5.	Automatische Generierung von CSV-Dateien	48
6.	Zusammenfassung und Ausblick	50
6.1	Zusammenfassung.....	50
6.2	Ausblick.....	51
	Literatur.....	52
	Anhang.....	54

Abbildungsverzeichnis

Abbildung 1: Prozesskette [5]	14
Abbildung 2: Infrarot-Empfänger unter dem Fahrzeug	15
Abbildung 3: Zusammensetzung von V2X [8]	16
Abbildung 4: Funktion der Sensordatenfusion	17
Abbildung 5: Umfeldmodell und Gleissystem	18
Abbildung 6: Weichen	20
Abbildung 7: Motor in der Weiche	20
Abbildung 8: IR-Sender am Fahrzeug	20
Abbildung 9: Systemarchitektur	22
Abbildung 10: Aufbau der Fahrzeuge	23
Abbildung 11: Zweidimensionales kartesisches Koordinatensystem	25
Abbildung 12: Zusammensetzung der Matrix-Elemente 1	26
Abbildung 13: Zusammensetzung der Matrix-Elemente 2	27
Abbildung 14: Koordinatentabelle	27
Abbildung 15: die Draufsicht des Korridors [14]	29
Abbildung 16: CSV-Datei in Excel	30
Abbildung 17: CSV-Datei in ASCII-Editor	31
Abbildung 18: Die Breite des Schienenmoduls	31
Abbildung 19: Die Breite des Schienenmoduls	32

Abbildung 20: Die Form des Moduls K333830.....	33
Abbildung 21: Visual Studio-Benutzerschnittstelle	34
Abbildung 22: Header-Dateien.....	35
Abbildung 23: Hauptfunktion	36
Abbildung 24: Programmablaufplan für das gesamte Projekt	38
Abbildung 25: Programmablaufplan für das Einlesen der CFG-Datei	39
Abbildung 26: Aufbau der Matrix.....	40
Abbildung 27: CFG-Datei des Beispiels.....	41
Abbildung 28: Zusammensetzung der Klasse 221150	41
Abbildung 29: CSV-Datei des Beispiels	41
Abbildung 30: Klasse 221150 mit 4 Ausrichtungen	42
Abbildung 31: Visualisierung vom Beispiel.....	42
Abbildung 32: Einfaches geschlossenes Schienensystem	43
Abbildung 33: die verkettete Liste drucken	45
Abbildung 34: Umfeldmodelle im Gleissystem	47
Abbildung 35: CSV-Datei und ihr Problem	48
Abbildung 36: Grundidee	49
Abbildung 37: Gleismodul [111053 und 111063]	55
Abbildung 38: Gleismodul [111053 und 111063]	55
Abbildung 39: Beispiel von CFG-Dateien.....	55
Abbildung 40: Gleismodul [111050 und 111060]	56
Abbildung 41: Aufbau [111050 und 111060]	56

Abbildung 42: Gleismodul [111051 und 111061]	56
Abbildung 43: Aufbau [111051 und 111061]	56
Abbildung 44: Gleismodul [111052 und 111062]	57
Abbildung 45: Aufbau [111052 und 111062]	57
Abbildung 46: Gleismodul [221150 & 221160]	58
Abbildung 47: Aufbau [221150 & 221160]	58
Abbildung 48: Gleismodul [221151 & 221161]	59
Abbildung 49: Aufbau [221151 & 221161]	59
Abbildung 50: Gleismodul [221130]	60
Abbildung 51: Aufbau [221130]	60
Abbildung 52: Gleismodul [221131]	60
Abbildung 53: Aufbau [221131]	60
Abbildung 54: Gleismodul [222230]	61
Abbildung 55: Aufbau [222230]	61
Abbildung 56: Gleismodul [222231]	61
Abbildung 57: Aufbau [222231]	61
Abbildung 58: Gleismodul [222430]	62
Abbildung 59: Aufbau [222430]	62
Abbildung 60: Gleismodul [331230]	63
Abbildung 61: Aufbau [331230]	63
Abbildung 62: Gleismodul [331231]	63
Abbildung 63: Aufbau [331231]	63

Abbildung 64: Gleismodul [331430]	65
Abbildung 65: Aufbau [331430]	65
Abbildung 66: Gleismodul [331431]	65
Abbildung 67: Aufbau [331431]	65
Abbildung 68: Gleismodul [332330]	67
Abbildung 69: Aufbau [332330]	67
Abbildung 70: Gleismodul [332331]	67
Abbildung 71: Aufbau [332331]	67
Abbildung 72: Gleismodul [333830]	69
Abbildung 73: Aufbau [333830]	69
Abbildung 74: Gleismodul [110000]	70
Abbildung 75: Gleismodul [110030]	70
Abbildung 76: Gleismodul [110010]	70
Abbildung 77: Ausrichtung 1	71
Abbildung 78: Ausrichtung 2	71
Abbildung 79: Ausrichtung 5	71
Abbildung 80: Ausrichtung 6	71
Abbildung 81: Ausrichtung 7	71
Abbildung 82: Ausrichtung 8	71
Abbildung 83: Ausrichtung 3	71
Abbildung 84: Ausrichtung 4	71
Abbildung 85: Gleismodul [221120]	72

Abbildung 86: Gleismodul [221121]	73
Abbildung 87: Geschlossenes Gleissystem	74
Abbildung 88: CFG-Datei vom Geschlossenen Gleissystem	76
Abbildung 89: Lineares System	76
Abbildung 90: CFG-Datei vom Linearen System	77

Tabellenverzeichnis

Tabelle 1: die Größe des Speichers mit unterschiedlichen Auflösungen	25
Tabelle 2: Nummerierung der anderen Züge	26
Tabelle 3: Verschiedene Matrix-Elemente (1)	26
Tabelle 4: Verschiedene Matrix-Elemente (2)	28
Tabelle 5: Aufbau von CSV	30
Tabelle 6: Die Funktionen der Header- und C-Dateien	35
Tabelle 7: Funktionen in Header-Datei	37
Tabelle 8: Die Bedeutung von jedem Element (1)	55
Tabelle 9: Die Bedeutung von jedem Element (2)	56
Tabelle 10: Die Bedeutung von jedem Element (3)	56
Tabelle 11: Die Bedeutung von jedem Element (4)	57
Tabelle 12: Die Bedeutung von jedem Element (5)	58
Tabelle 13: Die Bedeutung von jedem Element (6)	59
Tabelle 14: Die Bedeutung von jedem Element (7)	60
Tabelle 15: Die Bedeutung von jedem Element (8)	61
Tabelle 16: Die Bedeutung von jedem Element (9)	62
Tabelle 17: Die Bedeutung von jedem Element (10)	64
Tabelle 18: Die Bedeutung von jedem Element (11)	66
Tabelle 19: Die Bedeutung von jedem Element (12)	68

Tabelle 20: Die Bedeutung von jedem Element (13)	69
Tabelle 21: Abstand [110000]	70
Tabelle 22: Abstand [110030]	70
Tabelle 23: Abstand [110080] [20]	72
Tabelle 24: Abstand [221120]	72
Tabelle 25: Abstand [221121]	73
Tabelle 26: CSV-Datei vom Geschlossenen Gleissystem	75
Tabelle 27: CSV-Datei vom Linearen System	77

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
ATLAS	Asynchroner Transport-, Logistik- und Automatisierungsmodus auf der Schiene
BMP	Bitmap Image
CFG	Konfiguration
CSV	Character-separated values
Ego	Position des eigenen Fahrzeugs
Fzg	Fahrzeug
ID	Identification
IDE	integrated development environment
IR	Infrarot
LKW	Lastkraftwagen
SPI	Serial Peripheral Interface
V2I	Fahrzeug-zu-Infrastruktur
V2N	Fahrzeug-zu-Netzwerk
V2P	Fahrzeug-zu-Fußgängern
VS	Visual Studio
V2V	Fahrzeug-zu-Fahrzeug
V2X	Fahrzeug-zu-Alles
WLAN	Wireless Local Area Network

1. Einleitung

1.1 Hintergrund

Die Eisenbahn ist ein schienengebundenes Verkehrssystem für den Transport von Gütern und Personen. Sie hat schon eine lange Geschichte [2]. Mit der ersten industriellen Revolution und der Verbesserung der Dampfmaschine wurde die Eisenbahn Realität. Seit ihrem Aufkommen veränderte sich das Verständnis der Menschen von Distanz. Gleichzeitig verändert die Lebensbedingungen der Menschen. Und die Eisenbahn ist auch für die nächsten 200 Jahre ein wichtiges Transportmittel geworden. Die Eisenbahnindustrie wächst mit Bevölkerungswachstum, Verstädterung, mehr Transportbedarf und Umweltproblemen.

Im modernen Verkehr stellt der Straßentransport einen großen Anteil dar. Im Vergleich zum Bahntransport ist der Straßentransport flexibler. Im Kurzstreckentransport sind die Straßentransportkosten niedriger. Aber wenn die LKWs für den Ferntransport genutzt werden, wird auf einige Probleme stoßen. Einerseits wird es mehr Ressourcen verbraucht und die Umwelt verschmutzt. Andererseits kann die plötzliche Situation während des Transports zu einer Beschädigung der Waren führen. Deswegen ist der Fernverkehr mit dem Zug besser [3].

Der Schienenverkehr hat viele Vorteile, wie zum Beispiel die Sicherheit, die Umweltfreundlichkeit und das große Ladungsvolumen. Aber das gesamte Transportvolumen von Straßentransport ist höher als das gesamte Transportvolumen von Eisenbahnen. Denn der Schienenverkehr kann jedoch nicht den gesamten Transportprozess abschließen. In den meisten Fällen müssen andere Verkehrsträger kombiniert werden. Das Be- und Entladen von Zügen hängt oft von den Lastwagen ab. Dies bedeutet, dass die Flexibilität des Schienenverkehrs gering ist und es schwierig ist, Punkt-zu-Punkt-Transporte durchzuführen.

In Zukunft wird das automatische Fahren der Eisenbahn der Schlüssel zum Transportsystem sein, was viele Vorteile mit sich bringen kann. Das automatisierte Fahren der Eisenbahn hilft, die Arbeitskosten zu senken. Dies bedeutet, dass die Arbeit der Menschen durch künstliche Intelligenzsysteme und Computer ersetzt wird. Das Fahrzeug wird kontinuierlich im System betrieben, und die Effizienz des Frachttransports kann verbessert werden.

Gleichzeitig können Bordcomputer Datenverbindungen mit der Infrastruktur herstellen, um das Ausnutzen der Infrastruktur zu verbessern. Die Auslegung des Gleissystems wird auch beim Bau neuer Infrastrukturen umfassend berücksichtigt.

Im Fernverkehr ersetzt der autonome systembasierte Zugverkehr die Lastkraftwagen als Hauptverkehrsmittel. Diese Art von Transport kann nicht nur die Umwelt, sondern auch die Wirtschaft verbessern und eine nachhaltige Entwicklung fördern.

1.2 Motivation

Mit der Weiterentwicklung von Industrie 4.0 muss ein umfassendes, vollständiges und intelligentes Schienenverkehrssystem entwickelt werden. Sie muss sich in Richtung der Automatisierung und dezentralem Transport bewegen. Um diese Ziele zu erreichen, wurde das ATLAS-Projekt initiiert. Das ATLAS-Projekt (Asynchroner Transport-, Logistik- und Automatisierungsmodus auf der Schiene) befasst sich mit der möglichen zukünftigen Nutzung der Infrastruktur Schiene [4].

Das gesamte Atlas-Projekt wird in Form eines Modells erstellt, das aus mehreren Teilen besteht, darunter der Bau und die Optimierung des Schienenmoduls, die Einrichtung der Datenbank, die Einstellung von Balisen und Weichen, die Einrichtung des Umfeldmodells und die Lokalisierung des Zuges usw. Jeder Teil wird von einem oder mehreren Studenten durchgeführt.

Es widmet sich der Lösung der Probleme, die bei heutigen Eisenbahnsystemen auftreten, wie z. B. die hohen Kosten für menschliche Arbeit, die lange Zeit für die Umleitung und Montage von Zügen usw. sowie der Einsatz von Informationstechnologie zur Verbesserung der Systemeffizienz und -flexibilität.

Das autonom fahrende Fahrzeug auf der Schiene benötigt genaue Information über die Gleispläne und die eigene Position. Diese Information kann der Zug durch Erstellung eines Umfeldmodells erhalten. Deshalb ist das Ziel der Arbeit:

- Modellierung des Gleissystem als Umfeldmodell.
- Konstruktion anhand der Datenbank des Gleissystems.
- Visualisierung, in Form einer BMP-Datei.

2. Autonome Systeme für die Schiene

Ein autonomes System kann auch als automatisches Fahrsystem bezeichnet werden. Es ist ein komplexes Projekt mit vielen Technologien. Das autonome System ist unabhängig und benötigt keine externe Steuerung. Und das automatische Fahren eines Zuges ist nicht so kompliziert wie das automatische Fahren eines Wagens. Weil der Zug im vorgeschriebenen Eisenbahnsystem fährt, gibt es nicht viele unerwartete Situationen, so dass die Anzahl der Sensoren und Aktoren beim Aufbau des autonomen Systems reduziert werden kann, wodurch die Kosten gespart werden.

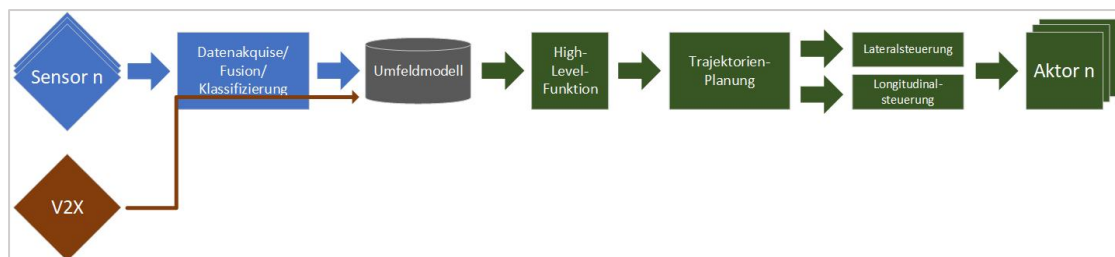


Abbildung 1: Prozesskette [5]

Abbildung 1 zeigt die Prozesskette eines allgemeinen autonomen Systems. Sie erklärt, wie das gesamte System Sensoren, V2X-Signale verarbeitet und wie Aktoren gesteuert wird. Die Signale des V2X und der Sensoren durchlaufen diesen Prozess und realisieren schließlich die Ansteuerung der Aktoren. Jeder Schritt wird in den folgenden Abschnitten kurz beschrieben.

2.1 Sensor

- **Infrarot-Empfänger:**

- ① An der Vorder- und Rückseite des Zugs befinden sich zwei Infrarotempfänger. Während der Zug Balise passiert, empfängt der IR-Empfänger Infrarotsignale von Balise, um die Geschwindigkeit und die Position vom Zug zu bestimmen.

Die Balisen, die an den Schwellen befestigt sind, bestehen aus Infrarot-Sender und Leiterplatte. Sie werden zum Speichern von eindeutige ID jeder Balise und zum Bestätigen von Zugstandorten verwendet.

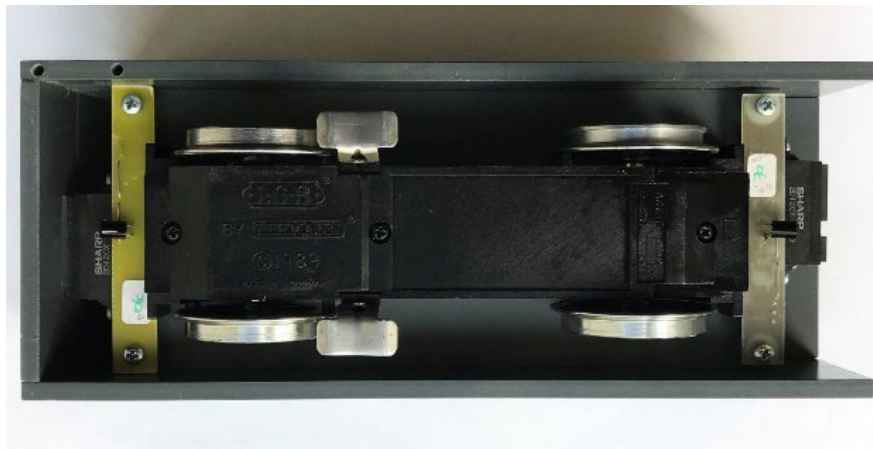


Abbildung 2: Infrarot-Empfänger unter dem Fahrzeug

- ② Jede Weiche enthält einen Infrarotempfänger. Diese IR-Empfänger empfangen das Schienenwechselfsignal vom Zug.

- **Kamera:**

Die Kameras sind neben den Schienen montiert, um den Betrieb des Fahrzeugs in Echtzeit zu überwachen. Das Bild wird auf den Monitor im Überwachungsraum übertragen. Gleichzeitig kann die Kamera in das Fahrzeug eingebaut werden. Es liefert ein vollständiges und natürliches Bild jeder Fahrsituation [6].

2.2 V2X

V2X ist der Oberbegriff für Verkehrsvernetzung, die elektronische Kommunikation der Verkehrsteilnehmer untereinander in den Formen Fahrzeug-zu-Fahrzeug(V2V), Fahrzeug-zu-Infrastruktur(V2I), Fahrzeug-zu-Netzwerk(V2N) und Fahrzeug-zu-Fußgängern(V2P) [7].

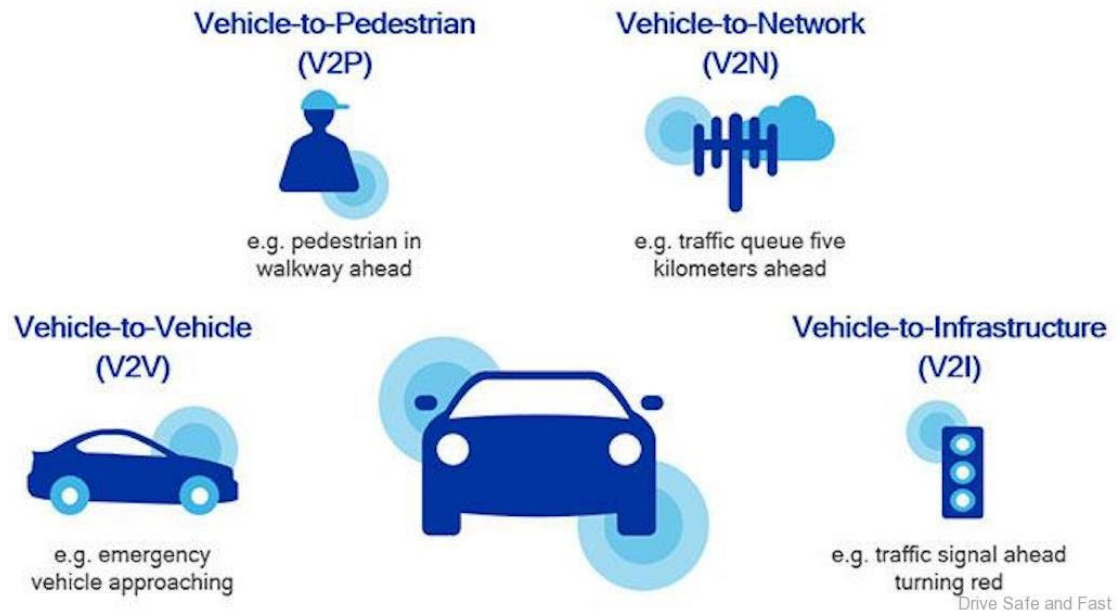


Abbildung 3: Zusammensetzung von V2X [8]

Die V2X-Technologie kann auch im ATLAS-Projekt verwendet werden. Züge können Informationen andere Züge oder Autos und verschiedene Signalzustände enthalten. Durch den Einsatz dieser Technologie können die Unfälle effektiv vermieden werden. So soll es durch die Kommunikation mit anderen Fahrzeugen möglich sein, deren Position und Fahrtrichtung zu ermitteln.

2.3 Datenakquise / Fusion

Wie in Abbildung 4 dargestellt, werden die von den Sensoren gewonnenen Daten mit Datenfusion verarbeitet. Die verarbeiteten Daten werden in das Umfeldmodell importiert.

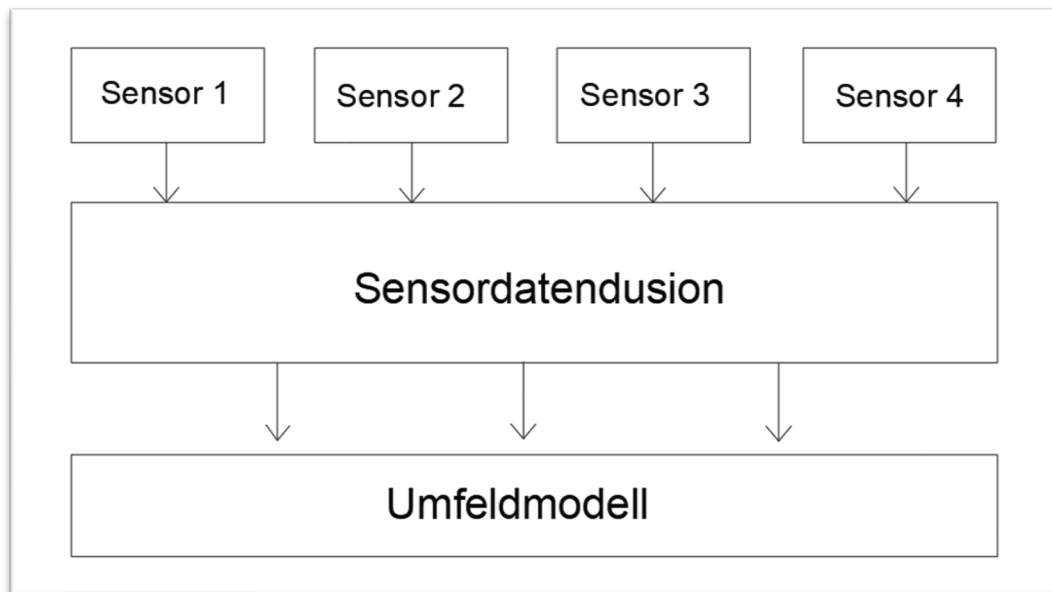


Abbildung 4: Funktion der Sensordatenfusion

Das Hauptziel der Datenfusion ist die Verarbeitung und Integration von Daten aus der Umgebung über Sensoren. Durch die Datenfusion werden Daten vollständiger, sicherer, einheitlicher und genauer [9].

Durch den Einsatz mehrerer Sensoren können Datenunvollständigkeiten umgangen werden. Die Datenfusion kann Mehrdeutigkeiten zwischen mehreren Daten beseitigen und die Daten können einheitlicher und sicherer sein.

2.4 Umfeldmodell

Die Abbildung 5 stellt die Architektur des ATLAS-Systems dar. Das Konzept der Systemarchitektur wird in Kapitel 3 erwähnt. Das Umfeldmodell basiert auf dem Fahrzeugmodul. Aus der Abbildung ist ersichtlich, dass sich in einem ATLAS-System mehrere Züge befinden können und jeder Zug ein Umfeldmodell hat. Es wird verwendet, um den Standort des Zuges sowie die Standortinformationen anderer Züge zu bestimmen und zu speichern.

Im Bild gibt es das rote Wort „Ego“. Ego ist lateinisch und steht für die eigene Position, die den Standort des Zuges in seinem Umfeldmodell darstellt. Das grüne Wort "Fzg." kennzeichnet den Positionen anderer Züge. In Wirklichkeit werden diese Koordinateninformationen nicht durch Bilder und farbige Wörter dargestellt, sondern durch Matrix und Daten.

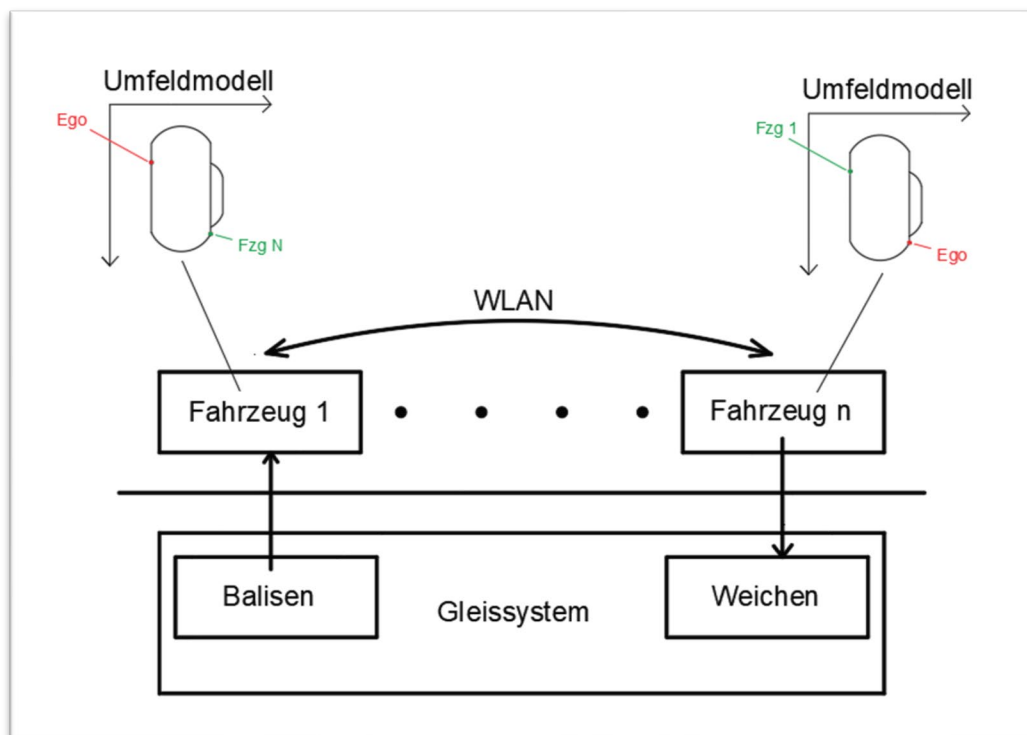


Abbildung 5: Umfeldmodell und Gleissystem

Die Merkmale von Umfeldmodell sind unabhängig und miteinander verbunden. Die Umfeldmodelle nutzen WLAN, um Informationen auszutauschen. Die Rolle des Umfeldmodells besteht darin, Zugpositionsinformationen zu erhalten. Der Mikrocomputer im Fahrzeug analysiert diese Informationen, um Zugkollisionen zu vermeiden, sodass das Umfeldmodell ein wichtiger Bestandteil des autonomen Systems ist.

Das Umfeldmodell repräsentiert die Umgebung des Fahrzeugs. Auf Basis der Daten im Umfeldmodell wird die aktuelle Szene interpretiert und daraus folgend über Pfadplanung, Lateral- und Longitudinal-steuerung die Aktoren angesteuert.

- ① Szenenanalyse kann die Bewegung einzelner Züge im Schienensystem analysieren und simulieren. Das Modul „Szenenanalyse“ erkennt die aktuelle Situation anhand der Fahrdaten des Ego-Fzg und bestimmt daraus das zu fahrende Manöver, wie z.B. überholen.
- ② Nach Abschluss der Szenenanalyse kann die am besten geeignete Pfadplanung für das Fahrzeug eingerichtet werden. Der Block „Pfadplanung“ ermittelt für das aktuelle Manöver den entsprechenden Fahrpfad (Trajektorie).
- ③ Laut Pfadplanung kann die Lateral- und Longitudinal-steuerung, d.h. die Quer- und Vertikalbewegung des Fahrzeugs verändern. Schließlich wird das Signal an Aktor übertragen, damit das Fahrzeug auf den richtigen Pfad fährt.

2.5 Aktor

- **Motoren:**

Der Zug wird von einem Motor angetrieben. Und der Motor des Fahrzeugs wird direkt angesteuert.

Die Weiche besteht aus einem Motor. Wenn die weiche das Signal empfängt, startet der Motor.

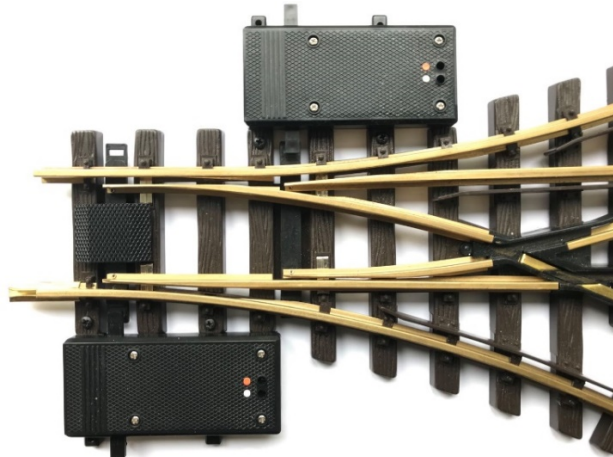


Abbildung 6: Weichen

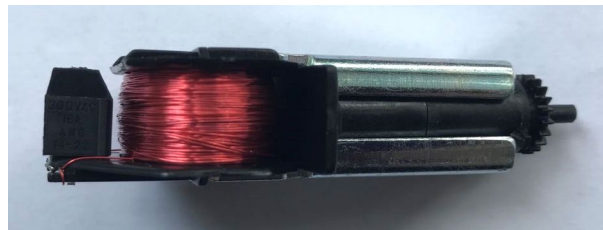


Abbildung 7: Motor in der Weiche

- **Infrarot-Sender für Weichenstellung:**

Die IR-Sender sind an der Vorder- und Rückseite des Fahrzeugs montiert und kann zur Weiche-steuerung durch Aussenden von Infrarotsignalen verwendet werden.

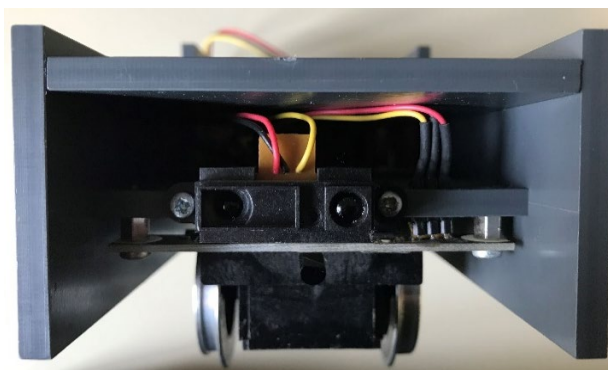


Abbildung 8: IR-Sender am Fahrzeug

3. Das ATLAS-System

3.1 Ziele und Anforderung

Das ATLAS-System beschäftigt sich mit der Verbesserung des Schienensystems. Viele Güter werden heutzutage auf der Schiene transportiert. Eisenbahntransport wird aufgrund seiner großen Ladekapazität und seiner guten Sicherheit häufig von Menschen genutzt. Aufgrund seiner Inflexibilität und hohen Wartungskosten spiegelt es seine Vorteile gegenüber dem Straßentransport nicht wider. Die Ziele des ATLAS-Projekts sind folgende: Flexibler und effizienter Güterverkehr, Hochautomatisiertes Schienensystem [10].

Das hochautomatisierte Eisenbahnsystem kann auch als autonomes System oder autonomes Fahren verstanden werden. Sein Konzept wird schon in Kapitel 2 erwähnt. Was hinzugefügt werden muss, ist Autonomiestufen. In Europa Bundesanstalt für Straßenwesen und den USA wird die Klassifizierung des autonomen Fahrens in sechs Stufen vorgenommen: Autonomiestufe 0 bis Autonomiestufe 5. Autonomiestufe 5 ist die höchste Stufe und steht für Vollautomatisierung. Das System ist kein menschliches Eingreifen erforderlich. Man muss nur die Ziele festlegen und das System starten. Ziel des ATLAS-Projekts ist es, ein automatisiertes, intelligentes und unbemanntes Frachtsystem der Autonomiestufe 5 zu schaffen [11].

Das ATLAS-System ist jetzt ein Projekt des Labors. Es ist ein Modell zum Testen des automatischen Fahrens von Fahrzeugen im Gleissystem. Um dieses Modell zu konstruieren, müssen Studenten zusammenarbeiten. Das Projekt umfasst den Bau von Schienenmodulen, den Bau von Datenbanken, die Installation von Infrarotmodulen und den Entwurf von Fahrzeugmodulen usw.

3.2 Systemarchitektur

Die Abbildung 9 unten zeigt die Systemarchitektur. Dieses Bild ist eine Vereinfachung von Abbildung 5. Das ATLAS-System besteht aus Zügen, Balisen und Weichen.

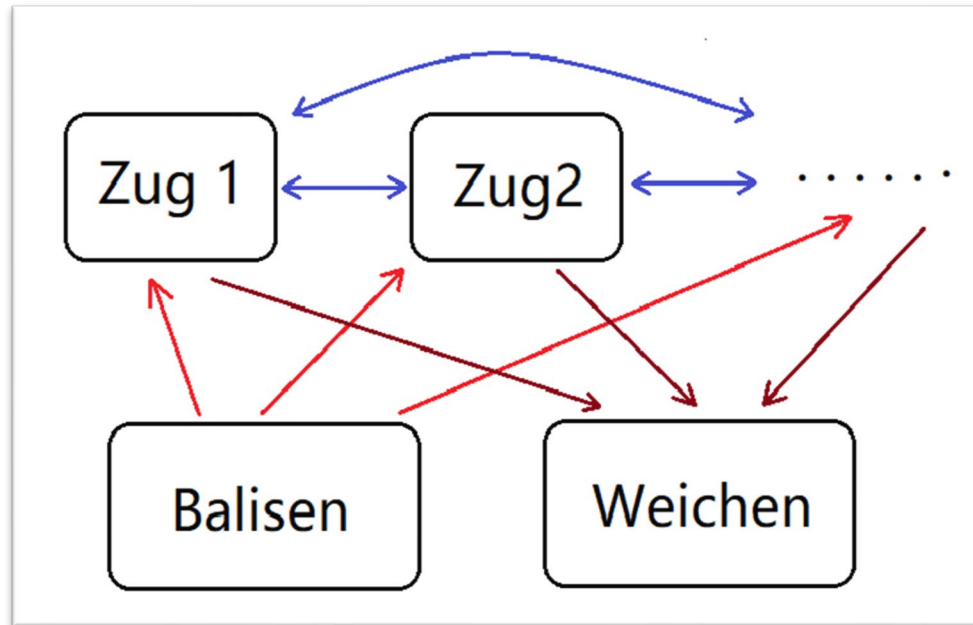


Abbildung 9: Systemarchitektur

Im Eisenbahnsystem verkehren mehrere Züge. Um die Konflikte zwischen Zügen zu vermeiden, müssen die Züge auch Informationen austauschen können. Der Informationsaustausch zwischen den Zügen erfolgt durch WLAN und wird durch die blaue Pfeillinien dargestellt. Die Auslassungspunkte stellen die anderen Züge dar. Je mehr Züge vorhanden sind, desto höher sind die Anforderungen an das autonome System.

Infrarot wird durch die rote Pfeillinien dargestellt. Es wird von Balisen gesandt und von dem darüberfahrenden Zug empfangen. So kann das System den Standort des Zuges ermitteln. Die Weiche besteht aus einem Motor und ist mit einem Infrarotempfänger verbunden. Der Zug sendet ein Infrarotsignal zur Steuerung der Weiche. Die dunkelrote Pfeillinien repräsentieren das Infrarotsignal.

Natürlich kann dieses System erweitert werden. Neben den in der Abbildung 9 gezeigten Geräten kann auch andere Hardware installiert werden, z.B. Kamera, Geschwindigkeitsmessgerät usw. Ein weiteres wichtiges Ziel ist die kontinuierliche Verbesserung und Erweiterung des ATLAS-Systems, um es vollständiger und sicherer zu machen.

3.3 Fahrzeuge

Die Abbildung 10 zeigt grob den Aufbau des Fahrzeugs. Fahrzeug besteht aus zwei Ebenen, High Level und Low Level. High Level ist eine KI-Plattform mit Funktionen für Pfadplanung und Umgebungsmodule. Low Level ermöglicht die Motorsteuerung und die Identifizierung des Balisen. Zwei Ebenen erreichen die Informationsübertragung über SPI.

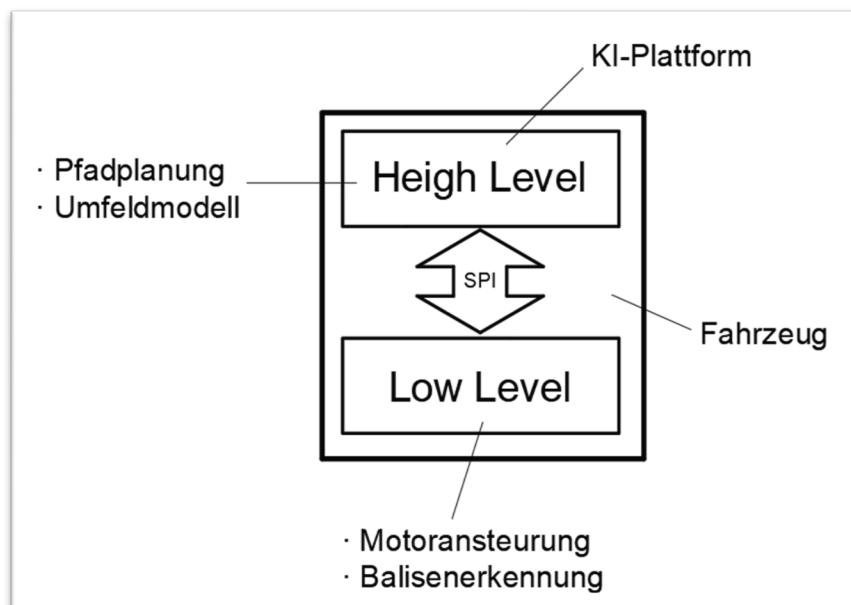


Abbildung 10: Aufbau der Fahrzeuge

High Level ist eine KI-Plattform, die zum Beispiel auf „Nvidia Jetson nano“ aufgebaut ist und die Daten verarbeiten und eigene Lösungen generieren kann. Sie integriert Pfadplanung und Umweltmodelle. Low Level enthält Hardwaremodule. Sie ermöglicht Motorsteuerung, Balisenerkennung und Weichensteuerung.

Zwischen High Level und Low Level erfolgt die Informationsübertragung über SPI. SPI ist die Abkürzung für Serial Peripheral Interface. Es ist ein synchroner Hochgeschwindigkeits-Vollduplex-Kommunikationsbus.

Vorteile von SPI:

- Unterstützung für Vollduplex-Kommunikation
- Einfache Kommunikation
- Schnelle Datenübertragung [12].

4. Spezifikation eines Umfeldmodells

Umfeldmodell ist im ATLAS-Projekt ein Container zum Speichern von Schienensystemen oder zum Erfassen des Zustands anderer Fahrzeuge. Jedes Objekt wird durch einen Zustandsvektor mit den folgenden Elementen dargestellt: Position in Fahrzeugkoordinaten, Geschwindigkeit in Fahrzeugkoordinaten, Länge und Breite. Es gibt ein Umfeldmodell für jedes Fahrzeug im Gleissystem, um die Änderungen jedes Fahrzeugs widerzuspiegeln [13].

Matrix- und zweidimensionale kartesische Koordinaten sind ein wichtiger Bestandteil des Umfeldmodells. Jedes Element der Matrix entspricht einer Zelle des kartesischen Koordinatensystems. Ein Schienensystem kann durch Setzen der Werte der Elemente in der Matrix konstruiert werden, dh. die Matrix entspricht in etwa dem zweidimensionalen kartesischen Koordinatensystem.

4.1 Grundsätzlicher Struktur

Der Zug bewegt sich nicht in vertikaler Richtung, so dass das ATLAS-System sich in einer zweidimensionalen Ebene befindet. Es kann durch ein kartesisches Koordinatensystem dargestellt werden.

Das Matrix-Array wird zum Speichern der Eisenbahndaten verwendet. Im Programm kann eine Matrix wie z. B. „char Matrix [x] [y];“ festgelegt werden, wobei „char“ den Matrixelementtyp darstellt und ein Matrixelement 8 Bits belegt. Die Anzahl „n“ der Elemente in der Matrix wird bestimmt durch x, y:

$$n = x \times y .$$

Die Größe von x, y wird durch die Größe des Eisenbahnsystems bestimmt. Je komplizierter das Eisenbahnsystem ist, desto größer ist der Wert von x, y.

Wie in der Abbildung 11 gezeigt, kann ein zweidimensionales kartesisches Koordinatensystem mit der x-Achse nach rechts und der y-Achse nach unten erstellt werden.

Der obere linke Punkt in den Koordinaten ist der Koordinatenursprung. Er entspricht dem linken oberen Punkt des realen Bereichs, in dem sich das Gleissystem befindet. Der Ursprung der Koordinaten ist fest. Und es ist notwendig, seine Position in der realen Welt entsprechend der tatsächlichen Situation zu bestimmen.

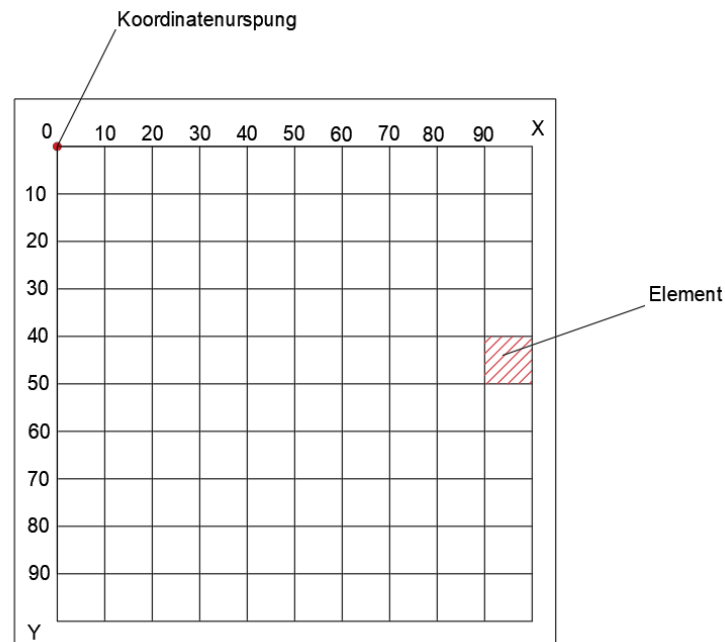


Abbildung 11: Zweidimensionales kartesisches Koordinatensystem

Der Maßstab des Koordinatensystems in der Abbildung beträgt 10 cm, dh die Auflösung jeder Zelle beträgt 10 cm. Da jede Zelle im Koordinatensystem jedem Element in der Matrix entspricht, beträgt die Größe einer Zelle im Speicher 8 Bits. In diesem Projekt wurden 100 mm / Zelle verwendet, um den Speicherplatz zu reduzieren. Und bei der Visualisierung des Gleissystems darf die erzeugte Bilddatei nicht zu groß sein. Um das Umfeldmodell genauer zu gestalten, kann man eine kleinere Auflösung einstellen, die Folge ist jedoch, dass mehr Speicherplatz belegt wird. Die folgende Tabelle beschreibt die Größe des Speichers, die bei unterschiedlichen Auflösungen benötigt wird:

1 Zelle $\hat{=}$ 8 Bits	max. X = 20 m		max. Y = 40 m
Auflösung:	1 mm/Zelle	10 mm/Zelle	100 mm/Zelle
Speicher:	5,96 Gigabits	7,63 Megabytes	80 Kilobytes

Tabelle 1: die Größe des Speichers mit unterschiedlichen Auflösungen

Die Größe des zweidimensionalen Koordinatensystems kann frei geändert werden. Um Speicherplatz zu sparen und den Programmablauf effizienter zu gestalten, muss das Koordinatensystem entsprechend der tatsächlichen Situation festgelegt werden.

4.2 Beschreibung des Matrixelement

Es gibt zwei Möglichkeiten, ein Matrixelement zu beschreiben. Diese beiden Möglichkeiten werden im Folgenden verglichen.

4.2.1 Möglichkeit 1

Ein Matrix-Element belegt 8 Bits mit unterschiedlichen Bedeutungen. Die Abbildung 12 beschreibt jedes Bit:

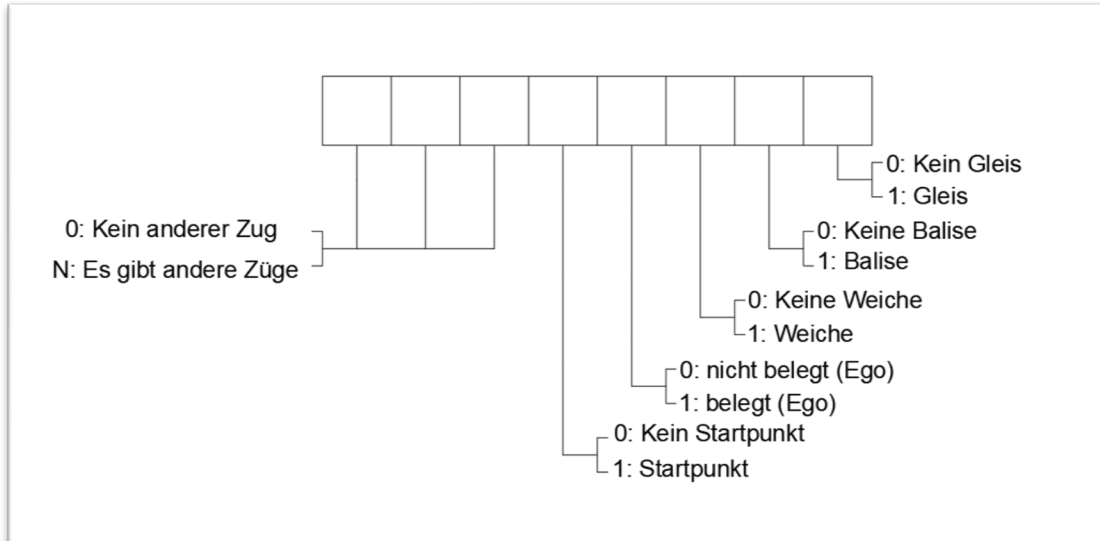


Abbildung 12: Zusammensetzung der Matrix-Elemente 1

N	001	010	011	100	101	110	111
Bedeutung	Zug 1	Zug 2	Zug 3	Zug 4	Zug 5	Zug 6	Zug 7

Tabelle 2: Nummerierung der anderen Züge

Tabelle 2 zeigt die verschiedenen Situationen. Wenn sich in Umfeldmodell andere Züge befinden, ändern sich die ersten drei Elemente des Elements. Es gibt sieben verschiedene Situationen, die unterschiedliche Züge darstellen.

Beispiele:

Binärzahl	Dezimalzahl	Bedeutung	Farbe in bmp
00000000	0	Stopp / Kein Gleis	Weiß
00000001	1	Gleis	Schwarz
00000011	3	Gleis mit Balise	Rot
00000101	5	Gleis mit Weiche	Gelb
00010001	17	Startpunkt eines Schienenmoduls	Grün
000x1xx1	9,11,13,25,27,29	Mit dem Zug belegt	Blau
Nx0xx1	Andere Nummern	Es gibt andere Züge	Andere Farben

Tabelle 3: Verschiedene Matrix-Elemente (1)

4.2.2 Möglichkeit 2

Die zweite Methode verwendet nur 4Bits. Die Koordinaten der Fahrzeuge werden in der Koordinatentabelle gespeichert.

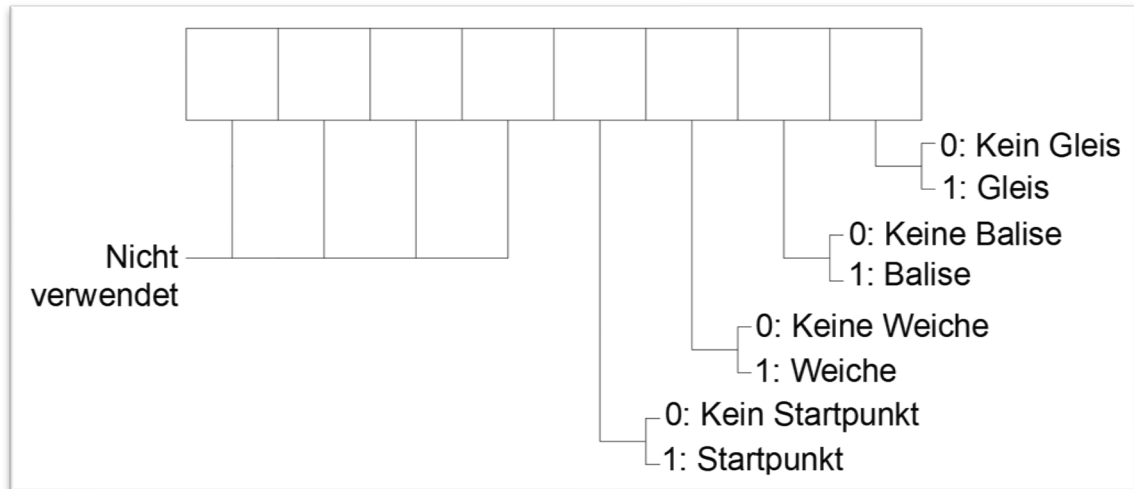


Abbildung 13: Zusammensetzung der Matrix-Elemente 2

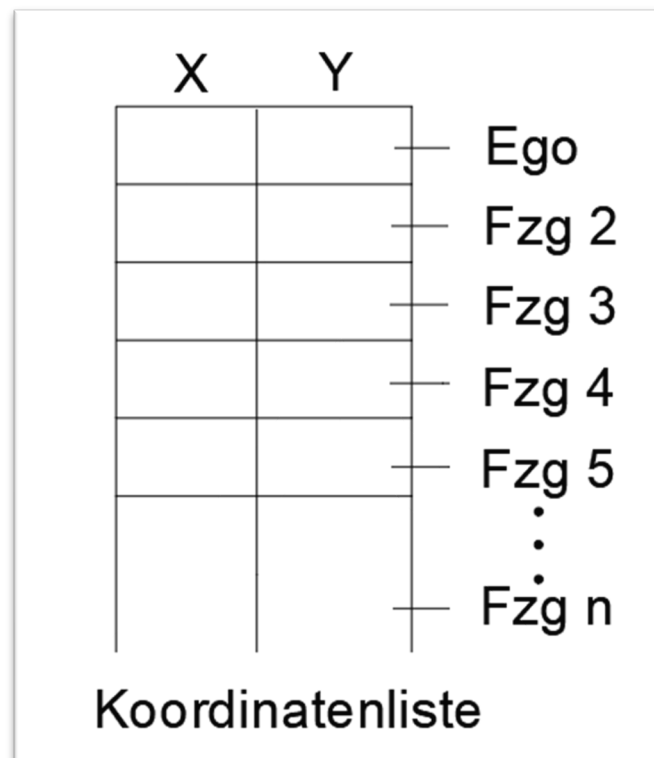


Abbildung 14: Koordinatentabelle

Die Koordinatenliste ist mit dem Umfeldmodell verknüpft (siehe Abbildung 34 für Details).

Beispiele:

Binärzahl	Dezimalzahl	Bedeutung	Farbe in bmp
00000000	0	Stopp / Kein Gleis	Weiß
00000001	1	Gleis	Schwarz
00000011	3	Gleis mit Balise	Rot
00000101	5	Gleis mit Weiche	Gelb
00001001	9	Startpunkt eines Schienenmoduls	Grün

Tabelle 4: Verschiedene Matrix-Elemente (2)

4.2.3 Vergleich der zwei Möglichkeiten

Die erste Möglichkeit besteht darin, alle Informationen im Element zu speichern. Dies vereinfacht die Programmierung. Bei dieser Methode treten einige Probleme auf: Wenn sich die Position des Fahrzeugs ändert, ändern sich die Daten im Element, und die gesamte Matrix muss neu generiert werden, was die Effizienz des Programms erheblich verringert. Und weil die Anzahl der Bits im Matrixelement festgelegt ist, gibt es eine Obergrenze für die Anzahl der Fahrzeuge im System.

Die zweite Möglichkeit verwendet nur die ersten 4 Bits des Elements, um die Daten der Schiene zu speichern. Diese Daten sind alle unverändert. Die Fahrzeugkoordinaten werden in einer neuen Tabelle gespeichert. Die Koordinatenliste und die Matrixelemente sind unabhängig. Wenn sich die Position des Fahrzeugs ändert, werden die Matrixelemente nicht geändert. Und das Programm muss die neue Matrix nicht neu generieren, sondern nur die Standortliste ändern. Die Größe der Liste kann frei geändert werden, deshalb ist die Anzahl der Fahrzeuge im System nicht begrenzt. Gegenüber der ersten Möglichkeit erhöht diese Möglichkeit die Effizienz des Programms.

Aufgrund des Vergleichs ist die zweite Möglichkeit geeigneter, so dass das Verfahren zum Aufbauen des Matrixelements in der zweiten Möglichkeit verwendet wird.

4.3 Beispielumgebung

Das Bild unten stellt die Draufsicht des Korridors dar. Wenn das Schienensystem auf dem Korridor aufgebaut ist, müssen Länge und Breite des Korridors berücksichtigt werden. Die Daten in der Figur sind in Millimetern aufgetragen und werden in einem Verhältnis von 1: 100 gezeichnet. Die obere Breite des Korridors beträgt 3350 mm, die untere Breite 2845 mm und die Länge 15385 mm. Daher wird der Schnittpunkt der linken Wand und der oberen Wand als Koordinatenursprung verwendet, das Maximum x wird auf 3400 und das Maximum y auf 15400 festgelegt. Bei der Anordnung des Eisenbahnmoduls muss man die Hindernisse im Korridor berücksichtigen, z. B. Türen, Heizung und Treppen.

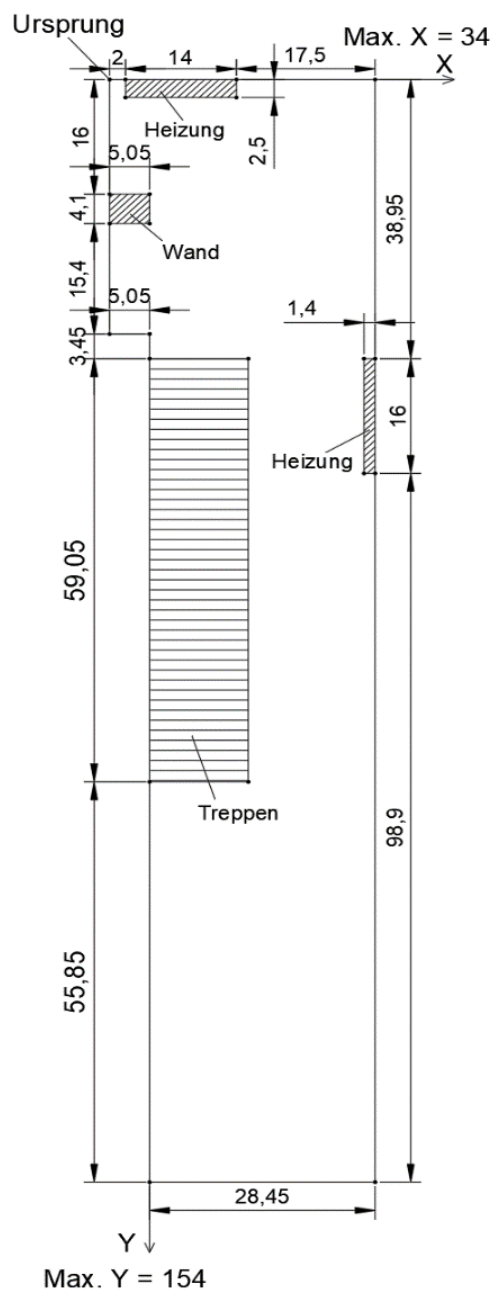


Abbildung 15: die Draufsicht des Korridors [14]

4.4 Import und Aufbau des Gleisplans

Die Abbildung 16 stellt eine CSV-Datei dar. Die Erstellung dieser Datei spielt eine wichtige Rolle bei der Generierung von Matrix. Das Dateiformat CSV steht für English Comma-separated values und beschreibt den Aufbau einer Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten [15].

In der CSV-Datei werden die verwendete Schienenmodule gespeichert. Die verwendeten Schienenmodule basieren auf dem Eisenbahnsystem. Vor dem Erstellen der CSV-Datei muss man das Schienensystem gemäß der tatsächlichen Situation erstellen und dann die Position und die Menge des zu verwendenden Moduls bestimmen.

	A	B	C	D	E
1	333830	100	310	4	1
2	111080	130	309	5	1
3	111080	163	229	1	1
4	111080	244	196	2	1
5	111080	325	229	6	1
6	221130	358	310	4	1
7	111080	130	490	7	1
8	111080	163	570	3	1

Abbildung 16: CSV-Datei in Excel

Nach dem Einlesen der CSV-Datei importiert das Programm die Daten, die dem Schienenmodul in CSV entsprechen, in die Matrix. Die Tabelle 5 zeigt den Aufbau von CSV.

Die erste Spalte repräsentiert die Nummer des verwendeten Schienenmoduls, die zweite Spalte ist die Abszisse des Startpunkts jedes Moduls und die dritte Spalte ist die Ordinate des Startpunkts jedes Moduls. Die vierte Spalte gibt die Ausrichtung des Moduls an. Die fünfte Spalte gibt die Modellnummer der Bretter an.

Nummer der Schienenmoduls	Abszisse des Startpunkts	Ordinate des Startpunkts	Ausrichtung des Moduls	Modellnummer der Bretter
333830	100	310	4	1
111080	130	309	5	1

Tabelle 5: Aufbau von CSV

Die Ausrichtung des Moduls bedeutet, dass die andere Klasse außer Klasse 111080 vier Ausrichtungen hat, die Nummer 1: von links nach rechts, die Nummer 2: von rechts nach links, die Nummer 3: von oben nach unten, die Nummer 4: von unten nach oben. Abbildung 20 zeigt die vier Ausrichtungen des K333830.

Die CSV-Datei kann mit EXCEL erstellt und dann als Datei mit der Dateinamenserweiterung ".csv" gespeichert werden. Sie kann auch mit einem beliebigen ASCII-Editor bearbeitet werden.

Wenn sie vom ASCII-Editor bearbeitet wird, muss jedes Element durch ein Komma getrennt werden.

```
333830,100,310,4,1
111080,130,309,5,1
111080,163,229,1,1
111080,244,196,2,1
111080,325,229,6,1
221130,358,310,4,1
111080,130,490,7,1
111080,163,570,3,1
```

Abbildung 17: CSV-Datei in ASCII-Editor

4.5 Längen Anpassbare Gleismodule

Eine Konfigurationsdatei ist eine Textdatei auf einem Computer, in der bestimmte Einstellungen von Computerprogrammen oder Hardwarebestandteilen gespeichert sind [16]. Weil einige Schienenmodule nicht gemessen werden oder einige Module Datenfehler aufweisen, kann man das Schienenmodul mithilfe der CFG-Datei anpassen.

Die Abbildung 18 zeigt, dass das Programm diese Module durch Einlesen der CFG- und CSV-Dateien aufruft, um die Matrix zu erstellen. Anschließend wird die Matrix in das Umfeldmodell importiert.



Abbildung 18: Die Breite des Schienenmoduls

Im Wesentlichen unterscheidet sich die CFG-Datei nicht von einer TXT-Datei. Die CFG-Datei kann auch mit einem ASCII-Editor erstellt werden. Das Datenschreibformat basiert auf dem Programm. Jedes Element in diesem Projekt ist durch ein Komma getrennt und endet mit einem Punkt.

Die Breite des Schienenmoduls beträgt 150 mm, geteilt durch 10 mm in 15 gleiche Teile, und das gesamte Eisentopfmodul wird durch die mittleren 10 mm dargestellt. Die Breite des Moduls sollte beim Aufbau des Schienensystems in der Matrix berücksichtigt werden.



Abbildung 19: Die Breite des Schienenmoduls

Im Anhang werden nicht nur alle anpassbaren Module, sondern auch alle Module mit einer bestimmten Länge aufgelistet.

4.6 Visualisierung des Umfeldmodells

Die Visualisierung ist während des gesamten Projekts nicht zwingend notwendig. Autonome Systeme benötigen nur die Daten der zweidimensionalen Matrix. Aber bei der Visualisierung kann die Daten in einem zweidimensionalen Array als Bild auf einem Computer angezeigt werden. Und sie kann intuitiv den Positionsfehler widerspiegeln, der beim Aufbau des Gleissystems auftritt.

Die folgende Abbildung ist ein Beispiel. Es zeigt die Form des Moduls K333830 mit 4 Ausrichtungen.

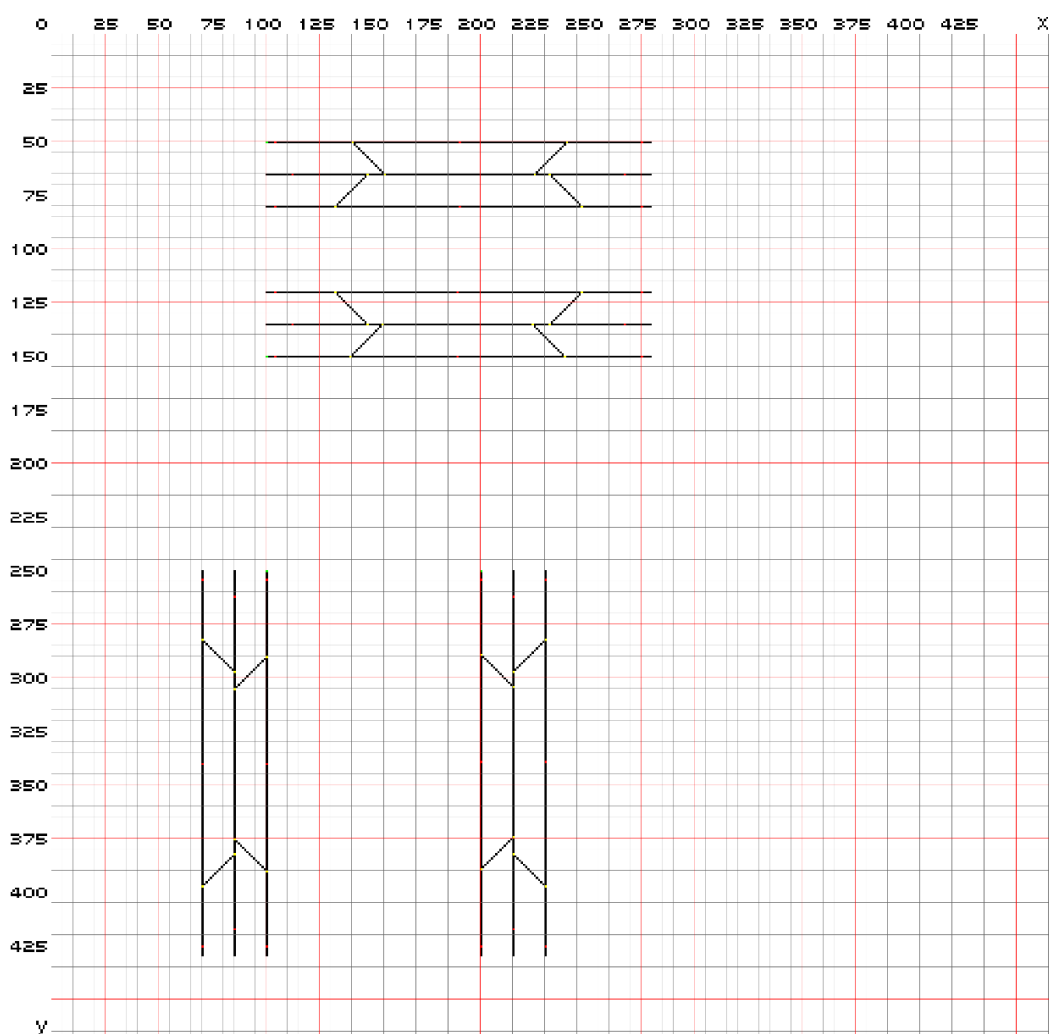


Abbildung 20: Die Form des Moduls K333830

Die Genauigkeit des Koordinatensystems beträgt 5 cm. Die x-Achsen des Koordinatensystems ist nach links und die y-Achsen ist nach unten gerichtet. Die Bildgröße kann durch Ändern von BMP_WIDTH und BMP_HEIGHT in der Programmkopffdatei „bmpauslesen.h“ angepasst werden.

4.7 Programmierung und Programmumgebung

4.7.1 Integrierte Entwicklungsumgebung

Eine integrierte Entwicklungsumgebung (Abkürzung IDE, von englisch integrated development environment) ist eine Sammlung von Computerprogrammen, mit denen die Aufgaben der Softwareentwicklung möglichst ohne Medienbrüche bearbeitet werden können [17].

In diesem Projekt wird Visual Studio 2019 von Microsoft verwendet. VS 2019 ist eine IDE-Software. Hierbei handelt es sich um ein grundlegendes vollständiges Set von Entwicklungstools. Es ist sehr gut in der Unterstützung von C und C++ [18]. Man kann Code mit weniger Fehlern schreiben.

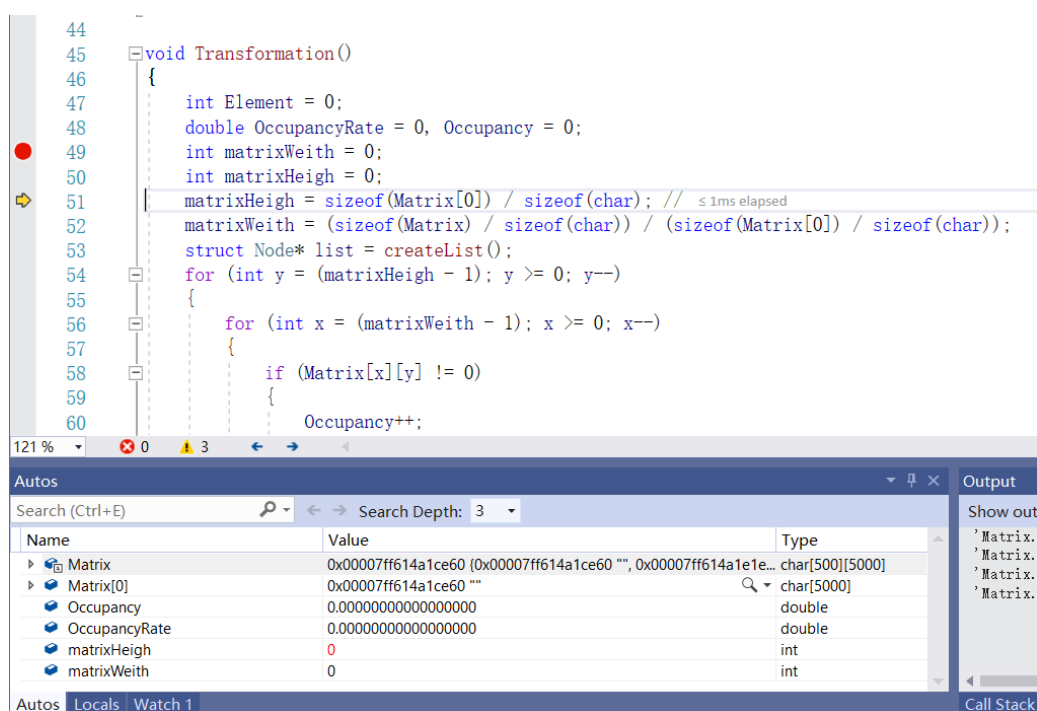


Abbildung 21: Visual Studio-Benutzerschnittstelle

4.7.2 Modulare Programmierung

Modulare Programmierung bedeutet, ein großes Programm nach Funktionen in mehrere kleine Module aufzuteilen und eine Verbindung zwischen diesen Modulen herzustellen.

Die Vorteile der modularen Programmierung sind wie folgt:

- (1) Steuert die Komplexität der Programmierung.
- (2) Verbessern Sie die Wiederverwendbarkeit des Codes.
- (3) Einfache Wartung und Funktionserweiterung.
- (4) Fördert die Teamentwicklung.

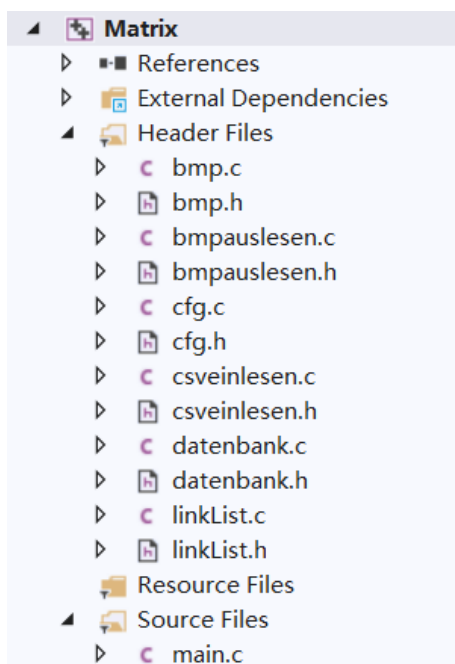


Abbildung 22: Header-Dateien

bmp.h / bmp.c	Generierung von BMP-Dateien
bmpauslesen.h / bmpauslesen.c	Erzeugung von BMP-Dateien mit Matrixdaten
cfg.h / cfg.c	Einlesen der CFG-Datei
csveinlesen.h / csveinlesen.c	Einlesen der CSV-Datei
datenbank.h / datenbank.c	Generierung von der Matrix
linkList.h / linkList.c	Erzeugung von der verketteten Liste

Tabelle 6: Die Funktionen der Header- und C-Dateien

```

#include "bmpauslesen.h"
#include "cfg.h"
#include "linkList.h"

void main()
{
    cfgpath = "d:\\Config\\111052.cfg";
    char* csvpath = "d:\\Database\\Datenbank 111080.csv";
    char* bmpath = "d:\\111080.bmp";
    cfg_einlesen(cfgpath);
    Bmp_Matrix(csvpath, bmpath);
    Transformation();
    system("pause");
}

```

Abbildung 23: Hauptfunktion

Abbildung 23 ist das Hauptprogramm, der Code im Hauptprogramm ist relativ kurz und besteht hauptsächlich aus Funktionen in der Header-Datei.

Die Header-Dateien werden bei

```

#include "bmpauslesen.h"
#include "cfg.h"
#include "linkList.h"

```

aufgerufen.

```

cfgpath = "d:\\Config\\111052.cfg"; // Speicherort der CFG-Datei
char* csvpath = "d:\\Database\\Datenbank 111080.csv"; // Speicherort der CSV-Datei
char* bmpath = "d:\\111080.bmp"; // Speicherort für den Export von BMP-Dateien

```

Die obigen drei Codezeilen sind CFG-, CSV- und BMP-Dateispeicherort. CFG- und CSV-Datei werden in den folgenden Code importiert. Und die BMP-Datei wird generiert.

```

cfg_einlesen(cfgpath); // CFG-Datei einlesen
Bmp_Matrix(csvpath, bmpath); // CSV-Datei einlesen und BMP-Datei exportieren

```

„void cfg_einlesen(cfgpath);“ ist eine Funktion in der „cfg.h“ Header-Dateien.
 „void Bmp_Matrix(csvpath, bmpath);“ ist eine Funktion in der „bmpauslesen.h“ Header-Dateien.

„void Transformation();“ ist in der „linkList.h“ Header-Datei enthalten. Diese Funktion kann die Matrix in eine Linkliste konvertieren.

Der Code im Hauptprogramm ist im Allgemeinen die Definition und Zuweisung von Variablen oder die aufgerufene Funktion. Der größte Teil des Codes wird in die Header-Datei geschrieben. Die folgende Tabelle zeigt die Funktionen, die in jeder Header-Datei aufgerufen werden können, und ihre Funktionen.

Header-Datei	Funktion	Beschreibung
<code>cfg.h</code>	<code>void cfg_einlesen(cfgpath);</code>	CFG-Datei einlesen
<code>bmppauslesen.h</code>	<code>void Bmp_Matrix(const char* csvpath, char* bmppath);</code>	CSV-Datei einlesen und BMP-Datei exportieren
<code>bmp.h</code>	<code>int bmp_create(char* filePath, unsigned char* data, int width, int height, int per);</code>	Ein BMP-Image generieren filePath : BMP-Exportpfad data : Zeiger auf Bildmatrixdaten width : Breite (Pixel) height : Hoch (Pixel) per : Bytes pro Pixel
<code>csveinlesen.h</code>	<code>void func_Chose(Klasse, num, board, spos);</code>	Klassenauswahl Klasse: Klasse des Moduls num: Unterschiedliche Ausrichtung board: Brett-Modell spos: Startpunktkoordinaten
	<code>void csvRead(path);</code>	CSV-Datei einlesen
<code>linkList.h</code>	<code>void Transformation();</code>	Eine Matrix in eine Linkliste konvertieren übertragen

Tabelle 7: Funktionen in Header-Datei

In den Header-Dateien sind Funktionen definiert, die aufgerufen werden können. Jede Header-Datei entspricht einer C-Datei. Die der Header-Datei entsprechende C-Datei wird zur Interpretation der definierten Funktion verwendet.

4.7.3 Flussdiagramm

Abbildung 24 zeigt das Programmablaufdiagramm des gesamten Programms. Das gesamte Programm ist nach Funktionen in mehrere Module unterteilt. Durch den Aufruf dieser Module kann die Struktur von Matrix realisiert werden.



Abbildung 24: Programmablaufplan für das gesamte Projekt

4.7.4 CSV-,CFG-Datei einlesen und Datenbearbeitung

Zuerst wird der Pfad zu den CFG- und CSV-Dateien in das C-Programm eingegeben, um die Datei in das C-Programm zu importieren.

Abbildung 25 zeigt die C-Datei „cfg.c“ Programmablaufplan. Die Daten in der CFG-Datei werden zeilenweise gescannt und der Variablen der entsprechende Wert zugewiesen.

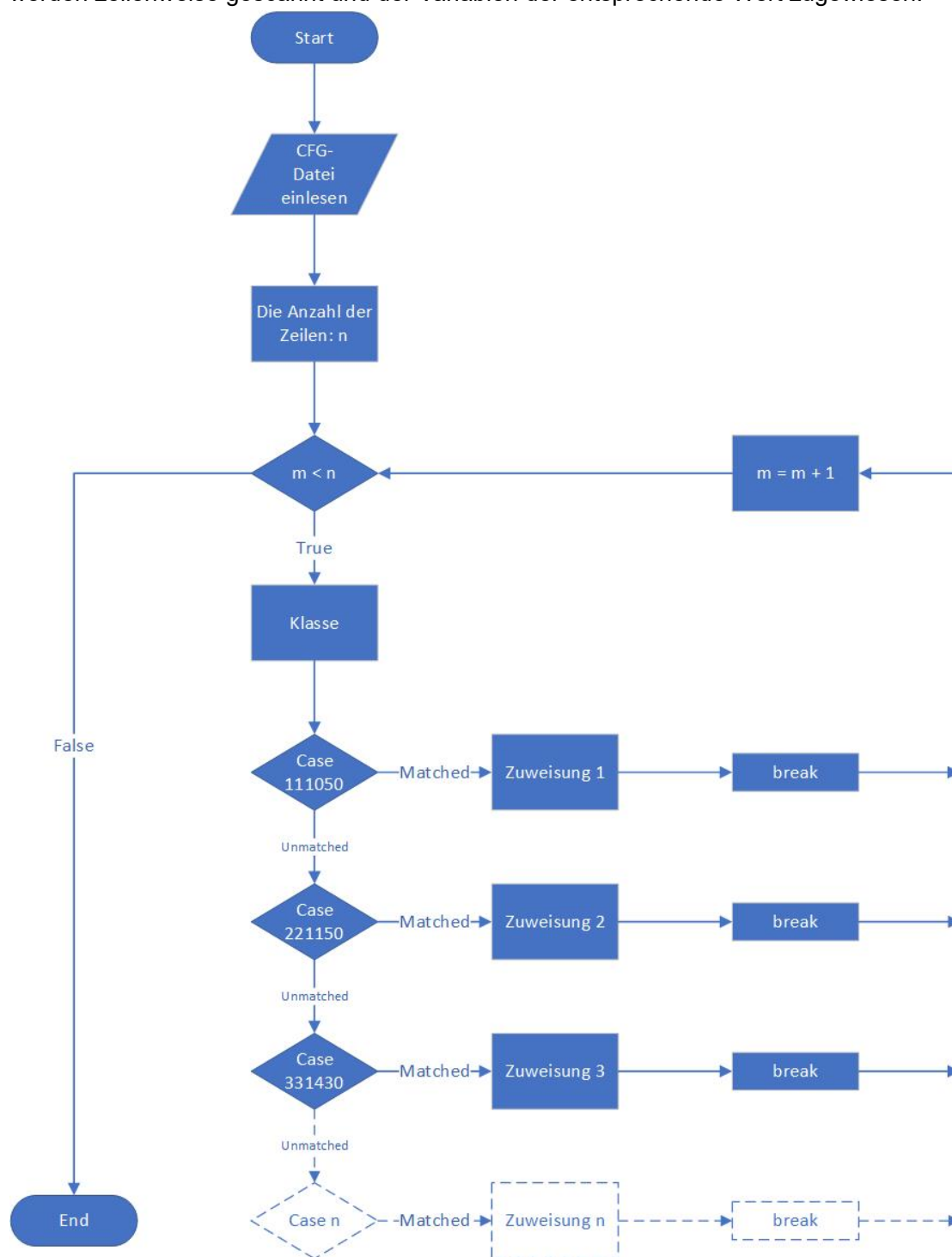


Abbildung 25: Programmablaufplan für das Einlesen der CFG-Datei

Unterschiedliche „Case“ entsprechen unterschiedlichen „Zuweisung“. Die Funktion der „Zuweisung“ besteht darin, der Variablen die Länge des Schienenmoduls zuzuweisen. Die gleiche Methode wird zum Datenlesen von CSV-Dateien verwendet.

4.7.5 Matrix bauen und Visualisierung

Durch die Variablen im vorherigen Schritt wird die Matrix durch die Funktionen im Programm erstellt.

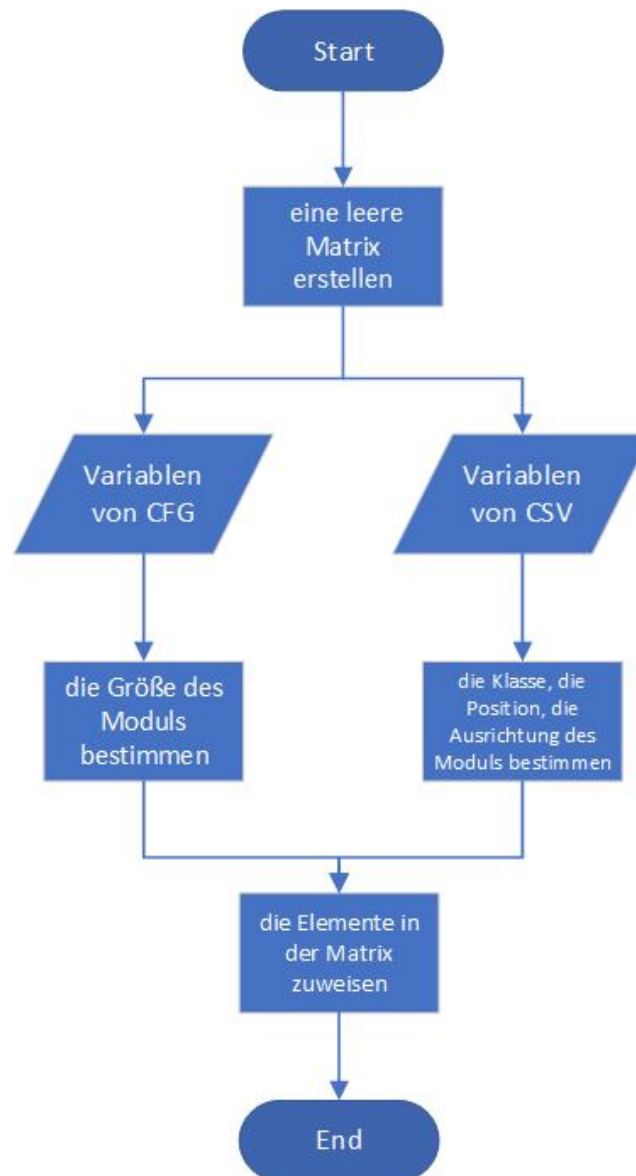


Abbildung 26: Aufbau der Matrix

Zuerst wird eine leere Matrix erstellt. Die Länge jeder Klasse wird durch Einlesen der CFG-Datei festgelegt. Durch Einlesen der CSV-Datei wird die Position und Ausrichtung der Klasse für jedes Modul in der Matrix bestimmt. Zuletzt werden die Elemente in Matrix zugewiesen.

Visualisierung bedeutet, eine BMP-Datei aus einer Matrix zu erstellen. Die Visualisierung ist in Abschnitt 4.6 dargestellt. Die BMP-Datei wird in den eingestellten Pfad exportiert und die Matrix wird in das Umfeldmodell importiert.

Im Folgenden werden die beiden Schritte des Flussdiagramms am Beispiel der Klasse 221150 erläutert.

① Die Größe des Moduls bestimmen:

221150,125,400,675,800,300,425.

Abbildung 27: CFG-Datei des Beispiels

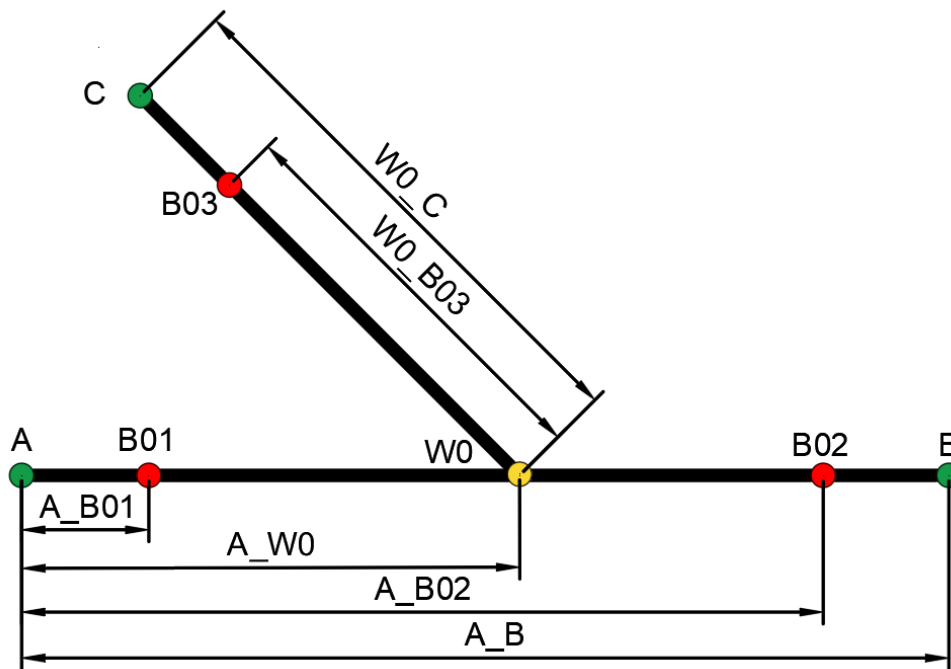


Abbildung 28: Zusammensetzung der Klasse 221150

```
int A_B01_02, A_W0_02, A_B02_02, A_B_02, W0_B03_02, W0_C_02; // K_221150
```

Die obige Codezeile ist die Deklaration der Variablen. Variablen werden durch Analysen der CFG-Datei Werte zugewiesen: $A_B01_02 = 125$, $A_W0_02 = 400$, $A_B02_02 = 675$, $A_B_02 = 800$, $W0_B03_02 = 300$, $W0_C_02 = 425$. Dann wird die Länge jedes Liniensegments in Klasse 221150 bestimmt.

② Die Klasse, die Position, die Ausrichtung des Moduls bestimmen:

221150,100,200,2,1

Abbildung 29: CSV-Datei des Beispiels

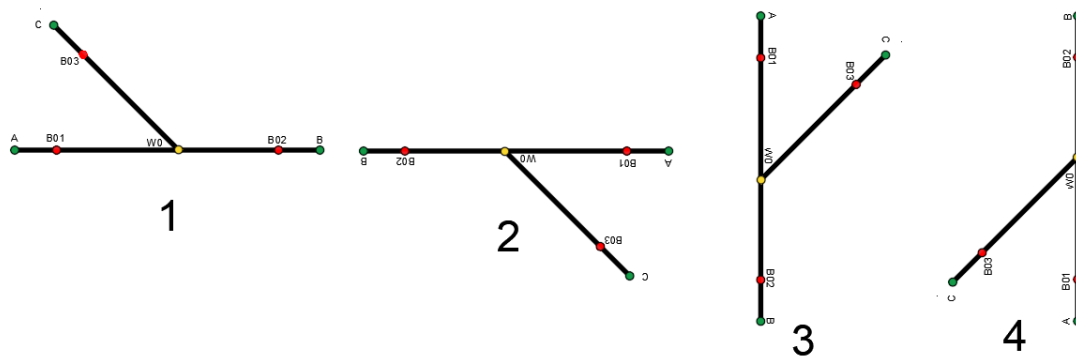


Abbildung 30: Klasse 221150 mit 4 Ausrichtungen

```
typedef struct position
{
    int x;
    int y;
}posType;
```

Der obige Code wird verwendet, um Verbund (Datentyp) zu definieren. Der folgende Code ist die Deklaration der Variablen. „num“ steht für Ausrichtung. "Klasse" wählt die Klasse des Schienenmoduls aus. „board“ steht für die Nummer des Brettes. Die Standardnummer des Brettes ist 1. „spos“ ist der Startpunkt des Schienenmoduls.

```
int num, Klasse, board;
posType spos;
```

Das Programm analysiert die CSV-Datei und weist Variablen Werte zu: Klasse = 221150, num = 2, board = 1, spos.x = 100, spos.y = 200. Matrix wird basierend auf diesen Variablen generiert. Die folgende Abbildung ist eine BMP-Datei, die sich auf die Daten in Matrix bezieht.

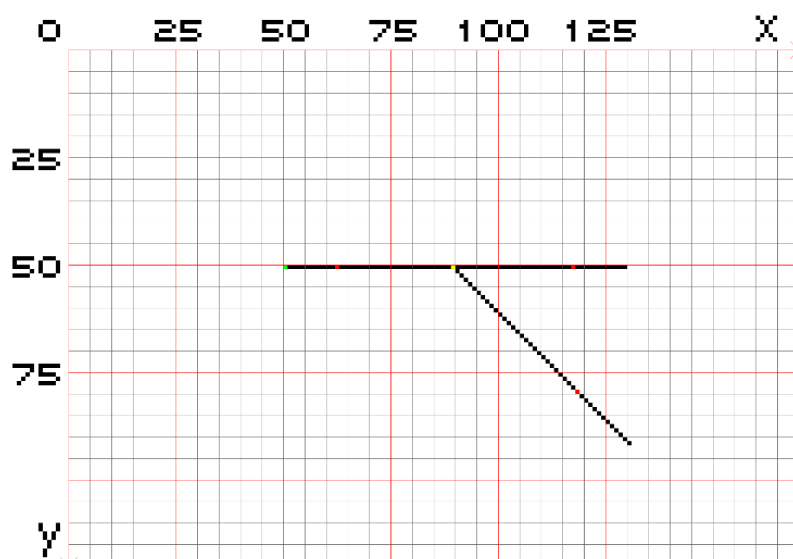


Abbildung 31: Visualisierung vom Beispiel

4.8 Effiziente Datenspeicherung

4.8.1 Zweidimensionaler Array

In der Matrix werden die Daten der Schienen gespeichert, wobei es sich um eine Matrix mit den Abmessungen „ $m * n$ “ handelt. Die Anzahl der Schienendatenblöcke beträgt „ s “. Der Anteil der Schienendatenblöcke beträgt A :

$$A = \frac{s}{m * n} \times 100\%$$

Beispiel:

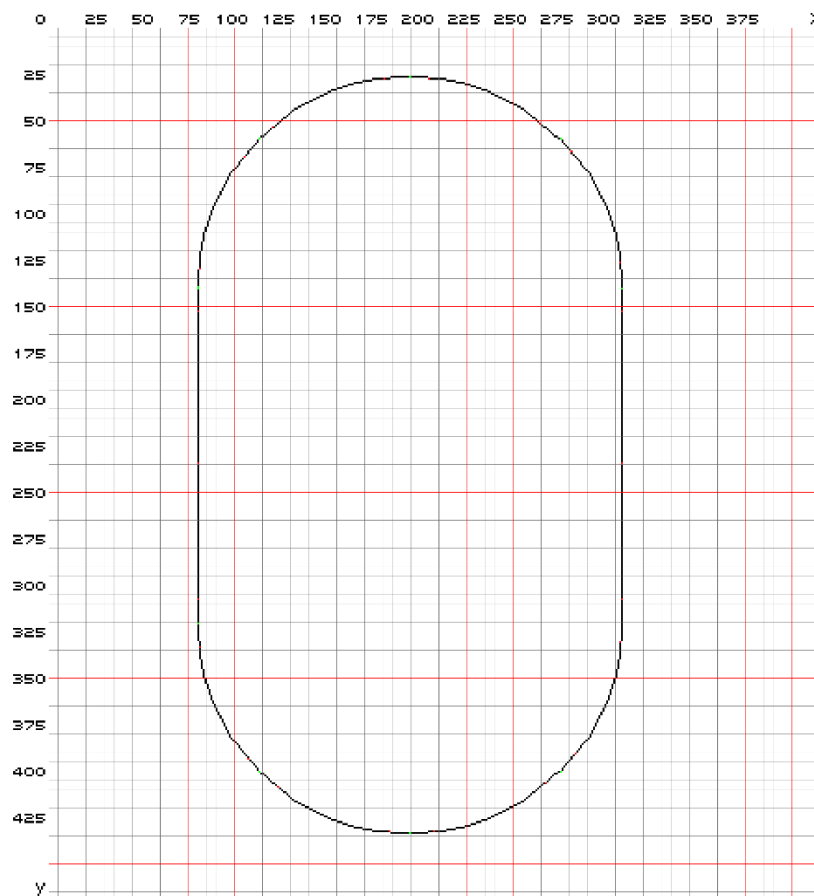


Abbildung 32: Einfaches geschlossenes Schienensystem

Die obige Abbildung zeigt ein einfaches kreisförmiges Gleissystem mit $s = 1004$, $m * n = 450 * 400$. Deshalb ist der Anteil der Schienendatenblöcke A :

$$A = \frac{s}{m * n} \times 100\% = \frac{1004}{450 * 400} \times 100\% = 0.558\%$$

4.8.2 Verkettete Liste

Wie man dem Beispiel oben entnehmen kann, ist das Verhältnis A sehr niedrig. Dies bedeutet, dass in Matrix nur sehr wenige Elemente verwendet werden. Mehr als 99% der Elemente sind unbenutzt und sowohl benutzte als auch unbenutzte Elemente belegen 8 Bits, so dass das Speichern von Daten in einem zweidimensionalen Array ungeeignet ist. Es muss eine neue Datenspeichermethode gefunden werden, um Daten zu speichern, den Speicherplatz zu reduzieren und die Datenverarbeitungsrate zu erhöhen.

Die verkettete Liste ist eine dynamische Datenstruktur, die eine geordnete Speicherung von Datenelementen implementiert [19]. Durch die Verwendung von verknüpften Listen kann der Speicherplatz für Daten reduziert werden.

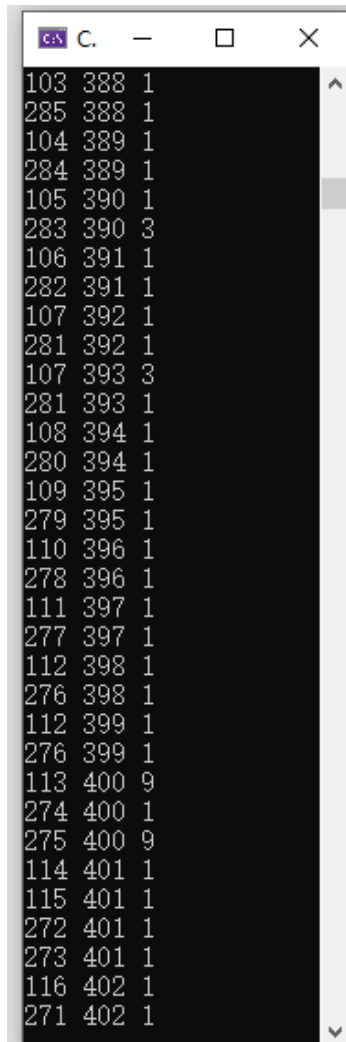
In diesem Projekt wird die Matrix mit dem folgenden Programm in eine verkettete Liste konvertiert.

```
void Transformation()
{
    int Element = 0;
    double OccupancyRate = 0, Occupancy = 0;
    int matrixWeith = 0;
    int matrixHeigh = 0;
    matrixHeigh = sizeof(Matrix[0]) / sizeof(char);
    matrixWeith = (sizeof(Matrix) / sizeof(char)) / (sizeof(Matrix[0]) /
sizeof(char)); // Die Anzahl der Zeilen und Spalten eines zweidimensionalen
Arrays
    struct Node* list = createList(); // Listekopf erstellen
    for (int y = (matrixHeigh - 1); y >= 0; y--)
    {
        for (int x = (matrixWeith - 1); x >= 0; x--)
        {
            if (Matrix[x][y] != 0)
            {
                Occupancy++;
                Element = Matrix[x][y];
                insertNodeByHead(list, x, y, Element); // Wenn Matrix [x] [y]
nicht 0 ist, werden x, y und Matrix [x] [y] in die verknüpfte Liste importiert.
            }
        }
    }
    OccupancyRate = Occupancy / ((double)(matrixWeith * matrixHeigh));
    // der Anteil der Schienendatenblöcke
    printList(list); // Liste drucken
    printf("%0.9f%\n", OccupancyRate);
}
```



```
void printList(struct Node* headNode) // Liste drucken
{
    struct Node* pMove = headNode->next;
    while (pMove)
    {
        printf("%d %d %d", pMove->x, pMove->y, pMove->Element);
        pMove = pMove->next;
        printf("\n");
    }
}
```

Liste durch Aufruf der Funktion `void printList(struct Node* headNode);` drucken.
Im Allgemeinen ist es nicht erforderlich, Listen auszudrucken, Man kann die Funktion nicht aufrufen.



```
C.  -  □  ✕
103 388 1
285 388 1
104 389 1
284 389 1
105 390 1
283 390 3
106 391 1
282 391 1
107 392 1
281 392 1
107 393 3
281 393 1
108 394 1
280 394 1
109 395 1
279 395 1
110 396 1
278 396 1
111 397 1
277 397 1
112 398 1
276 398 1
112 399 1
276 399 1
113 400 9
274 400 1
275 400 9
114 401 1
115 401 1
272 401 1
273 401 1
116 402 1
271 402 1
```

Abbildung 33: die verkettete Liste drucken

4.9 Lokalisierung der eigenen Position

Die Position des Zuges wird durch die Balisen auf der Schiene bestimmt und die Koordinatendaten werden an das Programm übertragen. Es gibt zwei Möglichkeiten, den Zug zu lokalisieren.

- ① Der erste Weg, die Position zu bestimmen, führt über die Balise. z. B. Wenn der Zug die erste Balise B01 auslöst, werden die Koordinateninformationen von B01 in das Programm übertragen und das Beleg-bit in dem entsprechenden Element in der Matrix auf „1“ gesetzt. Wenn der Zug die nächste Balise B02 passiert, wird das Beleg-bit der alten Position auf „0“ gesetzt und das belegt-Bit des neuen Standorts auf „1“ gesetzt.
- ② Die zweite Methode ist komplizierter als die erste Methode. Zuerst wird die Anfangsposition bestimmt, und die Zugkoordinaten werden nach Zuggeschwindigkeit und Intervall T Sekunden an das Programm übergeben. Weil T frei eingestellt werden kann, ist es flexibler als die erste Methode.

Wenn sich mehrere Züge in einem Gleissystem befinden, hat jeder Zug ein eigenes Umfeldmodell. Ego repräsentiert die Koordinaten des Zuges in seinem eigenen Umfeldmodell, während die Koordinaten anderer Züge auch in diesem Umfeldmodell angezeigt werden. Die koordinierte Informationskommunikation zwischen Umfeldmodellen wird von WLAN implementiert. Das Folgende ist ein Beispiel.

Beispiel:

Die folgende Abbildung zeigt die Umfeldmodelle verschiedener Züge im selben Gleissystem. Jedes Umfeldmodell verfügt über eine Tabelle zum Speichern von Fahrzeugkoordinaten. Wenn sich die Zugposition ändert, ändern sich auch die Daten in der Tabelle.

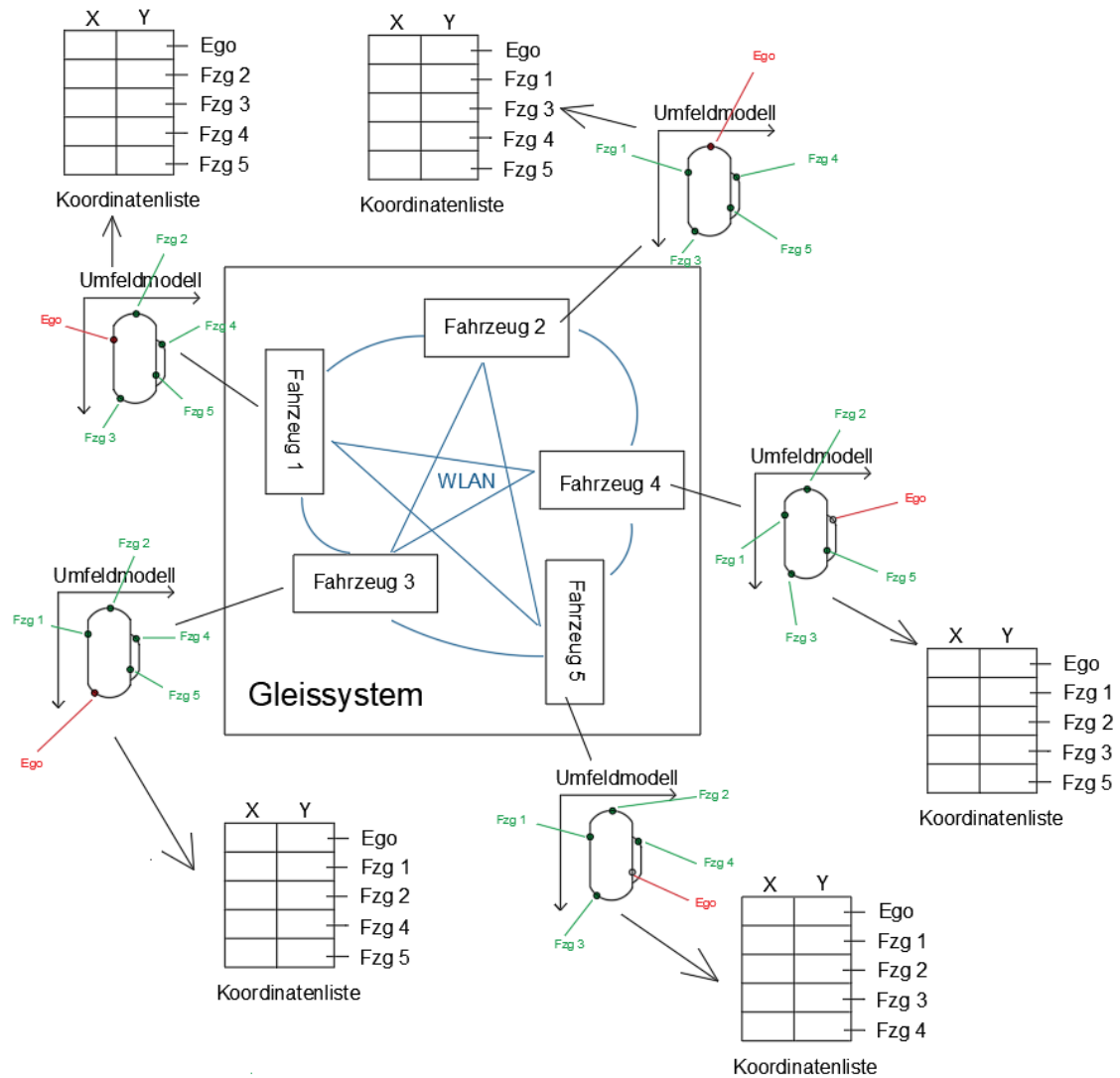


Abbildung 34: Umfeldmodelle im Gleissystem

5. Automatische Generierung von CSV-Dateien

In diesem Kapitel werden einige ungelöste Probleme in der CSV-Datei beschrieben und die einige Ideen und Lösungsvorschläge bereitgestellt.

Wie man in Abschnitt 4.4 sehen kann, muss das Erstellen einer CSV-Datei die spezifische Länge jedes Moduls kennen, um den Startpunkt jedes Moduls zu berechnen. Diese Berechnung muss mehrmals wiederholt werden und ist relativ zeitaufwendig.

Klasse vom M01	Abszisse des Startpunktes vom ganzen Gleissystem	Ordinate des Startpunktes vom ganzen Gleissystem	Ausrichtung vom M01	Brettnummer vom M01
Klasse vom M02	Abszisse des Startpunktes vom M02	Ordinate des Startpunktes vom M02	Ausrichtung vom M02	Brettnummer vom M02
Klasse vom M03	Abszisse des Startpunktes vom M03	Ordinate des Startpunktes vom M03	Ausrichtung vom M03	Brettnummer vom M03
Klasse vom M04	Abszisse des Startpunktes vom M04	Ordinate des Startpunktes vom M04	Ausrichtung vom M04	Brettnummer vom M04
Klasse vom M05	Abszisse des Startpunktes vom M05	Ordinate des Startpunktes vom M05	Ausrichtung vom M05	Brettnummer vom M05
Klasse vom M06	Abszisse des Startpunktes vom M06	Ordinate des Startpunktes vom M06	Ausrichtung vom M06	Brettnummer vom M06
.



Problem: Alle Punkte außer dem Anfangspunkt des Systems müssen berechnet werden.

Abbildung 35: CSV-Datei und ihr Problem

Um die Erstellung von CSV-Dateien zu vereinfachen, muss man eine neue Konstruktionsmethode finden. Das Ziel ist: Es ist nicht erforderlich, die Koordinaten jedes Startpunktes zu kennen. Die Benutzer müssen die Schienenmodule nur über eine andere Software aufrufen und diese Module im Schienensystem kombinieren, um CSV-Dateien zu generieren. Die Abbildung 36 zeigt eine Idee.

- **Auswahl geeigneter Zeichenprogramme**

Die Software sollte CFG-Dateien einlesen und CSV-Dateien exportieren können.

- **Template und Bild bauen**

Template wird in der Zeichen-Software erstellt, die alle Eisenbahnklassen enthält.

Benutzer können diese Module aus der Vorlage aufrufen, um das gewünschte Modul zu bilden. Anschließend erkennt die Software den Startpunkt und die Ausrichtung jedes Moduls und exportiert sie in die CSV-Datei.

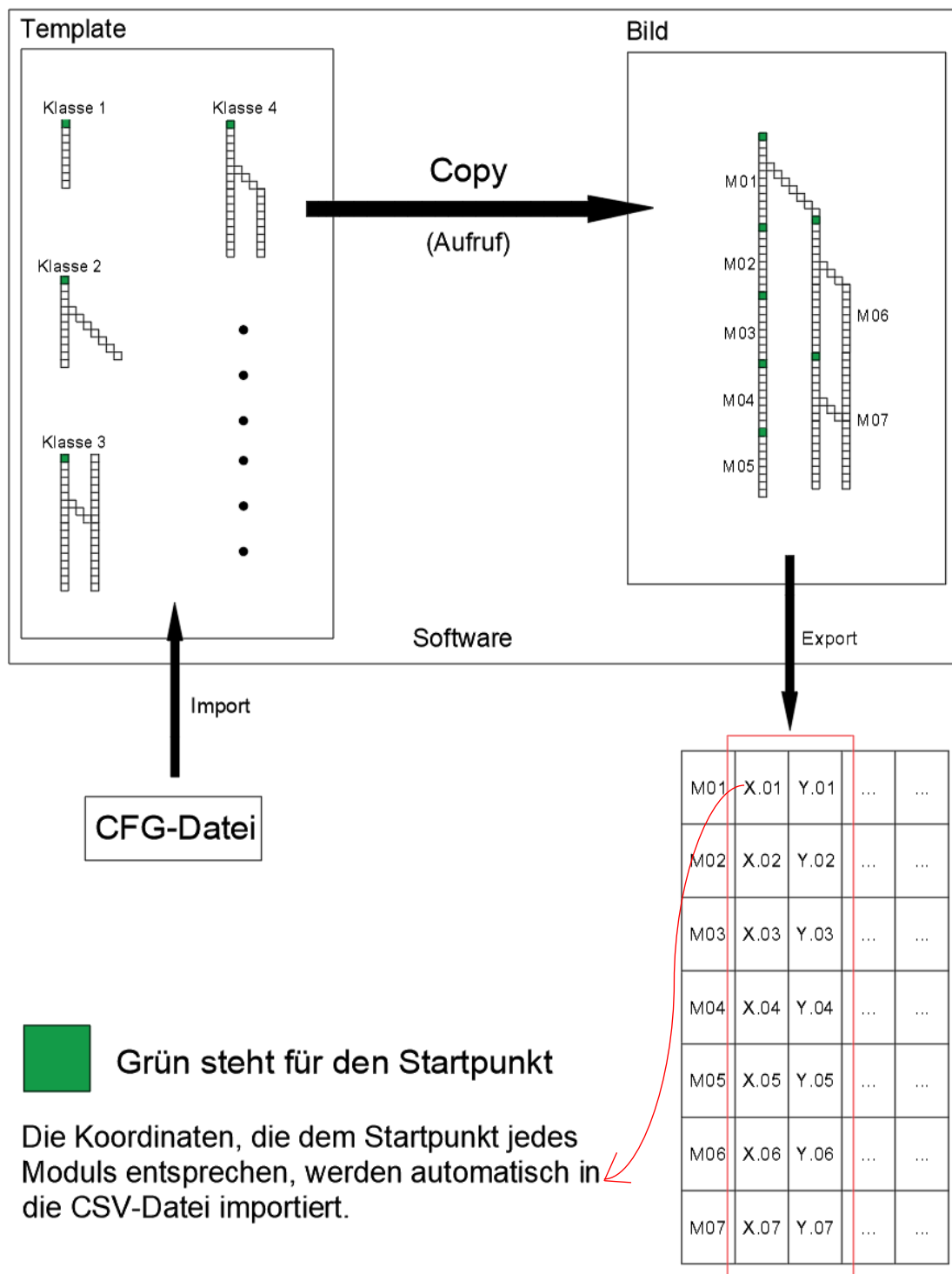


Abbildung 36: Grundidee

6. Zusammenfassung und Ausblick

6.1 Zusammenfassung

Das Ziel der Bachelorarbeit ist es, ein Umfeldmodell aufzubauen. Das Umfeldmodell ist die Grundlage des autonomen Systems. Es wird zwischen Zügen eingesetzt, um die gegenseitige Position zu bestätigen und Konflikte zu vermeiden. Daher ist es notwendig, ein geeignetes Umfeldmodell aufzustellen.

Das gesamte Projekt widmet sich der Erforschung der Modernisierung des Verkehrsnetzes. Dies ist eine Ära der rasanten Entwicklung der Informationstechnologie. Mit der Entwicklung der Gesellschaft sind jedoch auch viele soziale und ökologische Probleme aufgetreten. Um die Umwelt zu verbessern, müssen ständig neue Transporttechnologien entwickelt werden, um einige traditionelle Transportmethoden zu ersetzen. Und das Autonome Fahren kann das Problem des durch die Alterung der Gesellschaft verursachten Arbeitskräftemangels lösen.

Das Umfeldmodell ist ein wichtiger Bestandteil des autonomen Fahrens. Beim Einrichten des Umgebungsmodells sind viele Probleme aufgetreten, z. B. Datenspeichermethoden, Matrixauflösung und der Import von CSV- und CFG-Dateien. Diese Probleme wurden nicht vollständig gelöst. Die Art und Weise, wie CFG-Dateien generiert werden, sollte ebenfalls überdacht werden. Wie die Arbeitseffizienz autonomer Systeme verbessert werden kann, wird die Forschungsrichtung sein.

Das autonome System enthält nicht nur Umfeldmodell sondern auch viele Aktoren und Sensoren, wie Balisen und Weichen. Das WLAN-Kommunikationsmodul wird ebenfalls verwendet. Die Informationsverarbeitung wird vom Bordcomputer übernommen. Die Entwicklung des autonomen Systems steckt noch in den Kinderschuhen, und viele Probleme sind noch nicht gelöst.

Das autonome System hilft bei der Lösung bestehender Verkehrsprobleme. Durch die Reduzierung des LKW-Einsatzes können Kosten gespart und Schäden an Straßen und Brücken verringert werden. Das autonome System wird im künftigen Verkehr eine wichtige Rolle spielen.

6.2 Ausblick

1. CSV-Datei wird durch Zeichensoftware generiert. Es ist keine Berechnung erforderlich.

- ① CFG-Datei kann in Zeichensoftware importiert werden. Die Modulgröße in der Tamplate wird durch die CFG-Datei geändert.
- ② Die Tamplate wird aufgerufen, um das Gleissystem wie ein Puzzle aufzubauen.
- ③ Die Daten in der Zeichen-Software werden in die CSV-Datei importiert.

2. Datenimport des Zugstandorts.

Die Daten für den Zugstandort werden in das Programm importiert und ändern dann die Elemente in der Matrix. Die anderen Daten in der Matrix ändern sich nicht.

3. Umfeldmodell

- ① Matrix oder die verkettete Liste wird in den Automikrocomputer importiert.
- ② Zugkoordinaten per WLAN zum Umfeldmodell eines anderen Zuges übertragen.

4. Definition von Matrixelementen

Wie aus Abschnitt 4.1 ersichtlich, beträgt die Auflösung des Matrixelements 10 cm x 10 cm. Um den Fehler zu reduzieren, kann man die kleinere Auflösung verwenden. Dadurch beansprucht die Matrix jedoch mehr Speicherplatz, und das Programm wird weniger effizient ausgeführt. Und die Funktion des Aufbaus des Schienenmoduls im Programm muss neu geschrieben werden. Das muss weiter diskutiert werden, um die passende Auflösung auszuwählen.

Literatur

- [1] Prof. Dr. -Ing. Christan Schulz: Gedanken zur grundlegenden technischen Umgestaltung der Bahn

- [2] Wikipedia: „Eisenbahn“, unter: <https://de.wikipedia.org/wiki/Eisenbahn> (abgerufen am 01.10.2019).

- [3] Mengtian Wan, Infrarot-Kommunikation und Lichtsteuerung für ATLAS-Fahrzeug, Mittweida, 2018

- [4] Prof. Dr. -Ing. Christian Schulz: Denkschrift zum Projekt ATLAS, Hochschule Mittweida 2006

- [5] Prof. Dr. -Ing. Jan. Thomanek, INW, Ausgewählte Kapitel der ET/IT-Sensordatenfusion im Automobil

- [6] Modul Ausgewählte Kapitel der Elektro- und Informationstechnik - Sensordatenfusion im Fahrzeug, 2019, Prof. Dr. -Ing. Jan. Thomanek

- [7] all-electronics: „V2X“, unter: <https://www.all-electronics.de/abkuerzungsverzeichnis/v2x/#> (abgerufen am 20.11.2019).

- [8] Kinney, Sean: „What is C-V2X“, unter: <https://www.rcrwireless.com/20180601/network-infrastructure/what-is-c-v2x-tag17-tag99#prettyPhoto> (abgerufen am 20.11.2019).

- [9] Modul Ausgewählte Kapitel der Elektro- und Informationstechnik - Sensordatenfusion im Fahrzeug -, 2019, Prof. Dr. -Ing. J. Thomanek

- [10] Mengtian Wan, Infrarot-Kommunikation und Lichtsteuerung für ATLAS-Fahrzeug, Mittweida, 2018
- [11] Wikipedia: „Autonomes Fahren“, unter:
https://de.wikipedia.org/wiki/Autonomes_Fahren
(abgerufen am 21.10.2019).
- [12] Tao, shao: „SPI-Kommunikationsprotokoll (SPI-Bus) lernen“, unter:
<https://www.cnblogs.com/deng-tao/p/6004280.html>
(abgerufen am 15.11.2019).
- [13] Multisensorsysteme im Automatisierten und Autonomen Fahrzeug, Jan Thomanek, IAV, Camotstraße 1, D-10587 Berlin
- [14] Shiyu Zhang, Minbo Shi, Überarbeitung von Gleismodulen und Bau neuer Module, Mittweida, 2018
- [15] CSV und Excel richtiger Importvorgang, 2014, CISS TDI GmbH, Sinzig
- [16] Wikipedia: „Konfigurationsdatei“, unter:
<https://de.wikipedia.org/wiki/Konfigurationsdatei>
(abgerufen am 09.10.2019).
- [17] Wikipedia: „Integrated development environment“, unter:
https://en.wikipedia.org/wiki/Integrated_development_environment
(abgerufen am 11.11.2019).
- [18] Microsoft: „Visual Studio 2019“, unter:
<https://visualstudio.microsoft.com/de/vs/> (abgerufen am 11.11.2019).
- [19] Wikipedia: „Liste_(Datenstruktur)“, unter:
[https://de.wikipedia.org/wiki/Liste_\(Datenstruktur\)](https://de.wikipedia.org/wiki/Liste_(Datenstruktur))
(abgerufen am 05.10.2019).
- [20] Naifan Bu, Gaoxian Wan, ATLAS-System-Datenbank, Mittweida, 2017

Anhang

Längenangepasbare Gleismodule.....	55
Gleismodule mit einer bestimmten Länger	70
Darstellung ausgewählter Gleispläne im Umfeldmodell	74

Längenangepasbare Gleismodule

Klasse: 111053 & 111063



Abbildung 37: Gleismodul [111053 und 111063]

Eintrag in der CFG-Datei für Klasse 111053 und 111063:

```
111053,A_B01,A_B02,A_B03,A_B.
111063,A_B01,A_B02,A_B03,A_B.
```

Abbildung 38: Gleismodul [111053 und 111063]

Die Bedeutung jedes Elements:

111053/111063	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
A_B	Entfernung von A nach B.

Tabelle 8: Die Bedeutung von jedem Element (1)

Beispiel:

```
111053,125,500,700,950.
111063,200,375,600,870.
```

Abbildung 39: Beispiel von CFG-Dateien

Klasse: 111050 & 111060



Abbildung 40: Gleismodul [111050 und 111060]

Eintrag in der CFG-Datei für Klasse 111050 und 111060:

```
111050,A_B.
111060,A_B.
```

Abbildung 41: Aufbau [111050 und 111060]

Die Bedeutung jedes Elements:

111050/111060	Klasse.
A_B	Entfernung von A nach B.

Tabelle 9: Die Bedeutung von jedem Element (2)

Klasse: 111051 & 111061



Abbildung 42: Gleismodul [111051 und 111061]

Eintrag in der CFG-Datei für Klasse 111051 und Klasse 111061:

```
111051,A_B01,A_B.
111061,A_B01,A_B.
```

Abbildung 43: Aufbau [111051 und 111061]

Die Bedeutung jedes Elements:

111051/111061	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B	Entfernung von A nach B.

Tabelle 10: Die Bedeutung von jedem Element (3)

Klasse: 111052& 111062

Abbildung 44: Gleismodul [111052 und 111062]

Eintrag in der CFG-Datei für Klasse 111052 und Klasse 111062:

```
111052,A_B01,A_B02,A_B.  
111062,A_B01,A_B02,A_B.
```

Abbildung 45: Aufbau [111052 und 111062]

Die Bedeutung jedes Elements:

111052/111062	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B	Entfernung von A nach B.

Tabelle 11: Die Bedeutung von jedem Element (4)

Klasse: 221150 & 221160

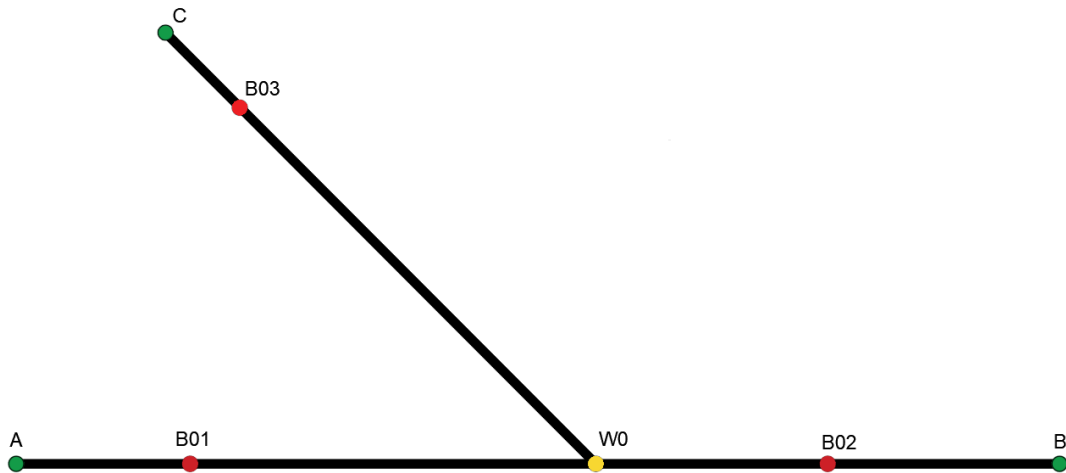


Abbildung 46: Gleismodul [221150 & 221160]

Eintrag in der CFG-Datei für Klasse 221150 und Klasse 221160:

```
221150,A_B01,A_W0,A_B02,A_B,W0_B03,W0_C.
221160,A_B01,A_W0,A_B02,A_B,W0_B03,W0_C.
```

Abbildung 47: Aufbau [221150 & 221160]

Die Bedeutung jedes Elements:

221150/221160	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_W0	Entfernung von A nach Weiche W0.
A_B02	Entfernung von A nach Balise B02.
A_B	Entfernung von A nach B.
W0_B03	Entfernung von Weiche W0 nach Balise B03.
W0_C	Entfernung von Weiche W0 nach C.

Tabelle 12: Die Bedeutung von jedem Element (5)

Klasse: 221151 & 221161

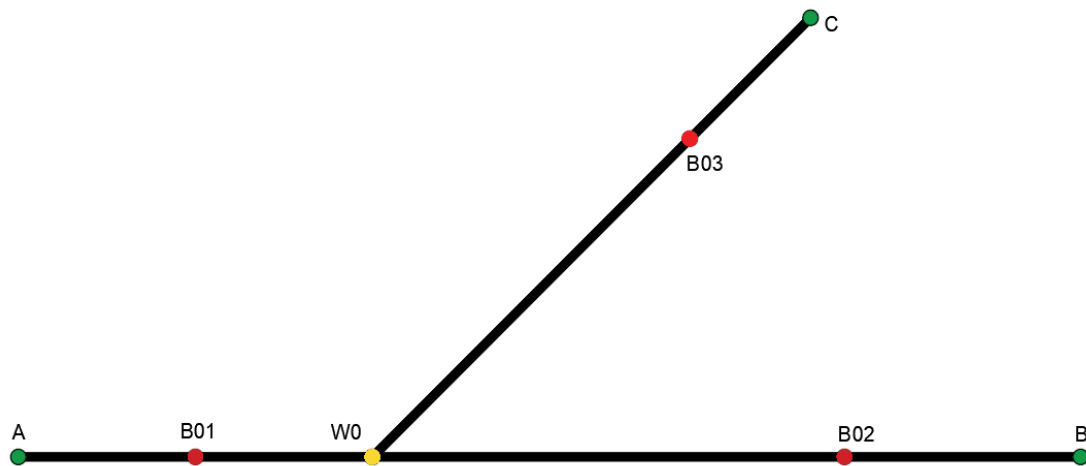


Abbildung 48: Gleismodul [221151 & 221161]

Eintrag in der CFG-Datei für Klasse 221151 und Klasse 221161:

```
221151,A_B01,A_W0,A_B02,A_B,W0_B03,W0_C.
221161,A_B01,A_W0,A_B02,A_B,W0_B03,W0_C.
```

Abbildung 49: Aufbau [221151 & 221161]

Die Bedeutung jedes Elements:

221151/221161	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_W0	Entfernung von A nach Weiche W0.
A_B02	Entfernung von A nach Balise B02.
A_B	Entfernung von A nach B.
W0_B03	Entfernung von Weiche W0 nach Balise B03.
W0_C	Entfernung von Weiche W0 nach C.

Tabelle 13: Die Bedeutung von jedem Element (6)

Klasse: 221130

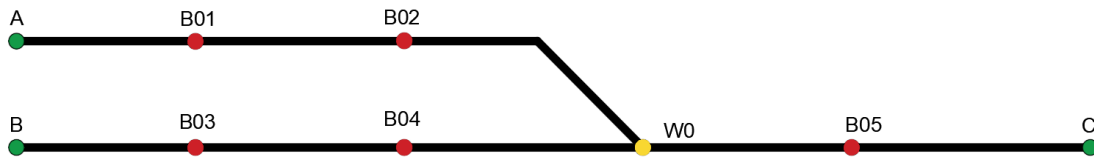


Abbildung 50: Gleismodul [221130]

Eintrag in der CFG-Datei für Klasse 221130:

```
221130,A_B01,A_B02,B_B03,B_B04,B_B05,B_W0,B_C,A_B.
```

Abbildung 51: Aufbau [221130]

Klasse: 221131

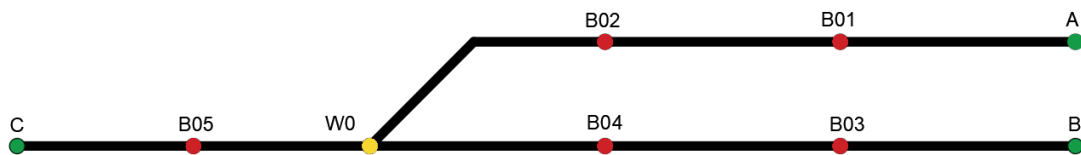


Abbildung 52: Gleismodul [221131]

Eintrag in der CFG-Datei für Klasse 221131:

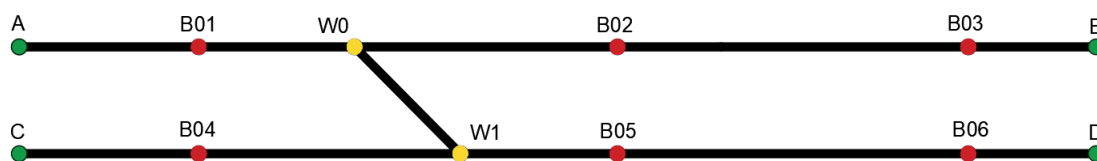
```
221131,A_B01,A_B02,B_B03,B_B04,B_B05,B_W0,B_C,A_B.
```

Abbildung 53: Aufbau [221131]

Die Bedeutung jedes Elements:

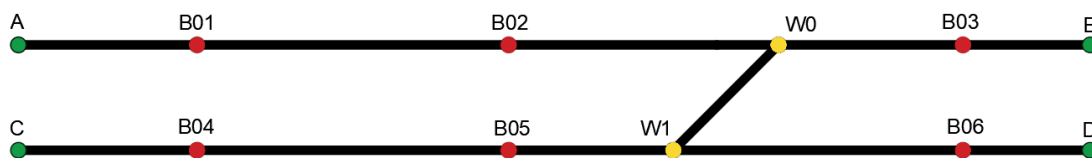
221130/221131	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
B_B03	Entfernung von B nach Balise B03.
B_B04	Entfernung von B nach Balise B04.
B_B05	Entfernung von B nach Balise B05.
B_W0	Entfernung von B nach Weiche W0.
B_C	Entfernung von B nach C.
A_B	Entfernung von A nach B.

Tabelle 14: Die Bedeutung von jedem Element (7)

Klasse: 222230**Abbildung 54: Gleismodul [222230]**

Eintrag in der CFG-Datei für Klasse 222230:

```
222230,A_B01,A_B02,A_B03,C_B04,C_B05,C_B06,A_B,C_D,A_W0,A_C.
```

Abbildung 55: Aufbau [222230]**Klasse: 222231****Abbildung 56: Gleismodul [222231]**

Eintrag in der CFG-Datei für Klasse 222231:

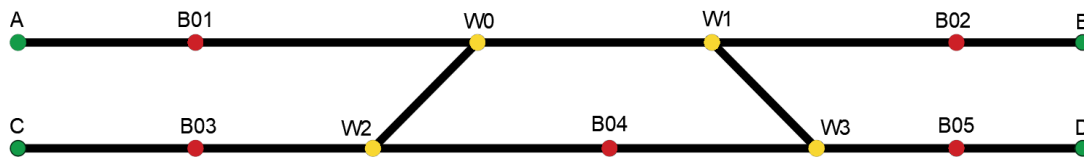
```
222231,A_B01,A_B02,A_B03,C_B04,C_B05,C_B06,A_B,C_D,A_W0,A_C.
```

Abbildung 57: Aufbau [222231]

Die Bedeutung jedes Elements:

222230/222231	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
C_B04	Entfernung von C nach Balise B04.
C_B05	Entfernung von C nach Balise B05.
C_B06	Entfernung von C nach Balise B06.
A_B	Entfernung von A nach B.
C_D	Entfernung von C nach D.
A_W0	Entfernung von A nach Weiche W0.
A_C	Entfernung von A nach C.

Tabelle 15: Die Bedeutung von jedem Element (8)

Klasse: 222430**Abbildung 58: Gleismodul [222430]**

Eintrag in der CFG-Datei für Klasse 222430:

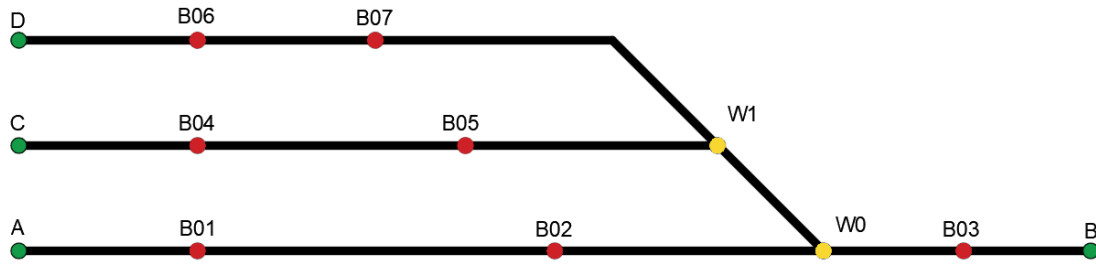
```
222430,A_B01,A_B02,C_B03,C_B04,C_B05,A_B,C_D,A_W0,A_W1,A_C.
```

Abbildung 59: Aufbau [222430]

Die Bedeutung jedes Elements:

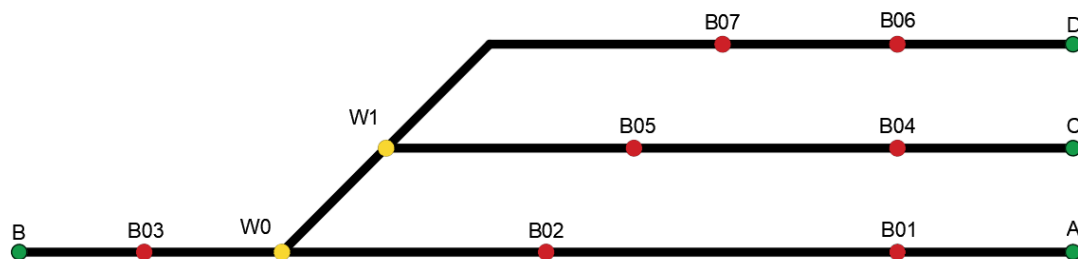
222430	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
C_B03	Entfernung von C nach Balise B03.
C_B04	Entfernung von C nach Balise B04.
C_B05	Entfernung von C nach Balise B05.
A_B	Entfernung von A nach B.
C_D	Entfernung von C nach D.
A_W0	Entfernung von A nach Weiche W0.
A_W1	Entfernung von A nach Weiche W1.
A_C	Entfernung von A nach C.

Tabelle 16: Die Bedeutung von jedem Element (9)

Klasse: 331230**Abbildung 60: Gleismodul [331230]**

Eintrag in der CFG-Datei für Klasse 331230:

```
331230,A_B01,A_B02,A_B03,C_B04,C_B05,D_B06,D_B07,A_W0,A_B,A_C,C_D.
```

Abbildung 61: Aufbau [331230]**Klasse: 331231****Abbildung 62: Gleismodul [331231]**

Eintrag in der CFG-Datei für Klasse 331231:

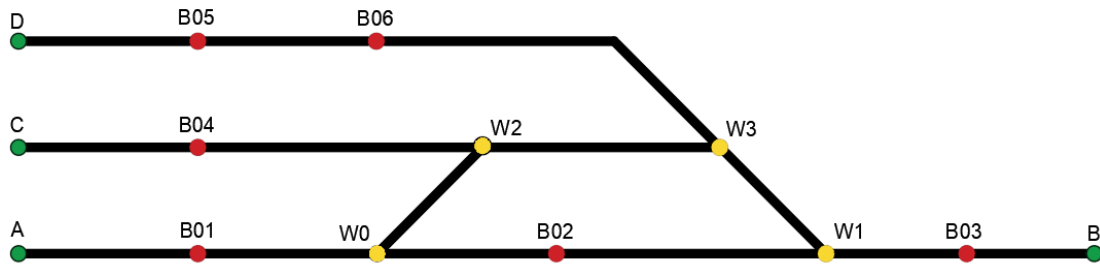
```
331231,A_B01,A_B02,A_B03,C_B04,C_B05,D_B06,D_B07,A_W0,A_B,A_C,C_D.
```

Abbildung 63: Aufbau [331231]

Die Bedeutung jedes Elements:

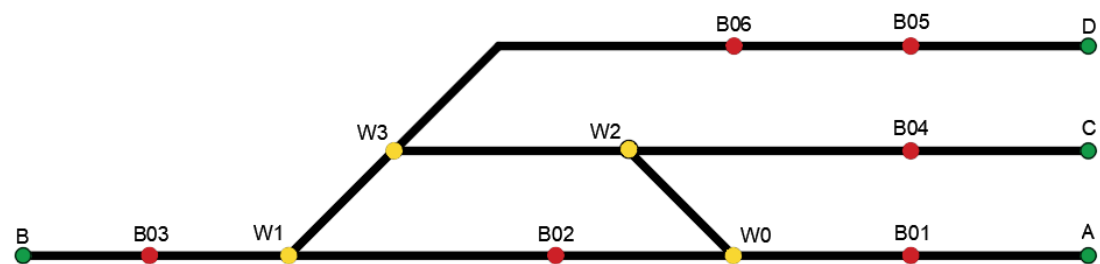
331230/331231	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
C_B04	Entfernung von C nach Balise B04.
C_B05	Entfernung von C nach Balise B05.
D_B06	Entfernung von D nach Balise B06.
D_B07	Entfernung von D nach Balise B07.
A_W0	Entfernung von A nach Weiche W0.
A_B	Entfernung von A nach B.
A_C	Entfernung von A nach C.
C_D	Entfernung von C nach D.

Tabelle 17: Die Bedeutung von jedem Element (10)

Klasse: 331430**Abbildung 64: Gleismodul [331430]**

Eintrag in der CFG-Datei für Klasse 331430:

```
331430,A_B01,A_B02,A_B03,C_B04,D_B05,D_B06,A_W0,A_W1,A_B,A_C,C_D.
```

Abbildung 65: Aufbau [331430]**Klasse: 331431****Abbildung 66: Gleismodul [331431]**

Eintrag in der CFG-Datei für Klasse 331431:

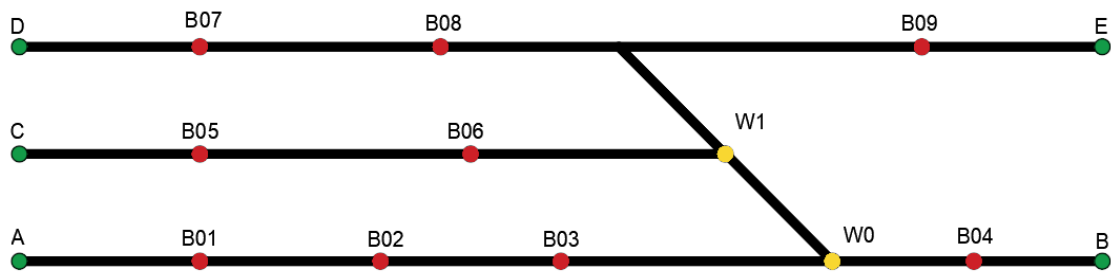
```
331431,A_B01,A_B02,A_B03,C_B04,D_B05,D_B06,A_W0,A_W1,A_B,A_C,C_D.
```

Abbildung 67: Aufbau [331431]

Die Bedeutung jedes Elements:

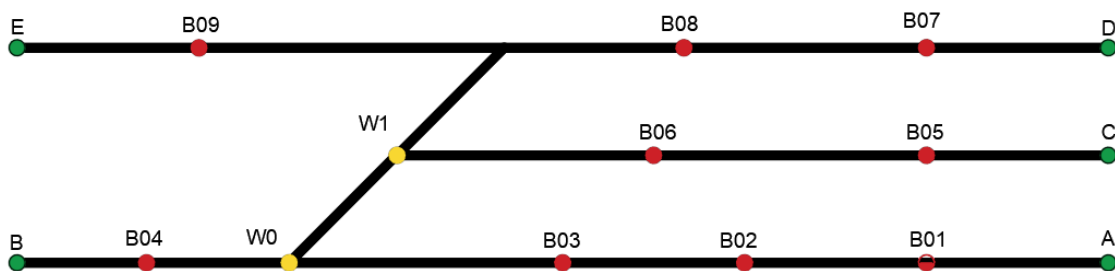
3331430/331431	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
C_B04	Entfernung von C nach Balise B04.
D_B05	Entfernung von D nach Balise B05.
D_B06	Entfernung von D nach Balise B06.
A_W0	Entfernung von A nach Weiche W0.
A_W1	Entfernung von A nach Weiche W1.
A_B	Entfernung von A nach B.
A_C	Entfernung von A nach C.
C_D	Entfernung von C nach D.

Tabelle 18: Die Bedeutung von jedem Element (11)

Klasse: 332330**Abbildung 68: Gleismodul [332330]**

Eintrag in der CFG-Datei für Klasse 332330:

```
332330,A_B01,A_B02,A_B03,A_B04,C_B05,C_B06,D_B07,D_B08,D_B09,A_B,D_E,A_W0,A_C,C_D.
```

Abbildung 69: Aufbau [332330]**Klasse: 332331****Abbildung 70: Gleismodul [332331]**

Eintrag in der CFG-Datei für Klasse 332331:

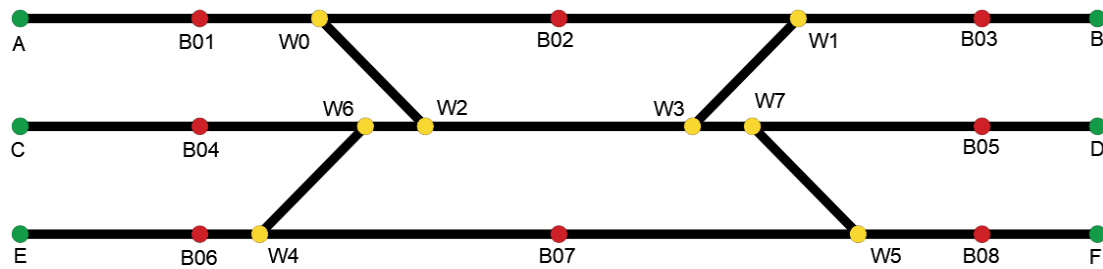
```
332331,A_B01,A_B02,A_B03,A_B04,C_B05,C_B06,D_B07,D_B08,D_B09,A_B,D_E,A_W0,A_C,C_D.
```

Abbildung 71: Aufbau [332331]

Die Bedeutung jedes Elements:

332330/332331	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
A_B04	Entfernung von A nach Balise B04.
C_B05	Entfernung von C nach Balise B05.
C_B06	Entfernung von C nach Balise B06.
D_B07	Entfernung von D nach Balise B07.
D_B08	Entfernung von D nach Balise B08.
D_B09	Entfernung von D nach Balise B09.
A_B	Entfernung von A nach B.
D_E	Entfernung von D nach E.
A_W0	Entfernung von A nach Weiche W0.
A_C	Entfernung von A nach C.
C_D	Entfernung von C nach D.

Tabelle 19: Die Bedeutung von jedem Element (12)

Klasse: 333830**Abbildung 72: Gleismodul [333830]**

Eintrag in der CFG-Datei für Klasse 338330:

```
333830,A_B01,A_B02,A_B03,C_B04,C_B05,E_B06,E_B07,E_B08,A_W0,A_W1,E_W4,E_W5,A_B,C_D,E_F,A_C,C_E.
```

Abbildung 73: Aufbau [333830]

Die Bedeutung jedes Elements:

333830	Klasse.
A_B01	Entfernung von A nach Balise B01.
A_B02	Entfernung von A nach Balise B02.
A_B03	Entfernung von A nach Balise B03.
C_B04	Entfernung von C nach Balise B04.
C_B05	Entfernung von C nach Balise B05.
E_B06	Entfernung von E nach Balise B06.
E_B07	Entfernung von E nach Balise B07.
E_B08	Entfernung von E nach Balise B08.
A_W0	Entfernung von A nach Weiche W0.
A_W1	Entfernung von A nach Weiche W1.
E_W4	Entfernung von E nach W4.
E_W5	Entfernung von E nach W5.
A_B	Entfernung von A nach B.
C_D	Entfernung von C nach D.
E_F	Entfernung von E nach F.
A_C	Entfernung von A nach C.
C_E	Entfernung von C nach E.

Tabelle 20: Die Bedeutung von jedem Element (13)

Gleismodule mit einer bestimmten Langer

Aufgrund einiger Module, deren Lange gemessen wurde, ist keine CFG-Datei erforderlich. Sie konnen direkt angerufen werden. Die Module mit fester Lange wird unten gezeigt.

Klasse: 110000

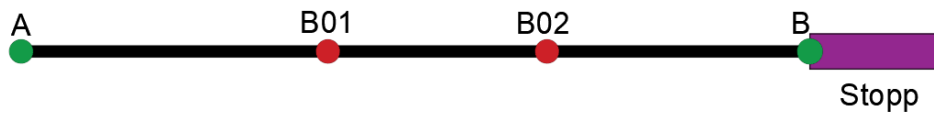


Abbildung 74: Gleismodul [110000]

Abstand:

Nummer des Bretts	A_B01	B01_B02	A_B
1, 4, 5, 8	100mm	275mm	900mm
3	125mm	275mm	900mm
7	100mm	250mm	900mm

Tabelle 21: Abstand [110000]

Eine Klasse kann mehrere Bretter haben. Man kann verschiedene Bretter uber CSV-Dateien aufrufen.

Klasse: 111030



Abbildung 75: Gleismodul [111030]

Abstand:

Nummer des Bretts	A_B01	B01_B02	B02_B03	B03_B
1 bis 9	125mm	725mm	825mm	125mm

Tabelle 22: Abstand [111030]

Klasse: 111010



Abbildung 76: Gleismodul [111010]

Abstand: A_B = 600mm

Klasse: 111080

Klasse 111080 hat 8 Ausrichtungen.

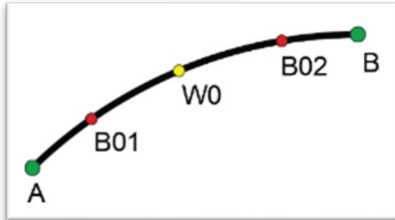


Abbildung 77: Ausrichtung 1

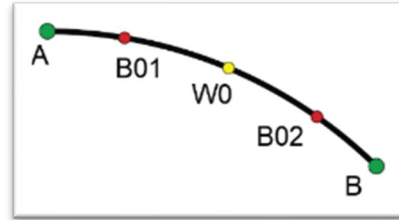


Abbildung 78: Ausrichtung 2

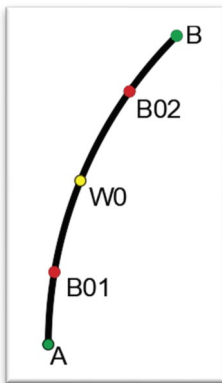


Abbildung 79: Ausrichtung 5

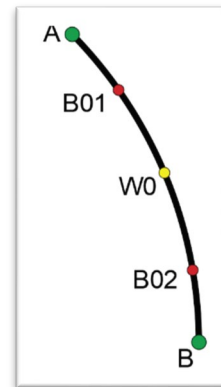


Abbildung 80: Ausrichtung 6

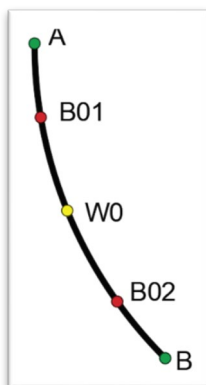


Abbildung 81: Ausrichtung 7

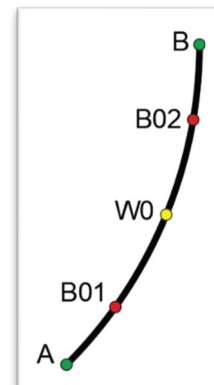


Abbildung 82: Ausrichtung 8

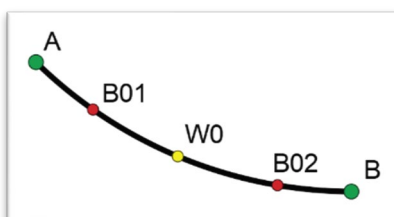


Abbildung 83: Ausrichtung 3

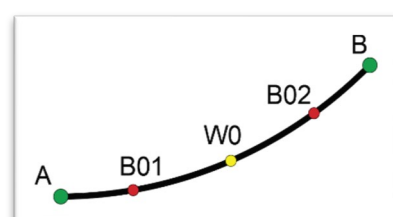


Abbildung 84: Ausrichtung 4

Eine Zahl von 1 bis 8 wird ausgewählt, und dann wird sie in die vierte Spalte der CSV-Datei eingegeben, um die Ausrichtung des Moduls 111080 zu bestimmen.

NR des Bretts	A_B01	B01_W0	W0_B02	B03_B
1, 9, 10	105mm	365mm	335mm	135mm
2 bis 8	135mm	335mm	335mm	135mm

Tabelle 23: Abstand [110080] [20]

Klasse: 221120

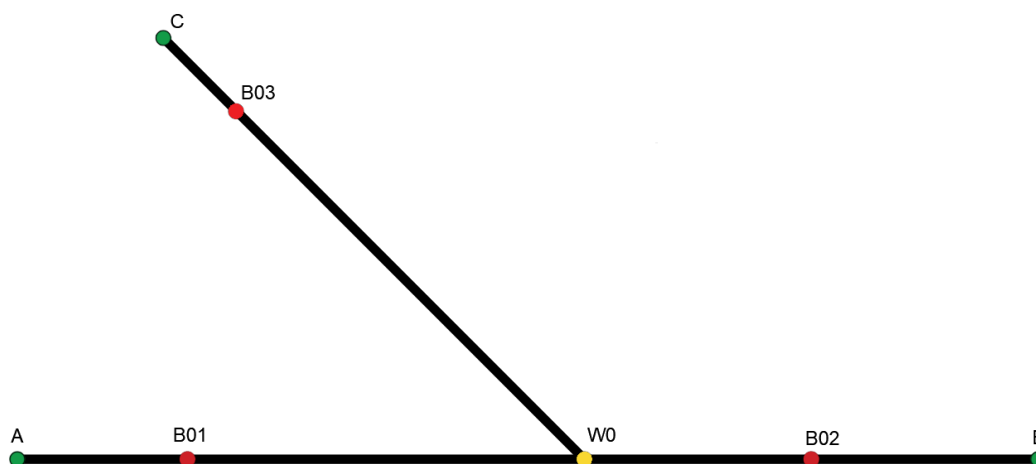
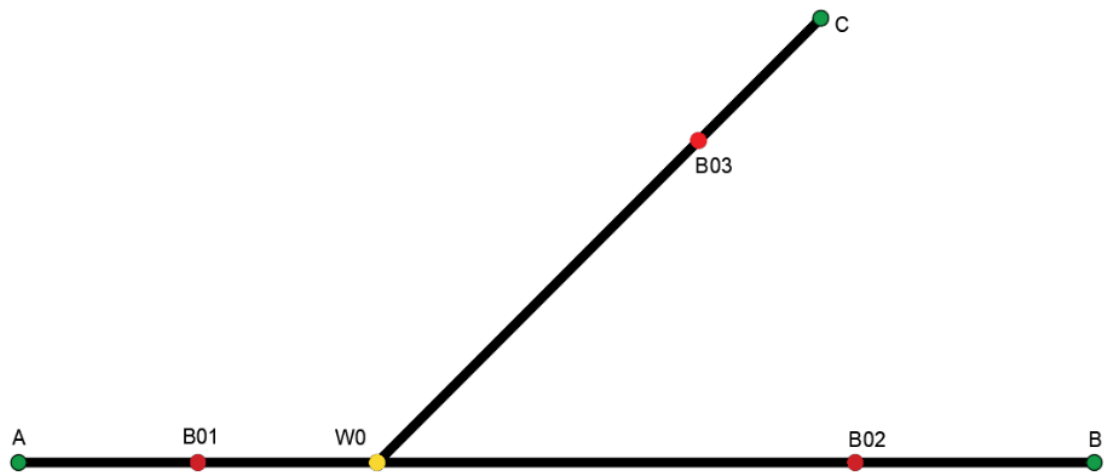


Abbildung 85: Gleismodul [221120]

Abstand von Klasse 221120:

Abstand	W0	A	B	C
B01	735mm	125mm	1075mm	/
B02	215mm	1075mm	125mm	/
B03	825mm	/	/	145mm

Tabelle 24: Abstand [221120]

Klasse : 221121**Abbildung 86: Gleismodul [221121]**

Abstand von Klasse 221121:

Abstand	W0	A	B	C
B01	215mm	125mm	1075mm	/
B02	735mm	1075mm	125mm	/
B03	825mm	/	/	145mm

Tabelle 25: Abstand [221121]

Darstellung ausgewählter Gleispläne im Umfeldmodell

Beispiel 1:

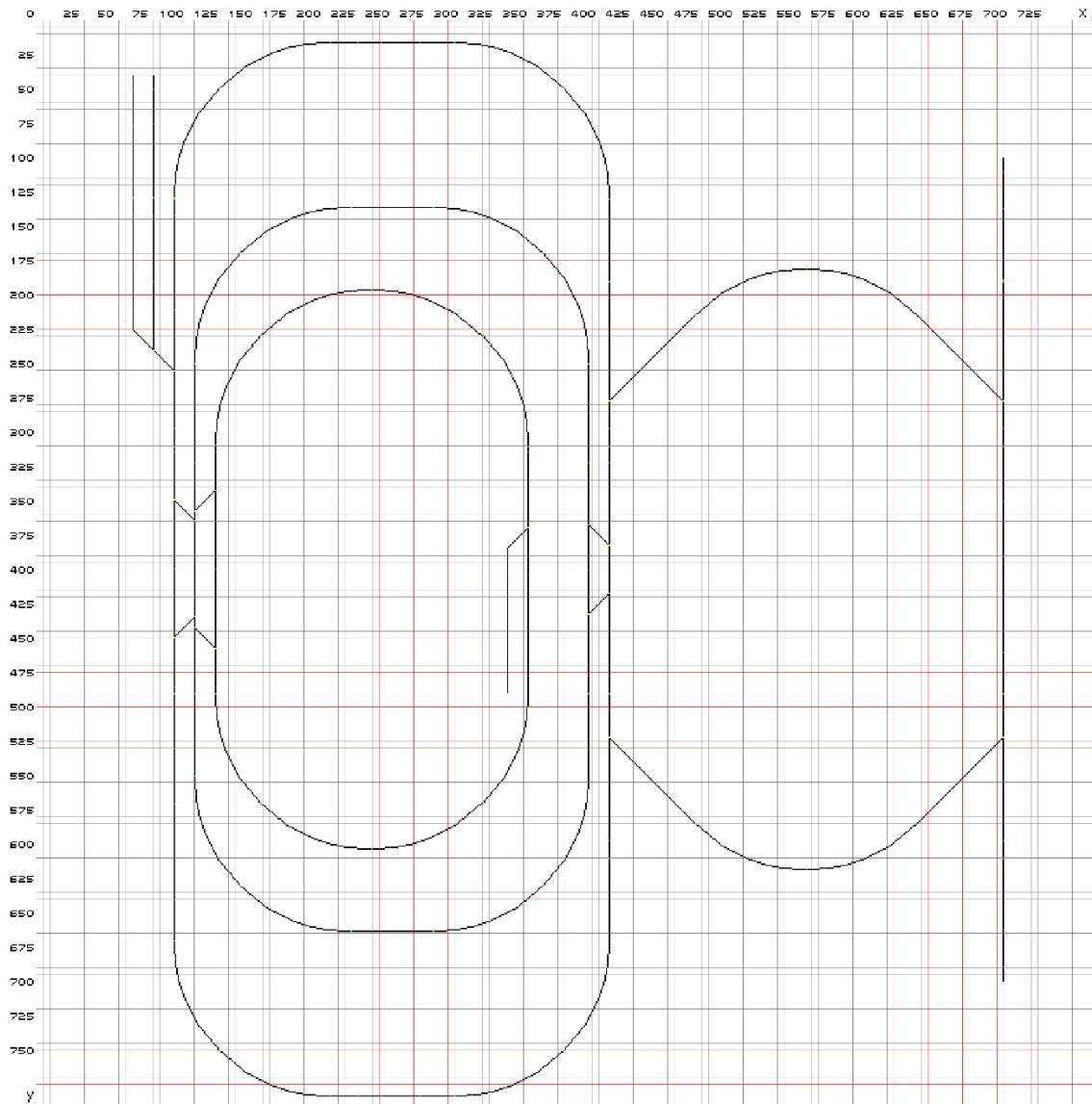


Abbildung 87: Geschlossenes Gleissystem

333830	100	310	4	1
111080	130	309	5	1
111080	163	229	1	1
111080	244	196	2	1
111080	325	229	6	1
221130	358	310	4	1
111080	130	490	7	1
111080	163	570	3	1
111080	244	603	4	1
111080	325	570	8	1
111010	115	250	4	1
111080	115	249	5	1
111080	148	169	1	1
111010	229	136	1	1
111080	289	136	2	1
111080	369	169	6	1
111010	402	250	3	1
222430	417	310	3	1
111010	115	490	3	1
111080	115	550	7	1
111080	148	630	3	1
111010	229	663	1	1
111080	289	663	4	1
111080	369	630	8	1
111010	402	490	4	1
331231	100	130	3	1
221120	417	190	3	1
111010	417	130	4	1
221121	417	490	3	1
111010	417	610	3	1
111050	100	490	3	1
110000	70	129	4	1
110000	85	129	4	7
111080	100	129	5	1
111080	133	49	1	1
111060	214	16	1	1
111080	304	16	2	1
111080	384	49	6	1
111080	100	670	7	1
111080	133	750	3	1
111060	214	783	1	1
111080	304	783	4	1
111080	384	750	8	1
111080	480	214	1	1
111080	561	181	2	1
221121	704	190	4	1
111080	480	585	3	1
111080	561	618	4	1
221120	704	490	4	1
111050	704	310	3	1
110000	704	189	4	1
110000	704	610	3	1

Tabelle 26: CSV-Datei vom Geschlossenen Gleissystem

333830,40,900,1750,125,1675,40,900,1750,400,1400,325,1475,1800,1800,1800,160,160.
 221130,125,860,125,860,1755,1200,1800,160.
 222430,125,1675,125,900,1675,1800,1800,720,1080,160.
 331231,125,810,1675,125,600,125,500,1250,1800,160,160.
 111050,400,800,1500,1800.
 111060,200,375,655,900.

Abbildung 88: CFG-Datei vom Geschlossenen Gleissystem

Beispiel 2:

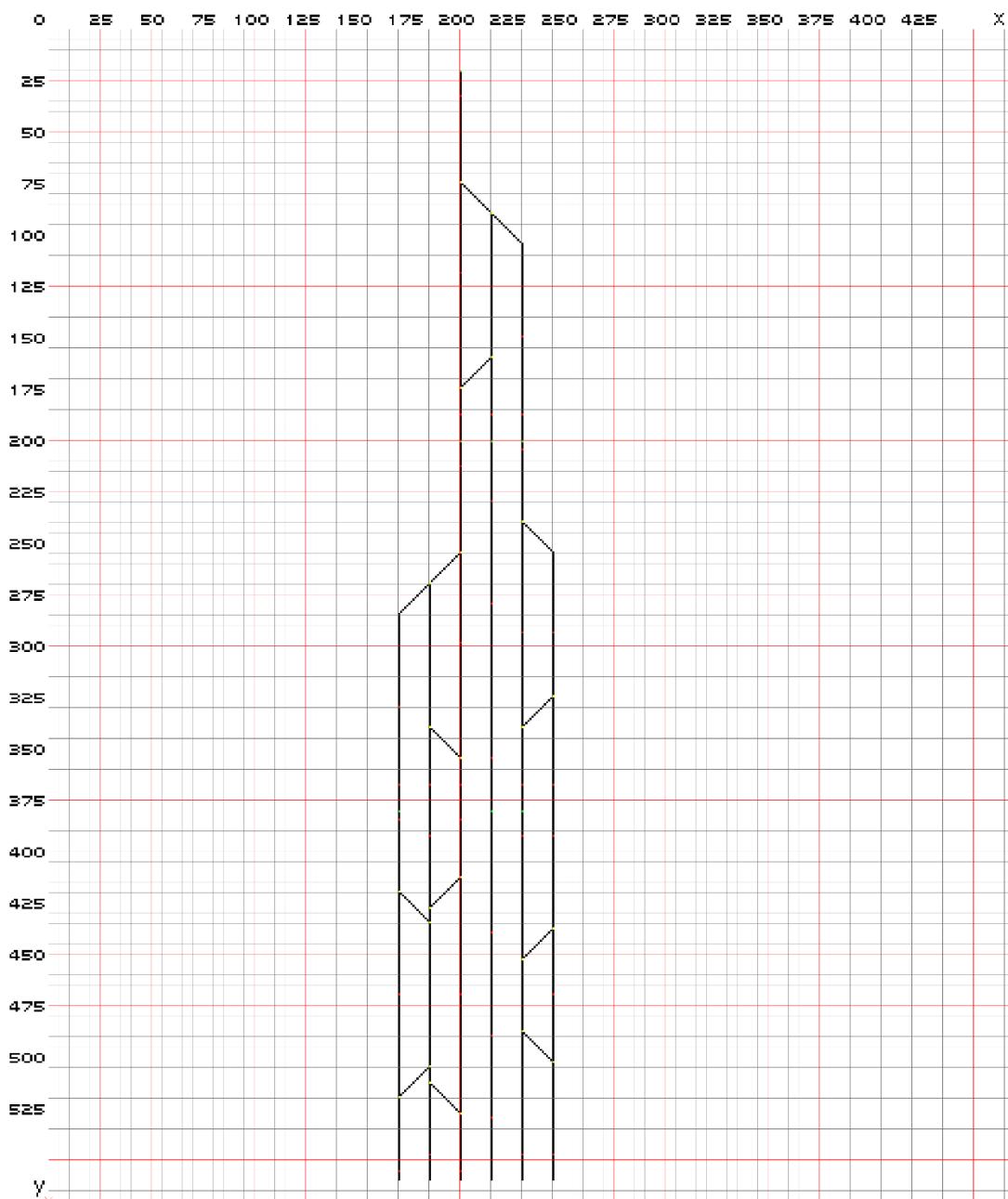


Abbildung 89: Lineares System

331431	200	20	4	1
331430	200	200	4	1
221231	230	200	3	1
222430	230	380	4	1
333830	170	380	4	1
111053	215	200	4	1
111063	215	380	4	1

Tabelle 27: CSV-Datei vom Linearen System

331431,125,810,1675,125,125,500,250,1250,1800,160,160.
 331430,125,810,1675,125,125,500,250,1250,1800,160,160.
 221231,125,860,125,860,1755,1400,400,1800,160.
 222430,125,1675,125,900,1675,1800,1800,720,1080,160.
 333830,40,900,1750,125,1675,40,900,1750,400,1400,325,1475,1800,1800,1800,160,160.
 111053,250,1000,1500,1800.
 111063,300,700,1200,1800.

Abbildung 90: CFG-Datei vom Linearen System

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Vorname Nachname