

DIPLOMARBEIT

Fahrsimulator Elektrolokomotive ÖBB 1042

Ausgeführt im Schuljahr 2012/13 von:

Sebastian Nestinger 5AHETE-10
Lukas Rudolf 5AHETE-13

Betreuer/Betreuerin:

Dipl.-Ing. Helge Frank

St. Pölsen, am 14.03.2013

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Verfasser / Verfasserinnen:

Sebastian Nestinger

Lukas Rudolf

St. Pölten, am 13.05.2013

Danksagung/Widmungen

Zu Beginn möchten wir uns bei all denjenigen bedanken, die uns während des Schuljahres unterstützt haben. Dabei gilt unserem Betreuer Herrn Dipl.-Ing. Helge Frank ein besonderer Dank.

Desweiteren möchten wir uns bei Herrn Ing. Markus Schinko von der Firma Cargo Service GmbH für die Unterstützung im Rahmen einer Kooperationsvereinbarung bedanken. Ohne dieses Sponsoring wäre dieses Projekt in der Form nicht realisierbar gewesen.

Zwecks Informationseinholung besuchten wir Anfang des Schuljahres die ÖBB Zentralschule Wörth um den Loksimulator der Baureihe 1042 zu besichtigen. Dabei bedanken wir uns recht herzlich bei Herrn Christian Haslinger und Herrn Josef Ratzinger von den ÖBB, die uns dies ermöglicht haben.

Herr Prof. Widmann gestattete uns den Fahrsimulator im Labor zu hinterstellen und die Arbeiten dort durchzuführen. Auch dafür einen herzlichen Dank.

Schlussendlich bedanken wir uns bei den Werkstättenlehrern, die uns stets mit Rat und Tat zur Seite standen. Im Speziellen seien Herr Dipl. Päd. Ing. Hermann Meiseneder und Herr Ing. Markus Chmelar erwähnt, die uns Bauteile und Material zur Verfügung stellten. Herr Dipl. Päd. Franz Büchinger und Herr FOL Ing. Walter Kapeller unterstützten uns bei mechanischen Arbeiten.

Zusammenfassung

Die Diplomarbeit umfasst den Aufbau eines Fahrsimulators, welcher die Tätigkeit eines Lokführers in verschiedensten Situationen äußerst realitätsnah darstellt. Der Fahrsimulator besteht aus den originalen Teilen des Führerstandes der Elektrolok ÖBB 1042 44. Diese einzeln gekauften Führerstandsteile werden mit einem geschweißten Stahlgerüst vorbildgerecht angeordnet, sodass ein Bedienpult entsteht. Alle Handlungen, die der Lokführer auf diesem Bedienpult setzt, gelangen in analoger oder digitaler Form in den Arduino MEGA-Prozessor. Dieser berechnet brauchbare Werte und überträgt Bytes über den virtuellen Serial Port an das Visual Basic-Programm am Computer. Dort werden die Daten weiterverarbeitet und schlussendlich an die Zugsimulationssoftware „ZUSI“ übergeben. Bei Meldelampen oder Messinstrumenten erfolgt der Übertragungsweg sinngemäß in umgekehrter Reihenfolge. ZUSI läuft am angeschlossenen Flachbildschirm im Vollbildmodus, um dem Benutzer ein möglichst realistisches Fahrgefühl zu vermitteln. Notwendig waren neben der komplexen Verkabelung der Teile auch 3 Platinen, die sich wie der Arduino MEGA-Prozessor und das Netzteil im Schaltkasten befinden. Der Fahrsimulator besitzt neben der 230V~ Spannungsversorgung auch einen USB Anschluss zum Arduino-MEGA und ein eigenes Lautsprechersystem mit Subwoofer. Der Fahrsimulator schafft Einblicke in die komplexen Vorgänge des täglichen Bahnbetriebs auf verschiedenen Fahrtrouten und ermöglicht am Beruf Lokführer Interessierten ihre Fähigkeiten in diesem Beruf zu testen.

Abstract

This diploma thesis contains of a driving simulator for a electric locomotive "1042" of the austrian federal railways. It is built out of parts from the original loco. These are held together by a welded steel framework which keeps them in the prototypical position. The driver of the virtual locomotive operates the switches and levers on the desk and his actions are recognised by a "Arduino MEGA" microprocessor. This information gets transfered to a program which runs on the PC all the time, which sends the commands to the "Zusi" train simulation software. "Zusi" also emits data, which gets transfered to the PC program and afterwards to the Arduino. The Arduino sends the data to the 3 pcb boards who actuate the indicators on the desk. This allows the user to get a very realistic feeling for the locomotive, because the operation behaves just like it does in the original locomotive. Overall, the simulator gives people the opportunity to test their skills in driving a train.

Inhaltsverzeichnis

1	Das Vorbild	1
1.1	Eisenbahntechnik allgemein	1
1.1.1	Grundsätzliches	1
1.1.2	Bremssysteme	1
1.1.2.1	Handbremse.....	1
1.1.2.2	Direkte Bremse (Lokbremse).....	2
1.1.2.3	Indirekte Bremse (Zugbremse).....	4
1.1.3	Signalsysteme	5
1.1.3.1	Signalbilder ÖBB.....	5
1.1.3.1.1	Streckensignale.....	5
1.1.3.1.2	Ersatzsignale	6
1.1.3.1.3	Schutzsignale	7
1.1.3.1.4	Formsignale.....	8
1.1.3.1.5	Geschwindigkeitstafeln.....	9
1.1.3.1.6	Geschwindigkeitsanzeiger.....	10
1.1.3.1.7	Langsamfahrtsignale (LA).....	10
1.1.3.2	Signalbilder DB.....	11
1.1.3.2.1	Streckensignale.....	11
1.1.3.2.2	Ersatzsignal	12
1.1.3.2.3	Formsignale.....	12
1.1.3.2.4	Geschwindigkeitstafeln.....	13
1.1.3.2.5	Geschwindigkeitsanzeiger.....	14
1.1.3.2.6	Langsamfahrtsignale.....	14
1.1.3.2.7	Haltetafel	15
1.1.3.3	Praxisbeispiele.....	15
1.1.4	Zugsicherungssysteme	17
1.1.4.1	Sicherheitsfahrerschaltung.....	17
1.1.4.2	Induktive Zugsicherung	18
1.1.5	Zugsammelschiene.....	19
1.2	Die Baureihe ÖBB 1042/1142.....	20
1.2.1	Geschichte	20
1.2.1.1	Technische Daten	22
2	Mechanische Arbeiten	22
2.1	Führerstandsteile.....	22
2.1.1	Beschreibung der Teile.....	23
2.1.1.1	Schalter/Meldergruppe links.....	23
2.1.1.1.1	Instandsetzung	23
2.1.1.1.2	Beschreibung der Schalterfunktionen	24
2.1.1.2	Fahrschaltereinheit	26
2.1.1.3	Tachograph.....	28
2.1.1.4	Führerbremssventil.....	29
2.1.1.5	Zusatzbremssventil	30
2.1.1.6	Zugfunkgerät KAPSCH ZFM90.....	31
2.2	Stahlgerüst.....	32
3	Digitale Datenübertragung.....	33
3.1	Dateneingabe.....	33
3.1.1	Übertragungsweg der Dateneingabe	33
3.1.2	Erfassung der Zustände am Pult.....	34

3.1.2.1	Kippschalter/Taster	34
3.1.2.1.1	Beschaltung der Eingänge	34
3.1.2.1.2	Liste der Digitaleingänge	35
3.1.2.2	Erfassung der Analogwerte.....	36
3.1.2.2.1	Fahrschalter	36
3.1.2.2.2	Führerbremssventil	36
3.2	Datenausgabe	37
3.2.1	Übertragungsweg der Datenausgabe	37
3.2.2	Ausgabeinstrumente.....	38
3.2.2.1	Leuchtmelder	38
3.2.2.1.1	Allgemeines.....	38
3.2.2.1.2	Indusi 1000 Hz.....	38
3.2.2.1.3	Indusi P70-85	38
3.2.2.1.4	Indusi D95-105.....	39
3.2.2.1.5	Hauptschalter aus	39
3.2.2.1.6	Ölpumpe steht	39
3.2.2.1.7	Lok steht.....	39
3.2.2.1.8	Elektrische Bremse	40
3.2.2.2	Zeigerinstrumente	40
3.2.2.2.1	Motorstromanzeigen	40
3.2.2.2.2	Bremsstromanzeigen	41
3.2.2.2.3	Hauptluftleitungs- und Hauptluftbehälterdruck	42
3.2.2.2.4	Stufenanzeige	43
3.3	Schaltplan	44
3.3.1	Schematische Darstellung	44
3.3.2	Schaltplan	45
3.4	Platinen.....	46
3.4.1	PWM Platine	46
3.4.1.1	Schaltplan.....	46
3.4.1.2	Funktionsprinzip	47
3.4.1.3	Entflechtung.....	47
3.4.1.4	Bestückung	48
3.4.1.5	Anschlussbelegung.....	48
3.4.2	Relaiskarte	49
3.4.2.1	Schaltplan.....	49
3.4.2.2	Funktionsprinzip	49
3.4.2.3	Entflechtung.....	50
3.4.2.4	Bestückung	50
3.4.2.5	Anschlussbelegung.....	51
3.5	Vollständige Verkabelung.....	52
4	Software	54
4.1	Zugsimulation "ZUSI"	54
4.2	Übertragungssoftware	57
4.2.1	Programm am PC.....	57
4.2.1.1	Programmoberfläche	57
4.2.1.1.1	Verbindungsteil	58
4.2.1.1.2	Überwachungsteil	59
4.2.1.1.3	Überwachungs- und Debugteil	60
4.2.2	Programm am Arduino.....	61
5	Inbetriebsetzung und Betrieb	62
5.1	Aufbau und mechanischer Test.....	62
5.2	Starten der Programme	62

5.3	Fahren eines Zuges.....	67
5.3.1	Praxissituationen anhand von Beispielen	67
5.3.2	Österreichische Signaltechnik	67
6	Wirtschaftlicher Teil	72
6.1	Arbeitsaufteilung	72
6.2	Kostenaufstellung	73

1 Das Vorbild

1.1 Eisenbahntechnik allgemein

1.1.1 Grundsätzliches

In Österreich und Deutschland gibt es neben den normalspurigen Eisenbahnlinien, welche einen Schienenabstand von 1435 mm aufweisen auch noch Schmalspurbahnen. Diese Variante mit 760 mm Schienenabstand ist vor Allem bei Gebirgsbahnen zu finden. In der Schweiz haben die Schmalspurbahnen im Gebirge einen Schienenabstand von 1000 mm, wobei hier teils Zahnstangen verwendet werden.

Die elektrifizierten Eisenbahnen in den genannten Ländern werden üblicherweise mit einer Fahrdrathspannung von 15kV Einphasenwechselfspannung und einer Frequenz von 16 2/3 Hz versorgt. Selbstverständlich gibt es Ausnahmen wie zum Beispiel die Mariazellerbahn, die mit 6500 V und 25 Hz betrieben wird.

1.1.2 Bremssysteme

In der Eisenbahntechnik kommen für verschiedene Zwecke mehrere Bremssysteme parallel zum Einsatz. Ein gewisses Maß an Grundwissen wie die verschiedenen Bremssysteme aufgebaut sind ist notwendig, um einen Zug sicher führen zu können und effizientes Fahren zu ermöglichen. Im Folgenden sind die auf der Vorbildlok verbauten Bremssysteme kurz erklärt.

1.1.2.1 Handbremse

Die Handbremse oder auch Feststellbremse genannt, bezeichnet dem Namen entsprechend eine rein mechanische Bremse auf der Lok, welche manuell über eine Kurbel am Führerstand bedient wird und die Lokomotive auch bei nicht vorhandener Druckluftversorgung am Wegrollen hindert. Diese Kurbel befindet sich auf Führerstand 1 und wird vor Fahrtantritt, sobald die Druckluftbremse einsatzbereit ist, gelöst, und beim Abstellen der Lok wieder angelegt. Auf moderneren Loks wurde sie durch die sogenannte Federspeicherbremse ersetzt.



Abbildung 01: Kurbel zur Betätigung der Handbremse

1.1.2.2 Direkte Bremse (Lokbremse)

Die direkte Bremse, auch Zusatzbremse genannt, umfasst ausschließlich die Bremsen an der Lokomotive selbst. Es handelt sich hierbei um eine Druckluftbremse, welche vom zentralen Druckluftversorgungssystem der Lok gespeist wird. Die Bremsbeläge werden mithilfe von 2 Bremszylindern gegen die Laufflächen der 4 Achsen gepresst und erzeugen somit eine Bremswirkung. Die Bremse ist über das Zusatzbremsventil (im folgenden auch als "ZbV" bezeichnet) am Führerstand regelbar. Auf der 1042 ist dieses vom Hersteller "Oerlikon" gefertigt und von der Type "FDS2".



Abbildung 02: Oerlikon FDS2 Zusatzbremsventil

Dieses Ventil besitzt folgende Stellungen zur Bedienung der direkten Bremse:

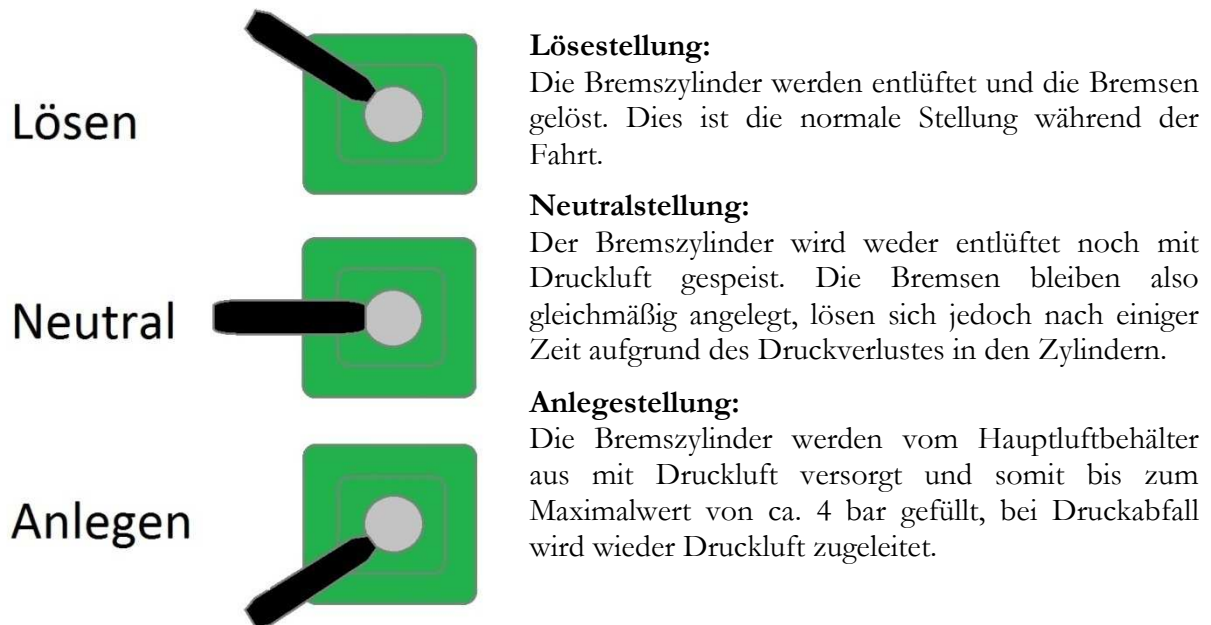
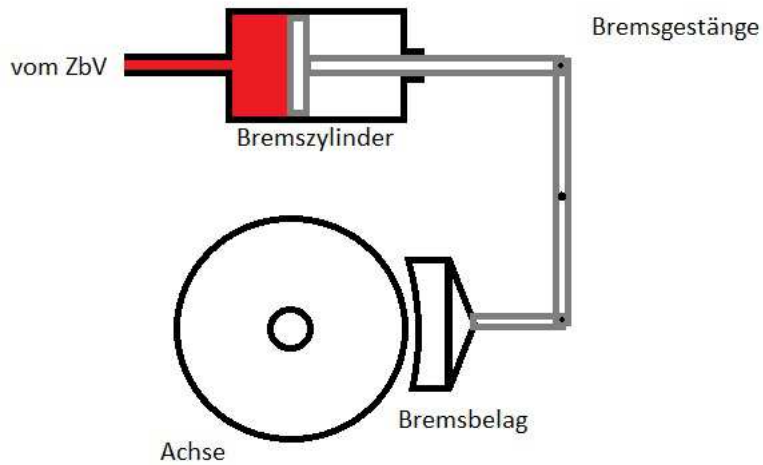


Abbildung 03: Stellungen Zusatzbremsventil

**Wirkungsweise der direkten Bremse:**

Durch eine Erhöhung des Luftdrucks im Zylinder wird das Bremsgestänge bewegt und der Bremsklotz samt Bremsbelag gegen die Lauffläche der Achse gepresst, die Lokomotive bremst.

Abbildung 04: Bremsanlage

1.1.2.3 Indirekte Bremse (Zugbremse)

Als indirekte Bremse wird das heute übliche System für die Bremsen des gesamten Zuges bezeichnet. Dabei sind alle Waggons des Zuges über eine Druckluftleitung, die sogenannte Hauptluftleitung, verbunden. Diese übernimmt dabei sowohl eine Speise- als auch eine Regelfunktion. Im normalen Zustand sind die Bremsen gelöst und der Druck in der Hauptluftleitung beträgt 5 bar. An den Wagen befinden sich die sogenannten Steuerventile, welche in diesem Zustand die Druckluft direkt in die Hilfsluftbehälter, welche ebenfalls an jedem Wagen vorhanden sind, weiterleiten. Sind diese Hilfsluftbehälter gefüllt, ist die Bremsanlage einsatzbereit.

Soll der Zug nun gebremst werden, zieht der Triebfahrzeugführer das Führerbremssventil an und senkt so den Druck in der Hauptluftleitung. Die Steuerventile an den Wagen nehmen diese Druckabsenkung wahr und leiten nun einen Teil der Druckluft, proportional der Höhe der Absenkung in der Hauptluftleitung, vom Hilfsluftbehälter in die Bremszylinder des Waggons. Das bedeutet also je stärker der Druck in der Hauptluftleitung abgesenkt wird, umso größer ist auch die Bremswirkung die dadurch erzielt wird.

Beim teilweisen oder vollständigen Lösen der Bremsen wird die überschüssige Druckluft in den Bremszylindern der Waggons ins Freie entlassen und die Bremswirkung wird geringer.

Ein Nachteil dieses Systems ist, dass die Hilfsluftbehälter nur im gelösten Zustand aufgeladen werden. Der Lokführer sollte es also vermeiden, die Zugbremse oftmals hintereinander anzuziehen und wieder zu lösen, da somit die Hilfsluftbehälter mit der Zeit geleert werden und die Bremswirkung nachlässt. Vor Allem bei längeren Gefällefahrten muss daher eine spezielle Technik des Bremsens angewendet werden, das sogenannte "Sägezahnbremsen". Dies bedeutet dass eine Zeit lang die Bremsen stärker als nötig angezogen werden, der Zug also ca. 10 bis 20km/h unter die Höchstgeschwindigkeit abgebremst wird, und anschließend wieder eine Periode lang mit gelösten Bremsen gerollt wird. Dies verhindert ein Nachlassen der Bremswirkung, sowie ein Überhitzen der Bremsanlage.



Abbildung 05: Bremsschläuche zwischen Lok und Waggon

1.1.3 Signalsysteme







Zu den wichtigsten Merkmalen des Bahnbetriebs gehören zweifelsohne die Signale. Diese zeigen an, ob und mit welcher Geschwindigkeit ein Gleisabschnitt befahren werden darf. Grundsätzlich wird zwischen Vorsignal und Hauptsignal unterschieden, wobei das Vorsignal jenes Signalbild ankündigt, welches beim Hauptsignal zu erwarten ist. Die ab dem Hauptsignal geltende Maximalgeschwindigkeit muss im gesamten Weichenbereich eingehalten werden, außer ein Geschwindigkeitsanzeiger legt eine neue Geschwindigkeit fest.

Hauptsignale erfüllen verschiedene Zwecke und werden daher auch unterschiedlich bezeichnet. Beispielsweise wird ein Hauptsignal, welches vor einem Bahnhof steht, und somit die Einfahrt der Züge regelt, Einfahrtssignal genannt. Weiters gibt es Ausfahrtsignale, Schutzsignale, welche in einem Bahnhof die Gleise in Abschnitte unterteilen und Blocksignale, welche die freie Strecke in Blockabschnitte unterteilen.

Da sich die Signalbilder in Deutschland von jenen in Österreich unterscheiden wird nachfolgend auf beide Ausführungen eingegangen. Es werden jedoch nur die wichtigsten Signalbilder und Tafeln behandelt, welche in ZUSI auftreten. Hier würde eine komplette Fassung jeden Rahmen sprengen.

1.1.3.1 Signalbilder ÖBB

1.1.3.1.1 Streckensignale

Vorsignal	Signalnachahmer	Hauptsignal	Bedeutung
			Halt Falls das Hauptsignal in einer unübersichtlichen Kurve steht, wird ein Nachahmer aufgestellt.
			Frei mit Streckenhöchstgeschwindigkeit

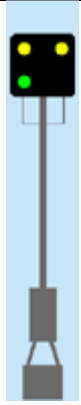
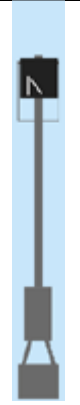
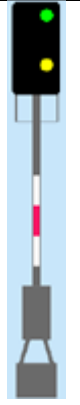

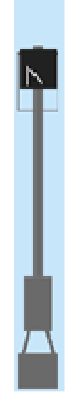

			<p>Frei mit 40km/h</p> <p>Der Nachahmer zeigt nur an, dass eine Geschwindigkeitsbeschränkung am Hauptsignal zu erwarten ist, nicht aber welche.</p>
			<p>Frei mit 60km/h</p> <p>Der Nachahmer zeigt nur an, dass eine Geschwindigkeitsbeschränkung am Hauptsignal zu erwarten ist, nicht aber welche.</p>

Tabelle 01: Österreichische Signale

1.1.3.1.2 Ersatzsignale

Falls das Stellwerkssystem gestört oder das Signal untauglich ist, kann der Fahrdienstleiter ein Ersatzsignal stellen.


Signalbild	Bedeutung
	Ein weiß blinkendes Licht erlaubt die Vorbeifahrt an einem HALT – zeigenden Hauptsignal mit maximal 40km/h.

Tabelle 02: Ersatzsignal Österreich

1.1.3.1.3 Schutzsignale

Schutzsignale teilen ein Gleis am Bahnhof in 2 oder mehrere Gleisabschnitte, welche von mehreren Zügen befahren werden können. So ist es zum Beispiel möglich, dass auf Bahnsteig 2A ein Zug steht, während gleichzeitig auf Bahnsteig 2B ein anderer Zug einfährt. Charakteristisch für ein Schutzsignal ist die weiße Umrandung.



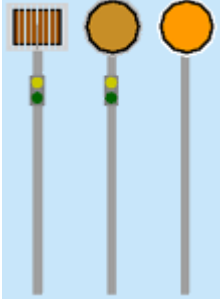
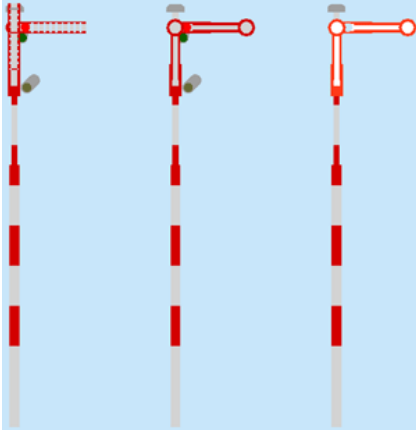
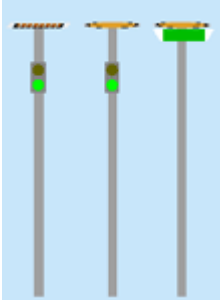
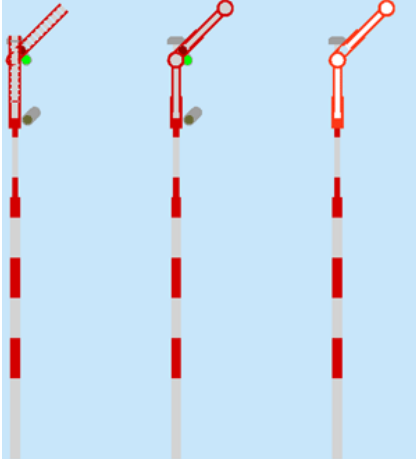
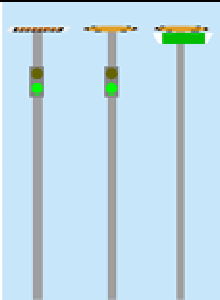
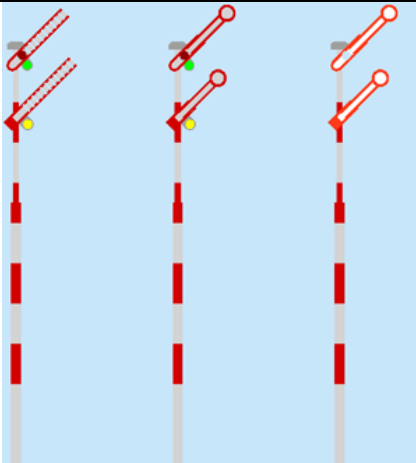
Signalbild	Bedeutung
	Fahrverbot
	Fahrverbot aufgehoben.

Tabelle 03: Schutzsignale Österreich

1.1.3.1.4 Formsignale

Auf manchen Eisenbahnstrecken in Österreich sowie in den ZUSI Strecken stehen noch Formsignale. Daher werden auch diese Signale kurz beschrieben.

Vorsignal	Hauptsignal	Bedeutung
		Halt
		Frei mit Streckenhöchstgeschwindigkeit
		Frei mit 40km/h Bei einer außerplanmäßigen Abzweigung wird das Vorsignal auf „Vorsicht“ gestellt.


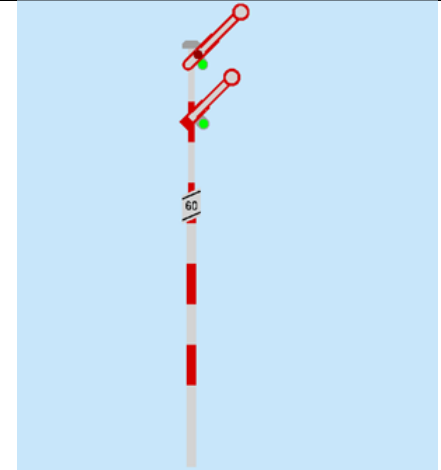
		<p>Frei mit 60km/h Seit 1980 abgeschafft.</p>
---	---	---

Tabelle 04: Formsignale Österreich

1.1.3.1.5 Geschwindigkeitstafeln

Geschwindigkeitstafeln informieren den Lokführer über die aktuell gültige Streckenhöchstgeschwindigkeit. Eine dreieckförmige Tafel kündigt an, welche Geschwindigkeit ab der viereckigen Tafel gilt. Angeschrieben wird immer 1/10 der tatsächlichen Geschwindigkeit.

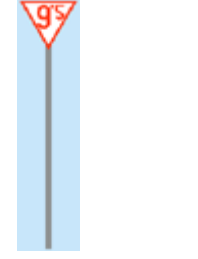
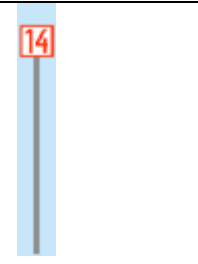
Tafel	Bedeutung
	<p>Ankündigungstafel 95km/h Diese werden nur bei einer Geschwindigkeitsreduktion von mehr als 20% aufgestellt. Bei einer Ankündigung <90 km/h ist PZB Wachsam zu betätigen.</p>
	<p>Geschwindigkeitstafel 140km/h</p>

Tabelle 05: Geschwindigkeitstafeln Österreich

1.1.3.1.6 Geschwindigkeitsanzeiger

Diese werden an Vor- und Hauptsignalen aufgestellt um eine andere Abzweiggeschwindigkeit als 40 oder 60km/h anzuzeigen. Wie bei den Tafeln wird auch bei den Geschwindigkeitsanzeigen 1/10 der tatsächlichen Geschwindigkeit angezeigt.



Anzeige	Bedeutung
	Voranzeiger 80km/h Bei einer Ankündigung ≤ 90 km/h ist PZB Wachsam zu betätigen.
	Anzeiger 100km/h

Tabelle 06: Geschwindigkeitsanzeiger Österreich

1.1.3.1.7 Langsamfahrsignale (LA)

Befinden sich auf einem Streckenabschnitt Bauarbeiten oder eine schlechte Gleislage werden für den jeweiligen Zeitraum Langsamfahrstellen gebildet. Sie sind meistens nur einige hundert Meter lang.

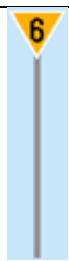


Tafel	Bedeutung
	Ankündigungssignal 60km/h
	Anfangssignal 120km/h
	Endsignal

Tabelle 07: Langsamfahrstellen Österreich

1.1.3.2 Signalbilder DB

1.1.3.2.1 Streckensignale

Vorsignal	Hauptsignal			Bedeutung
	Einfahrtsignal	Ausfahrtsignal	Blocksignal	
				Halt (Vr0/Hp0)
				Fahrt (Vr1/Hp1)
				Fahrt mit 40km/h (Vr2/Hp2)

Tabelle 08: Lichtsignale Deutschland

1.1.3.2.2 Ersatzsignal

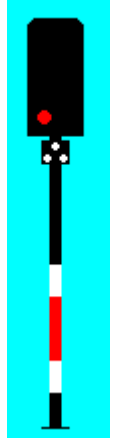
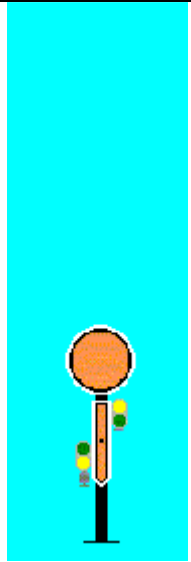
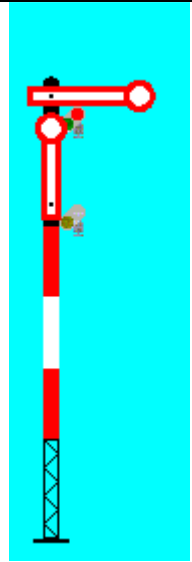
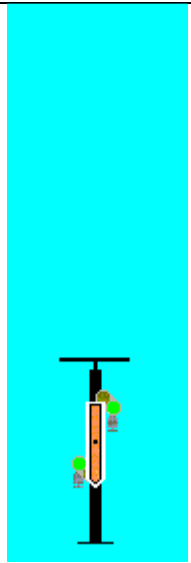
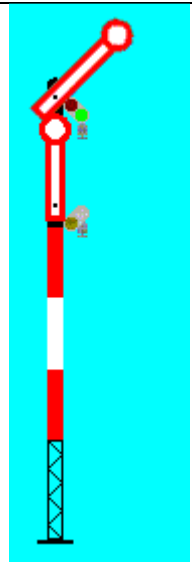
Signalbild	Bedeutung
	<p>3 weiße Lichter erlauben dem Lokführer mit gedrückter PZB Befehlstaste und maximal 40 km/h das „Halt“ zeigende Signal zu passieren.</p>

Tabelle 09: Ersatzsignal Deutschland

1.1.3.2.3 Formsignale

Vorsignal	Hauptsignal	Bedeutung
		<p>Halt (Vr0/Hp0)</p>
		<p>Fahrt mit Streckenhöchstgeschwindigkeit(Vr1/Hp1)</p>


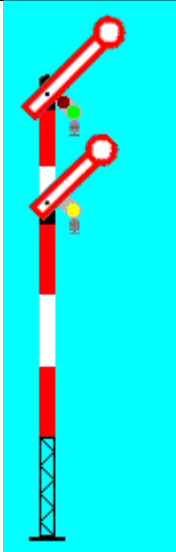
		<p>Fahrt mit 40km/h (Vr2/Hp2)</p>
---	---	-----------------------------------

Tabelle 10: Formsignale Deutschland

1.1.3.2.4 Geschwindigkeitstafeln

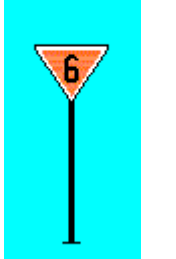
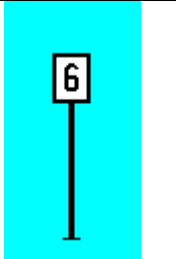
Tafel	Bedeutung
	<p>Ankündigungstafel 60km/h</p>
	<p>Geschwindigkeitstafel 60km/h Ist die nachfolgende Streckengeschwindigkeit höher als die derzeit gültige, wird keine Tafel aufgestellt. Der Lokführer muss dies im Buchfahrplan ablesen.</p>

Tabelle 11: Geschwindigkeitstafeln Deutschland

1.1.3.2.5 Geschwindigkeitsanzeiger

Da sich mit den Signalgrundbegriffen nur Streckenhöchstgeschwindigkeit und 40km/h signalisieren lassen, werden für alle anderen Geschwindigkeiten Anzeiger verwendet. Bis einschließlich 60km/h zeigt das Signal den Grundbegriff Hp2, darüber Hp1.

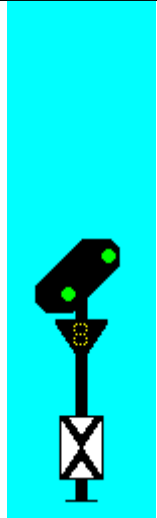
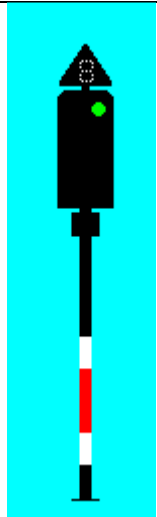
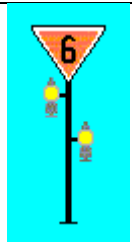
Anzeige	Bedeutung
	Voranzeiger 80km/h Bei einer Ankündigung ≤ 90 km/h ist PZB Wachsam zu betätigen.
	Anzeiger 80km/h

Tabelle 12: Geschwindigkeitsanzeiger Deutschland

1.1.3.2.6 Langsamfahrsignale

Tafel	Bedeutung
	Ankündigungssignal 60km/h Im Gegensatz zu ständigen Geschwindigkeitsbeschränkungen stehen unter der Tafel 2 zusätzliche gelbe Lichter.

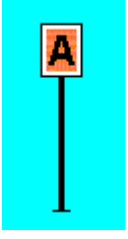
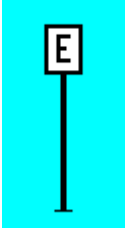
	Anfangssignal
	Endsignal

Tabelle 13: Langsamfahrstelle Deutschland

1.1.3.2.7 Haltetafel

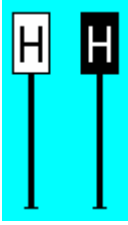
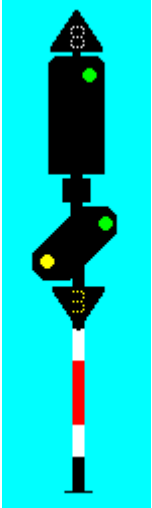
Tafel	Bedeutung
	Die Haltetafel zeigt an, wo die Lokomotive eines planmäßigen Zuges am Bahnsteig zu halten hat. Abhängig von örtlichen Gegebenheiten ist die Tafel schwarz oder weiß ausgeführt.

Tabelle 14: Haltetafel

1.1.3.3 Praxisbeispiele

In der Realität kommt es häufig vor, dass mehrere Signale kombiniert aufgestellt werden.

Signalbild	Bedeutung
	<p>DB Signal: Hauptsignal zeigt „frei“, Vorsignal zeigt „Vorsicht“.</p> <p>Diese Signalkombination darf also mit 80km/h passiert werden und ab dem nächsten Hauptsignal gilt eine Höchstgeschwindigkeit von 30km/h.</p>

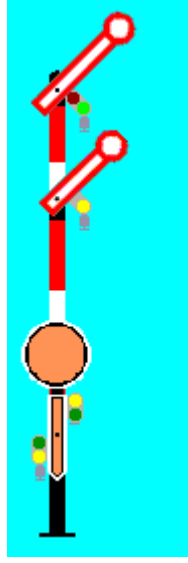
	<p>DB Signal:</p> <p>Hauptsignal zeigt „Frei mit 40km/h“, Vorsignal zeigt „Vorsicht“</p> <p>Dieses Signal darf mit 40km/h passiert werden und das nächste Hauptsignal zeigt „Halt“.</p>
---	---

Tabelle 15: Praxisbeispiele Signale

1.1.4 Zugsicherungssysteme

Damit auf der Eisenbahn ein sicherer Betrieb herrscht, gibt es verschiedene Einrichtungen der Zugsicherung, welche die Beachtung von Signalen und Geschwindigkeitsbeschränkungen, sowie die Aufmerksamkeit des Triebfahrzeugführers sicherstellen.

1.1.4.1 Sicherheitsfahrschaltung

Die Sicherheitsfahrschaltung (kurz: SIFA) ist eine Sicherheitseinrichtung in Triebfahrzeugen. Der Lokführer muss das Sifa Pedal (Totmannknopf) während der Fahrt gedrückt halten und in regelmäßigen Abständen loslassen. Geschieht dies nicht wird eine Schnellbremsung eingeleitet. In Österreich und Deutschland ist das Sifa Pedal am gebräuchlichsten, in anderen Ländern gibt es noch Drucktaster und Zugbügel. In den österreichischen und deutschen Lokomotiven sind verschiedene Arten der Sifa eingebaut:

Zeit-Zeit Sifa

Hier hält der Lokführer das Pedal gerückt und lässt es mindestens alle 30 Sekunden kurz los. Dadurch erhält die Sifa die Rückmeldung dass der Lokführer noch reaktionsfähig ist. Ist er nicht in der Lage es loszulassen setzt ein akustisches Signal ein, welches ihn daran erinnert. Lässt er es immer noch nicht los, setzt eine Zwangsbremung ein.

Zeit-Weg Sifa

Bei dieser Variante wird nicht nur die Zeit sondern auch der zurückgelegte Weg seit der letzten Betätigung berücksichtigt. Der Vorteil liegt darin, dass im Stillstand die Sifa nicht betätigt werden muss.

Weg-Weg Sifa

Hier muss der Lokführer nach 900m das Sifa Pedal kurz loslassen, unabhängig von der Geschwindigkeit mit der er fährt. Tut er dies nicht ertönt nach 75m ein akustisches Signal, welches ihn daran erinnert. Falls der Lokführer auch jetzt noch nicht loslässt setzt nach weiteren 75m eine Schnellbremsung ein. Diese Variante ist bei den ÖBB Lokomotiven am gebräuchlichsten und wird auch in der 1042 verwendet.



Abbildung 06: Sifa Pedal auf der Bodenplatte

1.1.4.2 Induktive Zugsicherung

Die sogenannte induktive Zugsicherung gehört zur Gattung der punktförmigen Zugbeeinflussungssysteme und sorgt dafür, dass sich der Zug sicher auf den Bahngleisen bewegt. Es wird überwacht, mit welcher Geschwindigkeit sich ein Zug einem Bereich annähert, in dem mit verminderter Geschwindigkeit zu fahren ist, oder wenn ein Halt bevorsteht.

Bereits in den 1930er Jahren machte man sich Gedanken, wie der Bahnverkehr sicherer gemacht werden könnte und entwickelte ab 1931 die induktive Zugsicherung, welche ab 1935 bis heute in Deutschland und Österreich Standard ist.



Abbildung 07: Indusi Fahrzeugmagnet an einer Lokomotive (Quelle: S. Nestinger)

Die Übertragung bei der Indusi findet mithilfe induktiver Schwingkreise verschiedener Frequenzen statt. Dafür gibt es im Wesentlichen 2 Bauteile, den Fahrzeugmagneten auf der Lok, ein aktiver Schwingkreis, sowie den Gleismagneten, welcher als passiver Schwingkreis ausgeführt ist. Überfährt die Lok einen solchen Schwingkreis, bricht die Spannung im Oszillator des Triebfahrzeuges aufgrund der Resonanz ein, wodurch eine Informationsübertragung stattfindet. Soll keine Beeinflussung stattfinden, so wird der gleisseitige Schwingkreis kurzgeschlossen und somit unwirksam - es findet also keine Beeinflussung am Fahrzeug statt. Da diese 2 Informationszustände nicht ausreichend sind für eine effektive Überwachung, gibt die Gleismagneten mit verschiedenen Resonanzfrequenzen von 500, 1000 und 2000Hz, welche verschiedene Bedeutungen haben.

1.1.5 Zugsammelschiene

Als Zugsammelschiene wird ein Kabel bezeichnet, das durch den ganzen Zug verläuft. Es versorgt alle Wagen mit Strom vom Transformator der Lokomotive. Da früher nur das Licht und die Heizung der Wagen über die Zugsammelschiene versorgt wurden, wird diese auch noch Heizleitung genannt. Heute verlaufen weitere Datenleitungen über die Zugsammelschiene, zum Beispiel Türsteuerung und Wendezugsteuerung. An jedem Wagenende befindet sich ein Stecker sowie eine Buchse für die Leitung. Die Spannung beträgt heute üblicherweise 1000V 16Hz ~ und kann mit bis zu 600A (Belastungsgrenze der Steckverbindungen) belastet werden. Da heutige Wagen mit leistungsstarken Klimageräten oft noch höhere Ströme aufnehmen, werden bei besonders langen Zügen beide Steckverbindungen genutzt, was die Belastungsgrenze erhöht.



Tabelle 16: Verbindung der Zugsammelschiene (vorne)

1.2 Die Baureihe ÖBB 1042/1142

1.2.1 Geschichte

Der Fahrsimulator besteht aus den originalen Teilen des Führerstandes der Elektrolok ÖBB 1042 44. Die Österreichischen Bundesbahnen beschafften von 1963 bis 1976 360 Loks dieser Baureihe um der fortschreitenden Elektrifizierung des österreichischen Eisenbahnnetzes Rechnung zu tragen. Die anfänglich universell eingesetzten Loks bespannten sowohl leichte und schwere Güterzüge, als auch innerösterreichische Regional-, Schnell- und Eilzüge. Ebenso waren sie später vor Eurocity und Intercity Zügen anzutreffen. In den 1990er Jahren wurden die meisten Loks dieser Baureihe wendezugtauglich modernisiert und auf 1142 umnummeriert. Im Zuge dieses Umbaus wurden die Eckfenster und 2 Türen entfernt und die Loks im Valousek-Design neu lackiert. Heute bespannen die Loks der Baureihe 1142 Regional- und Regionalexpresszüge, sowie Schnellzüge und Güterzüge und sind nach wie vor unverzichtbar. Einige Loks der Baureihe 1042 wurden für Nostalgie- und Sonderfahrten in die Ursprungslackierung zurückversetzt.



Abbildung 08: 1042 44 in den 1960er Jahren (Quelle: Digitales Eisenbahnfotoarchiv)



Abbildung 09: Die modernisierte 1142 598 in St. Pölten Hbf 2007 (Quelle: L. Rudolf)



Abbildung 10: Die modernisierte 1142 596 am 10.08.2012 (Quelle: S. Nestinger)



Abbildung 11: Der Führerstand einer Lok der Reihe 1142 (Quelle: bahnvideos.net)



Abbildung 12: 1142 680 in Tulln im August 2011 (Quelle: S. Nestinger)

1.2.1.1 Technische Daten

Leistung	3700 kW
Geschwindigkeit	130/150 km/h
Anfahrzugkraft	225 kN
Gewicht	84t
Länge	16,22m
Baujahre	1963-1973
Stückzahl	60
Stromsystem	15kV , 16 $\frac{2}{3}$ Hz
Zugheizung	600 kVA
Zugbeeinflussungsart	Indusi 60
Sicherheitsfahrerschaltung	Weg-Weg Sifa
Hersteller	SGP/ELIN

Tabelle 17: Technische Daten ÖBB 1042

2 Mechanische Arbeiten

2.1 Führerstandsteile



Abbildung 13: Übersicht der mechanischen Bauteile

2.1.1 Beschreibung der Teile

2.1.1.1 Schalter/Meldergruppe links

Auf diesem Baugruppenträger befinden sich die Schaltelemente für die Steuerung der Führerstandsheizung, der Zugsammelschiene (Zugheizung), sowie die Leuchtmelder und Bedienelemente für die Indusi.

2.1.1.1.1 Instandsetzung

Da sich dieses Bauelement in schlechtem Zustand befand, war eine Instandsetzung von Nöten. Die alte Farbe wurde entfernt und durch eine neue Farbschicht ersetzt, die Schalter und Knöpfe gereinigt, sowie auf Funktion geprüft. Die 3 Leuchtmelder mussten vollständig ersetzt werden.



Abbildung 14: Schaltergruppe vor Instandsetzung



Abbildung 15: Schaltergruppe nach Instandsetzung

2.1.1.1.2 Beschreibung der Schalterfunktionen

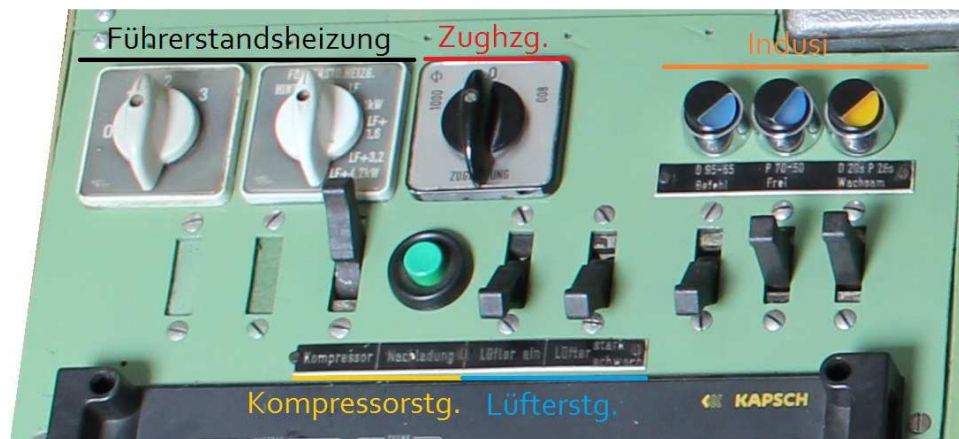


Abbildung 16: Beschreibung der Schalterfunktionen

Drehschalter 1, 2:

Wahlschalter für die Führerstandsheizung (keine Funktion im Simulator)

Drehschalter 3:

Wahlschalter für die Zugheizung (Zugsammelschiene), welche die Waggons mit Energie versorgt. (Näheres dazu im Kapitel: Vorbild - keine Funktion im Simulator)

Schalterstellungen des Drehschalters „Zugheizung“:

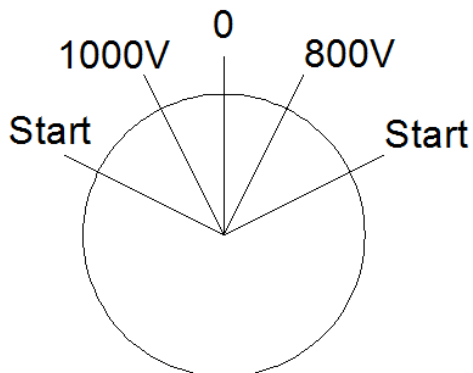


Abbildung 17: Schalterstellungen Zugheizung

Kompressor Aktivierung:

Kippschalter rastend, dient zum Einschalten des Kompressors im Automatikbetrieb. (keine Funktion im Simulator)

KompressorNachladung:

Drücktaster, dient zum Forcieren einer Aufladung des Hauptluftbehälters auf Maximallevel. (keine Funktion im Simulator)

Näheres zur Funktion der Bremsanlage im Kapitel 1.1.2

Lüfter Aktivierung:

Kippschalter rastend, dient zum Einschalten des Lüfters zur Kühlung der Fahrmotoren und der elektrischen Einrichtungen.

Lüfter Stufenauswahl:

Kippschalter rastend, dient zum Umschalten zwischen starker und schwacher Belüftung. (keine Funktion im Simulator)

Näheres zu den elektrischen Einrichtungen im Allgemeinen im Kapitel 1.1**Leuchtmelder IndusiZugart D:**

Dient zur Anzeige der Zugart D.

Leuchtmelder IndusiZugart P:

Dient zur Anzeige der Zugart P.

Leuchtmelder Indusi1000Hz:

Dient zur Anzeige einer 1000Hz Beeinflussung.

Schalter Indusi Befehl:

Kippschalter rastend, dient zum Umgehen einer 2000Hz Beeinflussung bei entsprechender Erlaubnis.

Schalter Indusi Frei:

Kippschalter tastend, dient zur Befreiung aus einer Geschwindigkeitsüberwachung bei Wegfall der nächstkommenden Geschwindigkeitsreduktion unter oder gleich 90km/h.

Schalter Indusi Wachsam:

Kippschalter tastend, dient zur Bestätigung der Wahrnehmung einer Geschwindigkeitsankündigung von unter oder gleich 90km/h.

Näheres zur Indusi im Kapitel 1.1.4

2.1.1.2 Fahrschaltereinheit



Abbildung 18: Fahrschaltereinheit mit Schaltwalzen und Stufenanzeige

Fahrschalter

In der Vorbildlok ist am Gestänge des Fahrschalters ein Winkelschrittgeber montiert, der sowohl die elektrische Bremsung als auch die Fahrstufen erkennt. Dieser wurde ausgebaut und durch ein Potentiometer ersetzt. Da jedoch der Fahrschalter eine wesentlich größere Auslenkung als ein Potentiometer aufweist, wurden zusätzlich 2 Zahnräder mit einem Übersetzungsverhältnis von 3:2 eingebaut. Das Potentiometer wird mit 5V und Masse versorgt, der Schleifer führt als Datenleitung zum Arduino MEGA. Desweiteren musste eine Kalibrierung durchgeführt werden, sodass die Position des Schleifers mit der Stufe des Fahrschalters optimal zusammenstimmt.

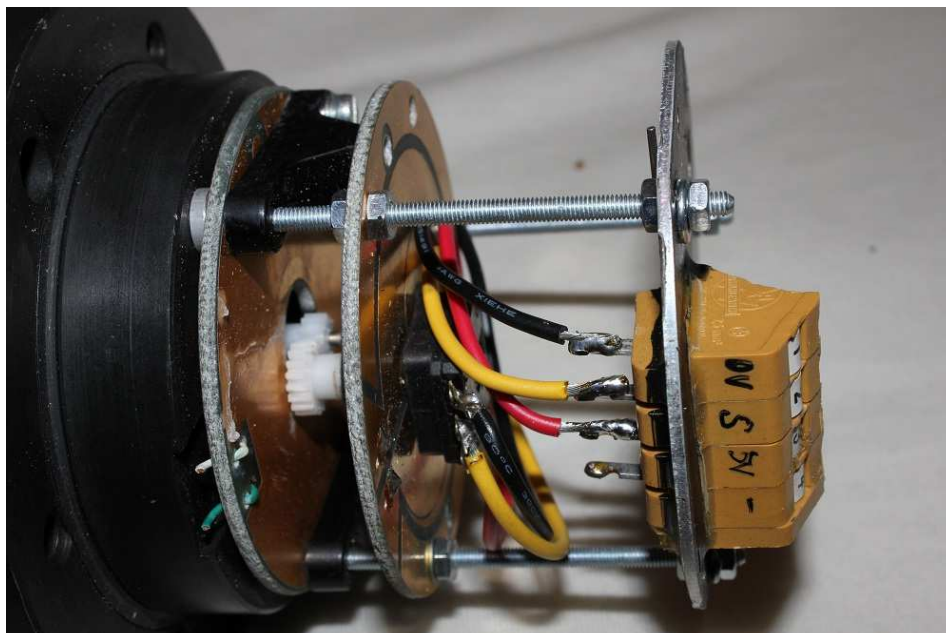


Abbildung 19: Potentiometer mit Zahnrädern am Gestänge des Fahrschalters

Hauptschalter

Wird der Hauptschalter nach oben „getastet“, wird mechanisch ein Schließer betätigt der eine Datenleitung zum Arduino geschaltet. Wird nach unten „getastet“, schaltet ein Öffner eine weitere Datenleitung.

Richtungswender

Der Richtungswender hat im Programm keine Funktion, da ZUSI nur Zugfahrten und keine Rangierfahrten unterstützt. Jedoch lässt sich der Richtungswender nicht nach V oder R schalten, wenn auf eine Fahrstufe aufgeschaltet wurde oder wenn sich der Stromabnehmerschalter auf 0 befindet. Es ist also beim Auf- und Abrüsten der Lokomotive immer auf die korrekte Vorgehensweise zu achten.

Stromabnehmer

Der Stromabnehmerschalter weist 3 mögliche Stellungen auf: 0, Tief und Hoch. Dieser Schalter ist in 0-Stellung abnehmbar und wird beim Richtungswechsel am anderen Führerstand der Lok wieder benötigt. In „Tief“-Stellung wird eine 5V Leitung geschaltet, sodass das Batteriespannungsanzeigeelement 24V anzeigt. In „Hoch“-Stellung wird eine Datenleitung geschaltet, welche zum Arduino MEGA führt.

Vorgehensweise beim Aufrüsten:

Stromabnehmer 0-T-H

Richtungswender 0-V

Hauptschalter EIN tasten

2.1.1.3 Tachograph

Der Tachometer aus schweizer Erzeugung (Fa. Hasler AG) funktioniert nach mechanischem Prinzip und wird von einem eigenen Motor angetrieben, welcher sich auf der Unterseite befindet und mit einer Kupplung zur Welle der Tachometermechanik verbunden ist. Aufgrund der einfacheren Ansteuerbarkeit wurde der Originalmotor, eine 24V 3~ Synchronmaschine, durch einen permanenterregten Gleichstrommotor mit einem 1:5 Planetengetriebe ersetzt. Dieser wird wie die Stromanzeigen über einen der LowSide - Kanäle der PWM - Platine angesteuert. Die Versorgungsspannung beträgt 12V, wobei im Betrieb eine Drehzahl von ca. 650 U/min erreicht wird. Der maximale Geschwindigkeitswert des Tachometers beträgt 180 km/h bei 800 U/min, aufgrund der Höchstgeschwindigkeit der Lok werden aber maximal 150km/h angezeigt.



Abbildung 20: Hasler Tachograph

2.1.1.4 Führerbremventil

In das Führerbremventil wurde genauso wie in den Fahrshalter ein Potentiometer eingebaut. Dabei wurde die Achse des Bremsventils mit einer aufwendigen Konstruktion mit der des Potentiometers verbunden. Für die Ausführung der Kabel musste ein eigenes Loch gebohrt werden. Gleich wie beim Fahrshalter wird auch hier das Potentiometer mit 5V und Masse versorgt und der Schleifer führt als Datenleitung zum Arduino MEGA. Der Fahrshalter und das Führerbremventil sind die einzigen Analogeingänge des Arduino MEGA.



Abbildung 21: Führerbremventil am Fahr Simulator

2.1.1.5 Zusatzbremsventil

Da das Zusatzbremsventil in der Vorbildlok rechts an der Führerstandswand montiert ist, musste ein geeignetes Befestigungsgestell geschweißt werden. Das Bremsventil wurde an das Gestell angeschraubt und dieses wiederum an den Fahrsimulator geschraubt. Somit wurde die gleiche Position für die Bremse geschaffen, wie man sie auch in der Vorbildlok vorfindet.

Da das Zusatzbremsventil im Wesentlichen nur drei Stellungen hat und im Zugsimulationsprogramm nur die beiden Stellungen „Lösen“ und „Anlegen“ relevant sind, wurden zwei Taster eingebaut. Diese befinden sich in Aussparungen einer Pressspanplatte an der Unterseite der Bremse. Wird die Bremse in Lösestellung gebracht, so fährt unten aus der Bremse ein mechanischer Stab aus, der einen Taster betätigt. In Anlegestellung wird der jeweils andere Taster betätigt, wodurch die beiden Zustände an das Programm übertragen werden können. (Näheres dazu im Kapitel „Erfassung der Zustände am Pult“.)



Abbildung 22: Zusatzbremsventil am Fahrsimulator

2.1.1.6 Zugfunkgerät KAPSCH ZFM90

Der Zugfunk ermöglicht die Kommunikation zwischen dem Lokführer und den Betriebsstellen entlang der Strecke (zb: Fahrdienstleiter, Verschieber). Dabei handelt es sich um Gespräche zwischen Mitarbeitern, Aufträge an den Lokführer und Meldungen vom Lokführer. Der Fahr Simulator enthält so wie auch die Vorbildlok ein analoges UIC Zugfunkgerät der Type ZFM 90 von Kapsch.

Leider war es nicht machbar, das Zugfunkgerät so zu adaptieren, dass eine Bedienung des Displays über das Bedienpaneel möglich ist. Dafür hätte man die gesamte Schaltung der Zugfunkanlage der Lok mit sämtlichen Bauteilen, welche nicht zur Verfügung standen, nachbilden müssen. Dieses Vorhaben war mit dem vorhandenen Budget bei weitem nicht realisierbar. Es wurde also die gesamte Elektronik aus dem Gerät ausgebaut und eine 7 Segmentanzeige mit passender Spannungsversorgung eingebaut. Wird der Fahr Simulator eingeschaltet, so zeigt das Display des Zugfunkgerätes rein aus optischen Gründen nun immer folgende Werte an:



Abbildung 23: Grundbild des analogen Zugfunkgerätes ZFM 90

Die folgende Tabelle beschreibt die wichtigsten Eingaben:

Befehlsart	Anzeige	Bedeutung
Auftrag	BEF	Der Lokführer fährt mit Befehl. zb: A-Befehl für Langsamfahrstelle
Zugnr.	7526	Zugnummer des Zuges (vgl. Buchfahrplan)
Meldung	--	Der Lokführer kann eine Meldung an den Fahrdienstleiter richten
BTR1	A72	Betriebsart 1: Funk für Zugfahrten
BTR2	C17	Betriebsart 2: Funk für Verschiebfahrten

Tabelle 18: Beschreibung der Anzeige des Zugfunks

2.2 Stahlgerüst

Um die Teile des Führerstandes wie im Vorbild anzuordnen, musste ein Gestell angefertigt werden, welches die Teile in den vorbildgerechten Positionen trägt. Dieses wurde aus 20x20mm Stahlprofilen mit 1mm Stärke geschweißt. Um den Transport zu erleichtern wurden auf der Unterseite 6 Rollen angebracht.



Abbildung 24: Das Stahlgerüst zu einem frühen Fertigungszeitpunkt (20.07.2012)

Die Abdeckung erfolgte mit Pressspanplatten, welche im Originalfarbton lackiert wurden.

3 Digitale Datenübertragung

Im Folgenden werden die prinzipielle Wirkungsweise der Erfassung der Daten und die Art der Übertragung dieser erläutert. Von der Erfassung der Zustände der Hardware bis zum Einlesen dieser in die Software werden zahlreiche Stellen durchlaufen, welche die Daten in abgewandelter Form weiterleiten. Damit die Daten auf den Anzeigen korrekt erscheinen sind verschiedene Anpassungen von Nöten.

3.1 Dateneingabe

Die Dateneingabe umfasst das Feststellen der Zustände der mechanischen Bedienelemente am Pult, bis hin zur Eingabe der daraus resultierenden Ereignisse an das Simulationsprogramm.

3.1.1 Übertragungsweg der Dateneingabe



Abbildung 25: Übertragungsweg der Dateneingabe

Ausgangspunkt bei der Erfassung der Daten sind die Zustände der Schalter, Taster und Regler am Fahrpult. Der Arduino übernimmt das Einlesen der Analog- und Digitalwerte und berechnet daraus brauchbare Werte zur Übertragung. Übertragen als Bytes über den virtuellen Serial Port des Arduino, werden diese vom Visual Basic Programm im PC aufgenommen und weiterverarbeitet. Schlussendlich erfolgt die Weitergabe an Zusi, was über virtuelle Tastendrucke geschieht. Um eine verlässliche Verbindung zum Simulator zu gewährleisten, werden einige Werte anschließend noch einmal ausgelesen und mit dem Sollzustand verglichen.

3.1.2 Erfassung der Zustände am Pult

3.1.2.1 Kippschalter/Taster

Ein großer Teil der Eingaben auf dem Pult erfolgt durch das Betätigen von Schaltern und Tastern. Die Zustände dieser werden vom Arduino erkannt und in weiterer Instanz zum PC weitergegeben. (Näheres dazu im Kapitel „Übertragung der Daten zum PC“.)

3.1.2.1.1 Beschaltung der Eingänge

Die Zustände der Schalter und Taster werden über den Arduino Mega erfasst. Dazu wurden im Programm die internen Pullup - Widerstände des Arduino aktiviert, welche das Potential der Datenleitung auf 5V anheben. Die Schalter schalten gegen Masse, was vom Arduino als Tastendruck erkannt wird. Ein Potential von 5V am Eingang bedeutet bei einem Schließer also „Nicht betätigt“, während 0V mit „Betätigt“ gleichzusetzen ist. Bei einem Öffner ist das Gegenteil der Fall.

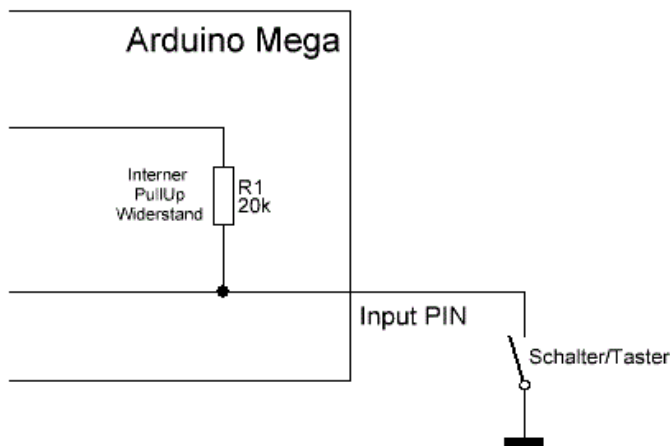


Abbildung 26: Beschaltung der digitalen Eingänge

3.1.2.1.2 Liste der Digitaleingänge

Insgesamt werden 15 digitale Eingänge eingelesen. Im Folgenden sind diese aufgelistet:

Eingang am Arduino	Klemmen - Nr. im Schaltkasten	Funktion
24	6	Lüfter
25	7	PZB Befehl
26	8	PZB Frei (Öffner)
27	9	PZB Wachsam (Öffner)
28	10	HS ein
29	11	HS aus (Öffner)
30	12	Horn
31	13	Sand
32	14	Stromabnehmer
33	15	SIFA
34	16	Lokbremse
35	17	Lokbremse
36	18	unbelegt
37	19	unbelegt
38	20	unbelegt
39	21	unbelegt

Tabelle 19: Liste Digitaleingänge

3.1.2.2 Erfassung der Analogwerte

Vom Arduino werden aus dem Pult 2 Analogwerte eingelesen. Diese repräsentieren die Stellungen des Fahrschalters und des Führerbremssventils. Für die Adaptierung der mechanischen Stellung auf ein elektrisches Signal wurden Potentiometer mit 20k verwendet, welche mit 0V und 5V versorgt werden, während der Schleifer zum Analogeingang des Arduino verdrahtet ist, welcher dort ein Signal von 0 bis 5V einliest und mittels eingebautem ADC in einen Digitalwert umwandelt.

3.1.2.2.1 Fahrschalter

Der Fahrschalter ist als Kombinationshandrad ausgeführt und steuert sowohl die Fahrstufen als auch die elektrische Bremse. Die Neutralstellung befindet sich auf Fahrstufe 0. Wird der Fahrschalter nach rechts gedreht, so befindet er sich im "Fahren" - Bereich, und eine der 34 Fahrstufen kann ausgewählt werden. Die 34 Fahrstufen sind über einen Winkel von 315° angeordnet, also eine 7/8 Umdrehung. Wird der Fahrschalter nach links gedreht, befindet er sich im "Bremsen" - Bereich, welcher bis auf die Initialisierungsstufe ohne Rasten ausgeführt ist. Der Drehwinkel nach links beträgt 135°, also eine 3/8 Umdrehung. Umgerechnet auf den Stufenabstand im Fahrbereich wären dies also 14 Bremsstufen. Da in Zusi nur mit äquidistanten Stufen gearbeitet wird, sind diese 14 Bremsstufen in den Eigenschaften der Lok eingetragen. Es ergeben sich insgesamt also 14 Bremsstufen + 34 Fahrstufen + Neutralstellung also 49 Sollstellungen für den Fahrschalter. Damit von dem eingelesenen Analogwert (A15), welcher von 0 bis 1024 reicht, auf diese Stellungen geschlossen werden kann, wurde im Arduino-Programm ein Array eingefügt, welches jedem dieser 1024 Werte genau einen Fahrschaltersollwert zuordnet. Dieser wird dann an den PC übertragen. (Siehe Arduino bzw. VB-Programm.)

Am PC-Programm wird der eingelesene Sollwert mit dem aus Zusi ausgelesenen Istwert verglichen. Ist eine Abweichung feststellbar, wird ein entsprechender Tastendruck für "Fahrschalter auf" oder "Fahrschalter ab" an das Simulationsprogramm gesendet, bis der Istwert gleich dem Sollwert ist.

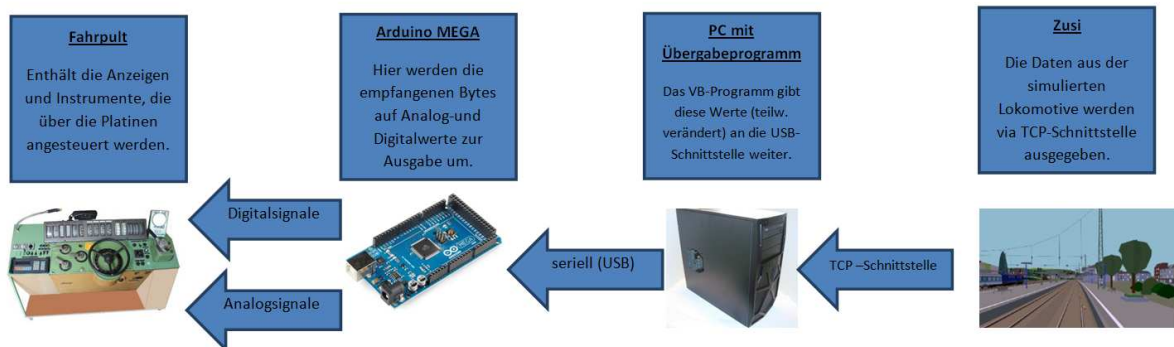
3.1.2.2.2 Führerbremssventil

Die Position des Führerbremssventils wird auch im Bremsventil über ein Potentiometer, dessen Schleifer an einen Analogeingang des Arduino geführt ist (A14). Im Programm wird dieser eingelesene Wert auf einen Soll - Hauptluftleitungsdruck umgerechnet. Dieser wird nun an das PC-Programm übertragen und der Ist-Hauptluftleitungsdruck aus Zusi ausgelesen. Wird eine Abweichung festgestellt, wird das Zusi - Bremsventil (welches als Auf/Ab Ventil definiert ist) in die entsprechende Stellung zum Erhöhen oder Verringern des Hauptluftleitungsdrucks gebracht. Entspricht der Istwert bis auf $\pm 0,05$ bar dem Sollwert, wird das Ventil in Abschlusstellung gebracht (Druck wird gehalten).

3.2 Datenausgabe

Die Datenausgabe umfasst das Auslesen der aktuellen Anzeigewerte vom Simulationsprogramm bis hin zur Anzeige dieser am Pult selbst. Auch hier dient als Schnittstelle zwischen Simulationsprogramm und Hardware wiederum das VB-Programm am PC und der Arduino Mega.

3.2.1 Übertragungsweg der Datenausgabe



Quelle für die Anzeigewerte ist die Zusi-Simulation mit den simulierten Werten für die Motorströme, Geschwindigkeit, Zustände der Leuchtmelder usw. Diese werden via TCP-Schnittstelle an das PC-Programm übertragen, welches die Werte am Sekundärbildschirm anzeigt und für den Arduino umrechnet. Die Werte werden anschließend als Informationsbytes mit vorhergehendem Identifikationsbyte an den Arduino gesendet. Dieser steuert nun die Platinen an, welche die Analoginstrumente und Leuchtmelder bestromen.

3.2.2 Ausgabeinstrumente

3.2.2.1 Leuchtmelder

3.2.2.1.1 Allgemeines

Insgesamt werden 7 Leuchtmelder angesteuert. Da die Leuchtmittel der einzelnen Lampen auf verschiedene Nennspannungen ausgelegt sind und zur optimalen Potentialtrennung, werden diese mit Relais geschaltet. Näheres dazu in der Beschreibung der Platine „Relaiskarte“. Im Folgenden werden die Bedeutung sowie die Details der verschiedenen Leuchtmelder beschrieben.

3.2.2.1.2 Indusi 1000 Hz

Dieser Leuchtmelder gehört zum punktförmigen Zugbeeinflussungssystem „Indusi I54“. Er zeigt an, ob eine sogenannte „1000Hz Beeinflussung“ aktiv ist, was üblicherweise nach Passieren eines Vorsignals oder einer Geschwindigkeitsankündigung mit zu erwartender Maximalgeschwindigkeit von unter oder gleich 90km/h erfolgt.

Leuchtmittel: 24V Glimmlampe, 5W

Farbschirm: Orange, Abschrägung 45°

Klemme im Schaltkasten: 38



Abbildung 27: Leuchtmelder Indusi 1000Hz

3.2.2.1.3 Indusi P70-85

Informiert über die aktive Zugart. Die „Indusi I54“ verfügt über 2 Zugarten, welche festlegen auf welche Geschwindigkeit der Zug nach erfolgter 1000Hz Beeinflussung überwacht wird. Ist dieser LM aktiv, so befindet sich der Zug in der niedrigeren Zugart und es erfolgt eine Überwachung auf 75 km/h nach 26 Sekunden.

Leuchtmittel: 24V Glimmlampe, 5W

Farbschirm: Blau, Abschrägung 45°

Klemme im Schaltkasten: 37



Abbildung 28: Leuchtmelder Indusi Zugart P70-50

3.2.2.1.4 Indusi D95-105

Dieser Leuchtmelder zeigt an, dass sich der Zug in der höheren Zugart befindet. Dies bedeutet, dass eine Überwachung auf 95 km/h nach 20 Sekunden stattfindet.

Leuchtmittel: 24V Glimmlampe, 5W

Farbschirm: Blau, Abschrägung 45°

Klemme im Schaltkasten: 36



Abbildung 29: Leuchtmelder Indusi Zugart D95-65

Näheres zur Indusi im Kapitel 1.1.4

3.2.2.1.5 Hauptschalter aus

Leuchtet diese Anzeige, so ist der Hauptschalter ausgeschaltet, der Transformator ist also von der Spannungsversorgung getrennt. Ein Anfahren ist somit nicht möglich. Durch die weiterhin aktive Batterieversorgung ist aber die einwandfreie Funktion vieler sonstiger Bordanlagen weiterhin gegeben.

Leuchtmittel: 36V Stiftsockellampe, 2,5W

Farbschirm: Weiß, Schrift leuchtend „Hauptschalter aus“

Klemme im Schaltkasten: 35

3.2.2.1.6 Ölpumpe steht

Leuchtet, wenn die Ölpumpe zur Kühlung des Transformators nicht läuft. Da dieses Aggregat nicht simuliert wird, jedoch nur bei eingeschaltetem Hauptschalter aktiv ist, wird dieser Leuchtmelder vereinfacht gleichzeitig mit „Hauptschalter aus“ aktiviert. (Realisiert als Parallelschaltung)

Leuchtmittel: 36V Stiftsockellampe, 2,5W

Farbschirm: Weiß, Schrift leuchtend „Ölpumpe steht“

Klemme im Schaltkasten: 35

3.2.2.1.7 Lok steht

Zeigt an, ob sich die Lokomotive im Stillstand befindet.

Leuchtmittel: 36V Stiftsockellampe, 2,5W

Farbschirm: Weiß, Schrift leuchtend „Lok steht“

Klemme im Schaltkasten: 34

3.2.2.1.8 Elektrische Bremse

Dieser Leuchtmelder leuchtet solange, wie die Schütze zur Aktivierung der elektrischen Bremse angezogen sind. Da diese etwas verzögert ansprechen, muss der Triebfahrzeugführer darauf achten dass diese Kontrolllampe leuchtet, bevor er die E-Bremse nutzt.

Leuchtmittel: 36V Stiftsockellampe, 2,5W

Farbschirm: Weiß, Schrift leuchtend „Elektr. Bremse“

Klemme im Schaltkasten: 32

3.2.2.2 Zeigerinstrumente

Die Anzeige der Motor- und Bremsströme, des Primärstroms, der Fahrdraht- und Batteriespannung, sowie der Geschwindigkeit erfolgen über analoge Zeigerinstrumente. Die Ansteuerung dieser gestaltet sich schwieriger als jene der Leuchtmelder.

3.2.2.2.1 Motorstromanzeigen

Die Anzeige der Ströme in den 4 Fahrmotoren erfolgt über 4 Dreheiseninstrumente mit einem Nennstrom von 5A. Im Vorbild werden diese von Stromwandlern versorgt, die sich in den Zuleitungen zu den Gleichstrommaschinen befinden, welche Ströme von bis zu 3400A aufnehmen. Diese Anzeigen sind vor allem beim Anfahren für den Lokführer sehr wichtig, da sie ihm anzeigen ob er bereits auf die nächste Fahrstufe aufschalten kann, oder es dann zu einer Überschreitung des Maximalstromes käme. Da im Simulator der Antrieb vereinfacht simuliert wird gibt es nur einen Sollwert des Motorstromes, welcher für alle Motoren gilt. Daher konnten diese 4 Instrumente in Serie geschaltet werden. Näheres zur Platine zur Ansteuerung dieser Instrumente in der Beschreibung der Platine „Stromverstärker“.

Nennstrom: 5A

Messwerk: Dreheisenmesswerk

Hintergrundbeleuchtung: Passiv

Skala: 0 bis 4000A

Klemmen im Schaltkasten: 25(I+II), 26(III,IV)

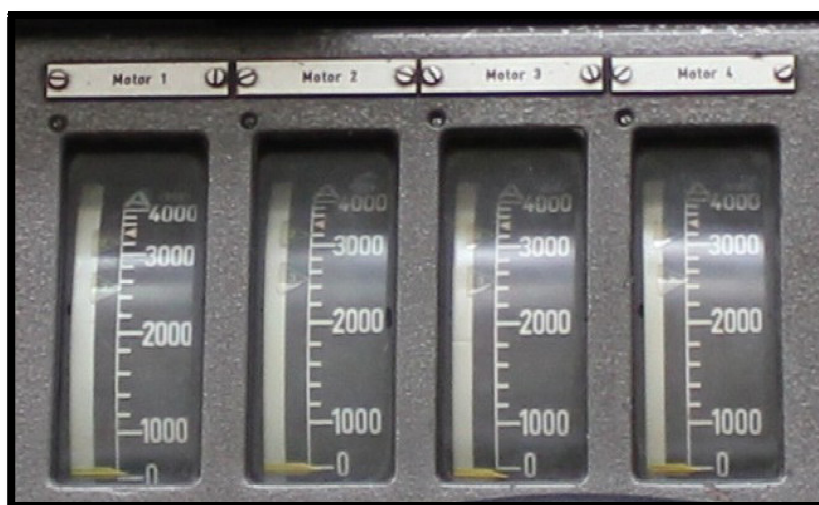


Abbildung 30: Motorstromanzeigen 1 bis 4

3.2.2.2 Bremsstromanzeigen

Diese 2 Anzeigen zeigen an, wie hoch die Ströme durch die 2 Bremswiderstände sind. Sie sind somit aktiv, wenn elektrisch gebremst wird. Wie die Motorstromanzeigen, sind dies Dreheiseninstrumente mit 5A Nennstrom. Auch diese Instrumente sind in Serie geschaltet angesteuert und werden von der PWM - Platine gespeist.

Nennstrom: 5A

Messwerk: Dreheisenmesswerk

Hintergrundbeleuchtung: Passiv

Skala: 0 bis 2500A

Klemmen im Schaltkasten: 27(I+II)



Abbildung 31: Bremsstromanzeigen

3.2.2.2.3 Hauptluftleitungs- und Hauptluftbehälterdruck

Dies ist eine doppelte Anzeige und informiert den Triebfahrzeugführer über den aktuellen Zustand der Bremsanlage. Die Zeiger werden mithilfe von in die Anzeigen eingebauten Servos bewegt. Damit der volle Stellbereich dieser ausgenutzt werden kann, mussten zusätzlich 2 Zahnräder mit einem Übersetzungsverhältnis von 2:5 eingebaut werden.

Hauptluftleitungsdruck:

Skala: 0 bis 10 bar

Antrieb: Servo

Hauptluftbehälterdruck:

Skala: 0 bis 14 bar

Antrieb: Servo

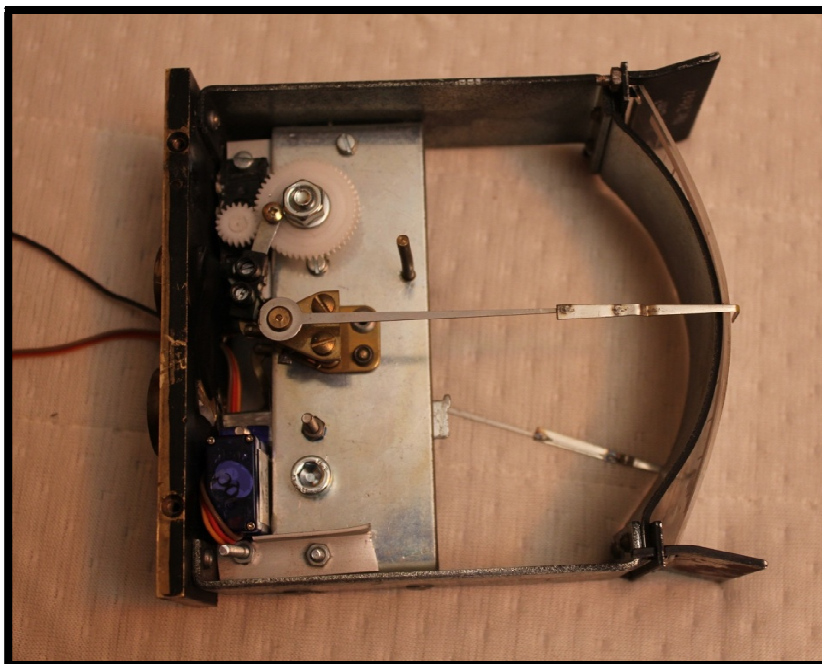


Abbildung 32: Antrieb der Druckmessgeräte

3.2.2.2.4 Stufenanzeige

Die Stufenanzeige informiert den Triebfahrzeugführer darüber, in welcher Position sich das Stufenschaltwerk im Maschinenraum der Lokomotive gerade befindet. Dies geschieht über eine Scheibe in der Mitte des Fahrschalters, welche mit einem Pfeil versehen ist. Diese Scheibe kann durch eine Welle, welche auf der Unterseite der Fahrschaltereinheit ausgeführt ist, bewegt werden. Dabei entspricht eine Umdrehung an der Welle genau einer Fahrstufe. Um eine genaue Positionierung der Anzeige zu ermöglichen, wurde als Antrieb ein bipolarer Schrittmotor der Type JK57H76-2504 gewählt, welcher von der Schrittmotorsteuerung M335-B angesteuert wird.



Abbildung 33: Die rotierende Scheibe der Stufenanzeige

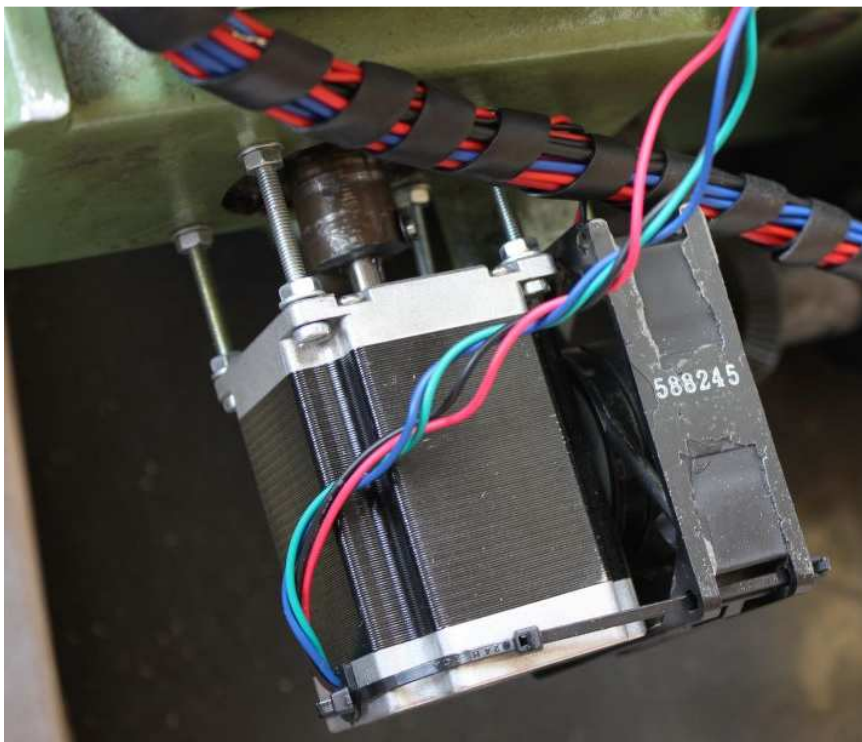


Abbildung 34: Schrittmotor an der Welle der Fahrschaltereinheit

3.3 Schaltplan

3.3.1 Schematische Darstellung

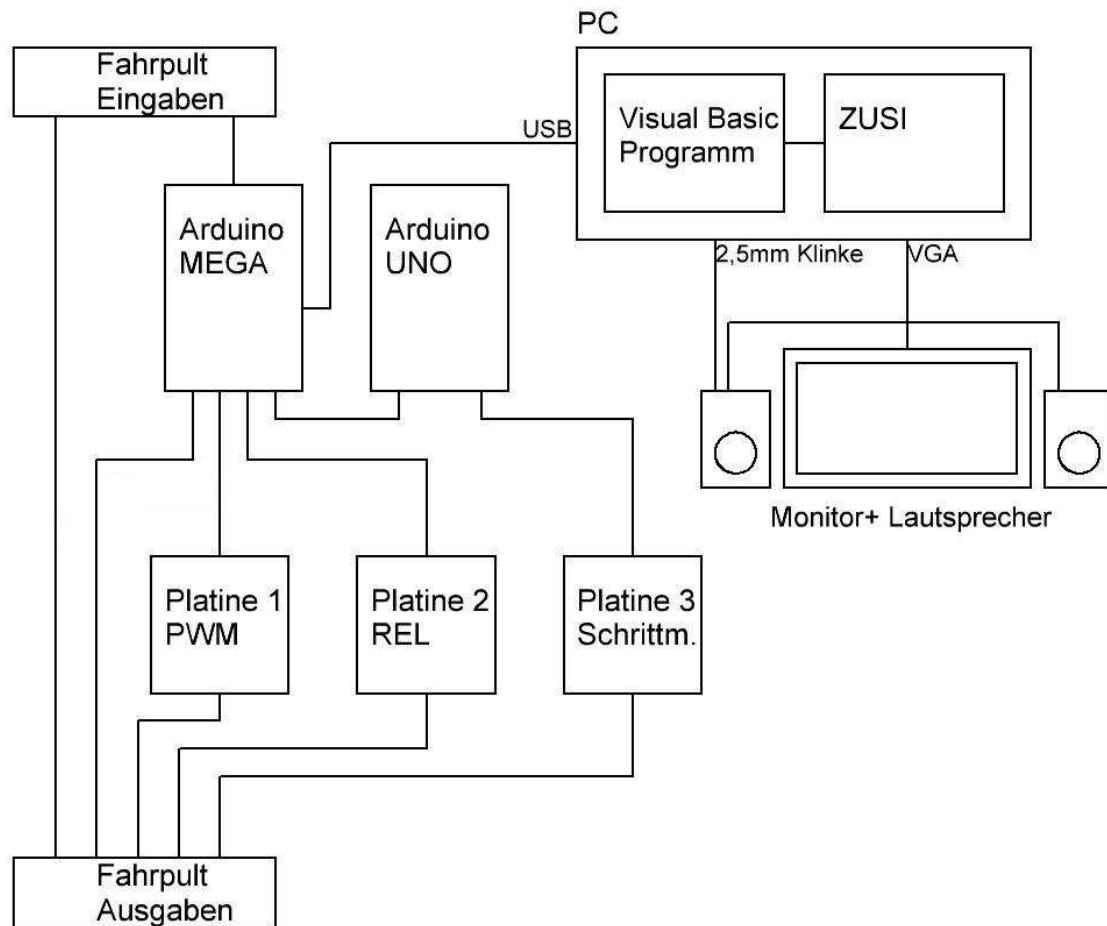
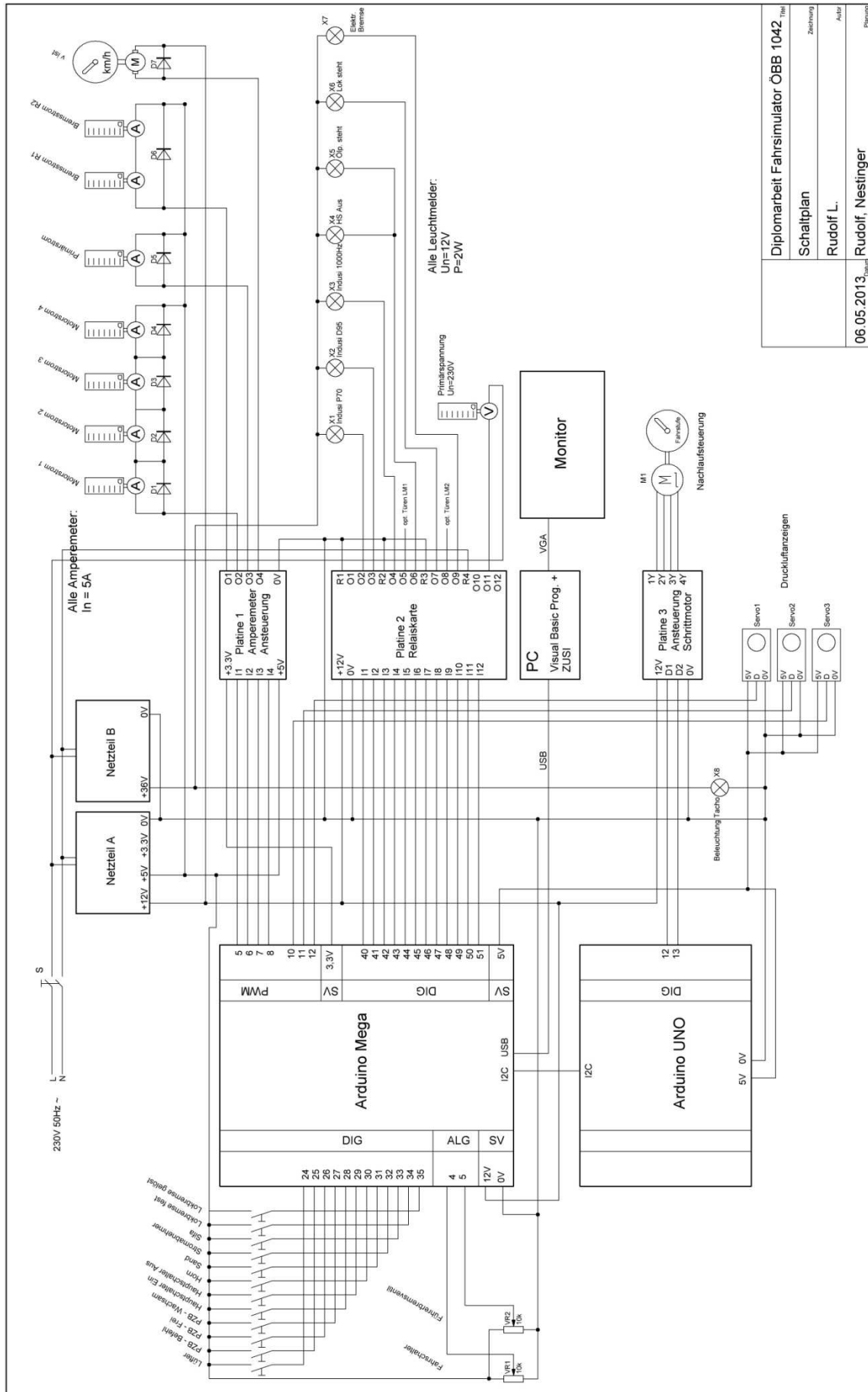


Abbildung 36: Schematische Darstellung des Datenflusses

Hier sind alle Wege des Informationsflusses eingezeichnet. Die Datenleitungen von den Eingabegeräten am Fahrpult (Fahrshalter, Bremsventile, Schalter, Taster) verlaufen zu den Eingängen des Arduino MEGA. Die einzige Ausnahme bildet das Anzeigeeinstrument der Batteriespannung, welches direkt vom Stromabnehmerschalter geschaltet wird. Die PWM Signale vom Arduino MEGA werden von der Platine 1 (AMP) verstärkt und führen zu den Anzeigeeinstrumenten (Amperemeter). Die digitalen Datenleitungen des Arduino verlaufen zur Platine 2 (REL) und schalten über Optokoppler und Relais die Stromkreise der Leuchtmelder. Der Schrittmotor für die Nachlaufsteuerung wird von einer eigenen Ansteuerung versorgt, welche vom Arduino UNO angesteuert wird. Dieser weitere Arduino war unvermeidbar, da sonst das Programm am Arduino MEGA zu komplex geworden wäre. Die beiden Arduino sind über einen I2C Bus gekoppelt. Über einen USB Anschluss werden die Daten vom Arduino MEGA zum Computer übertragen, wobei sie in einem Visual Basic Programm erneut angepasst werden. ZUSI umfasst neben der 3D Anwendung auch den Soundthesizer (realistischer Zugsound) und den TCP Server. Ein angeschlossener Flachbildschirm und ein integriertes Lautsprechersystem mit Subwoofer sorgen für ein realistisches Fahrgefühl.

3.3.2 Schaltplan



06.05.2013	06.05.2013	Rudolf, Nestinger
Schaltplan		Zeichnung
Schaltplan		Zeichnung
Rudolf L.		Autos
Rudolf, Nestinger		Planung

3.4 Platinen

Um die digital übertragenen Informationen auch auf den Anzeigeelementen wie Strom- und Spannungsmessern, mechanischen Anzeigen, sowie Leuchtmeldern anzeigen zu können, mussten Platinen entworfen werden, welche die dafür nötigen höheren Leistungen bewerkstelligen können. Von den ursprünglich geplanten 3 Platinen blieben schlussendlich 2 übrig. Einerseits eine PWM-Platine mit 4 LowSide Kanälen für die Ansteuerung der Stromanzeigen und des Tachometers, sowie eine Platine mit 12 Relais für die Leuchtmelder und die Anzeige der Fahrdrachtspannung.

3.4.1 PWM Platine

Diese Platine besteht aus 4 unabhängigen Kanälen mit LowSide Schaltern, welche über Optokoppler von der Eingangsseite getrennt sind. Die Ausgänge können mit ca. 10A belastet werden und sind für PWM-Signale im kHz Bereich geeignet. Erstellt wurde die Platine in der schuleigenen Leiterplattenfertigung.

3.4.1.1 Schaltplan

Im Folgenden ist der Schaltplan der Platine zu sehen:

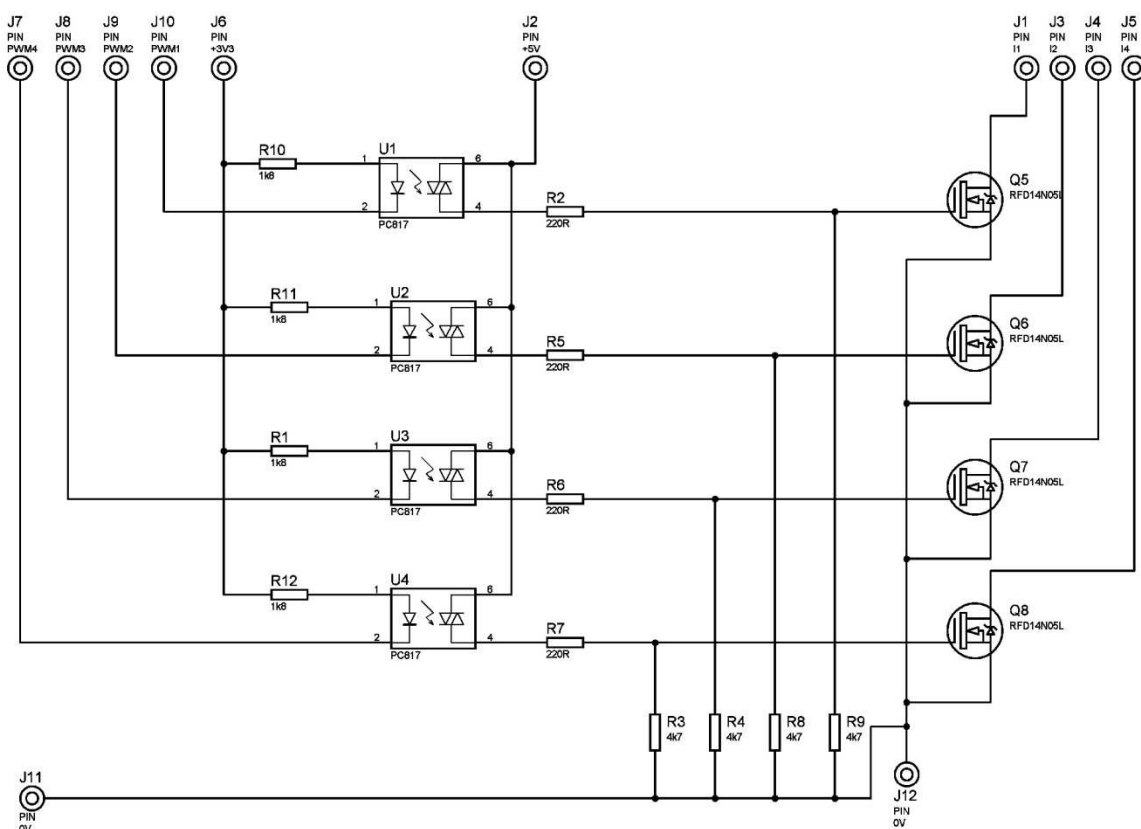


Abbildung 37: Schaltplan PWM-Platine

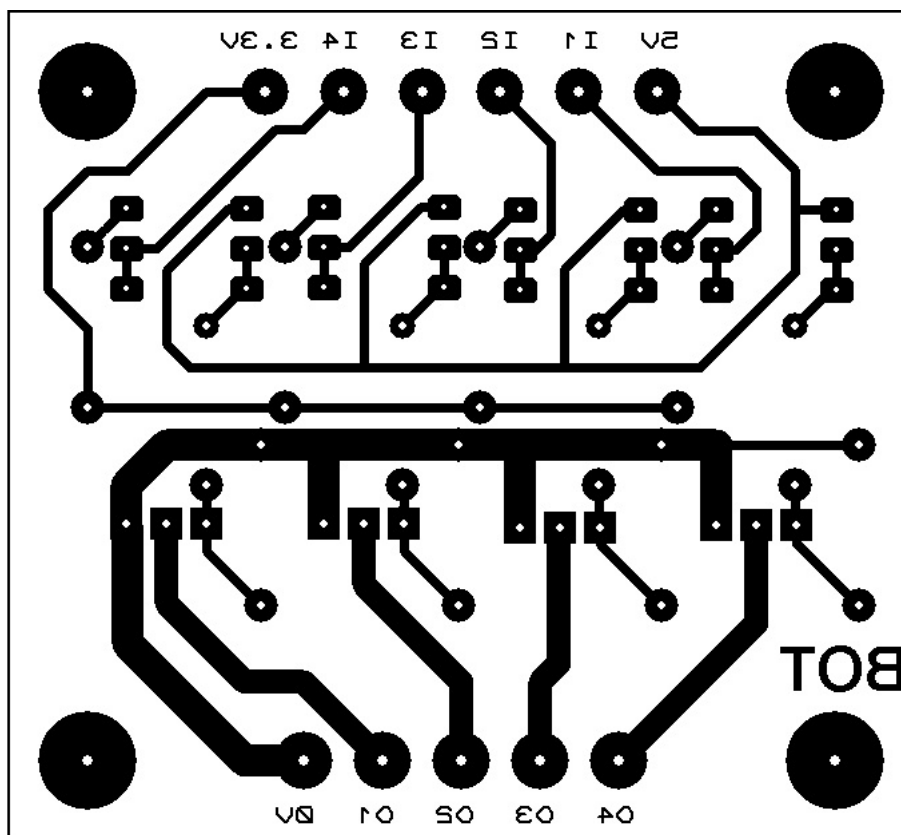
3.4.1.2 Funktionsprinzip

Es wird das Funktionsprinzip anhand des Kanals 1 der Platine erklärt, die weiteren Kanäle arbeiten nach demselben Prinzip, mit entsprechend anderen Bauteilbezeichnungen.

Der Pin J6 wird mit der positiven Spannungsversorgung der Signalquelle verbunden, in diesem Fall mit jener des Arduino. Der Pin J7 wird mit dem Digitalausgang jener verbunden und zum Aktivieren des Ausgangs auf 0V geschaltet. Dies bedingt eine Spannung an der LED des Optokopplers U1, welcher nun auf der Sekundärseite durchschaltet. Zur Strombegrenzung durch den Optokoppler dient dabei R10. Somit werden nun die 5V, welche an Pin J2 angeschlossen sind, über den Widerstand R2 zur Basis des Mosfet Q5, welcher den Ausgang O1 gegen Masse schaltet. Ein ungewolltes Durchschalten verhindert der Pulldown - Widerstand R9.

3.4.1.3 Entflechtung

Der fertig entflechtete Entwurf der Platine, wie sie anschließend gefertigt wurde.



3.4.1.4 Bestückung

Nachdem die Platine fertig geätzt war, wurde die Bestückung mit den vorgesehenen Bauteilen vorgenommen. Die Dimensionierung des Kühlkörpers erfolgte aus Erfahrungswerten.

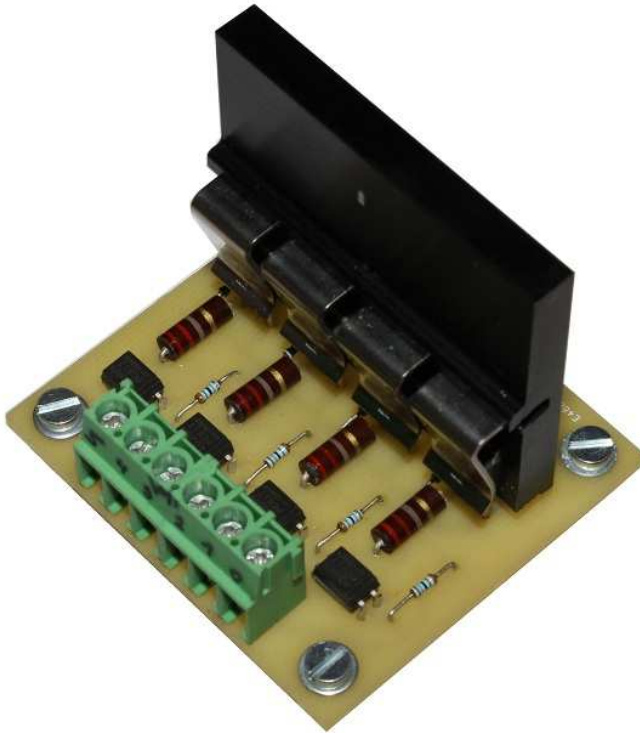


Abbildung 38: Fertig bestückte PWM - Platine

3.4.1.5 Anschlussbelegung

Klemmleiste	Anschluss	Bezeichnung
F1:	0	+5V
	1	Input 1
	2	Input 2
	3	Input 3
	4	Input 4
F2:	5	3.3V
	0	0V
	1	Output 1
	2	Output 2
	3	Output 3
	4	Output 4

Tabelle 20: Anschlussbelegung PWM-Platine

3.4.2 Relaiskarte

Für die Ansteuerung der Leuchtmelder wurde eine Platine mit 12 Schaltausgängen entworfen, wobei jeweils 3 Ausgänge zu einer Wurzel zusammengefasst wurden. Als Relais wurden 12 Stück der Type "Zettler az695-12" verwendet, welche mit bis zu 250V~ und 5A belastet werden können.

3.4.2.1 Schaltplan

Im Folgenden ist der Schaltplan der Platine zu sehen:

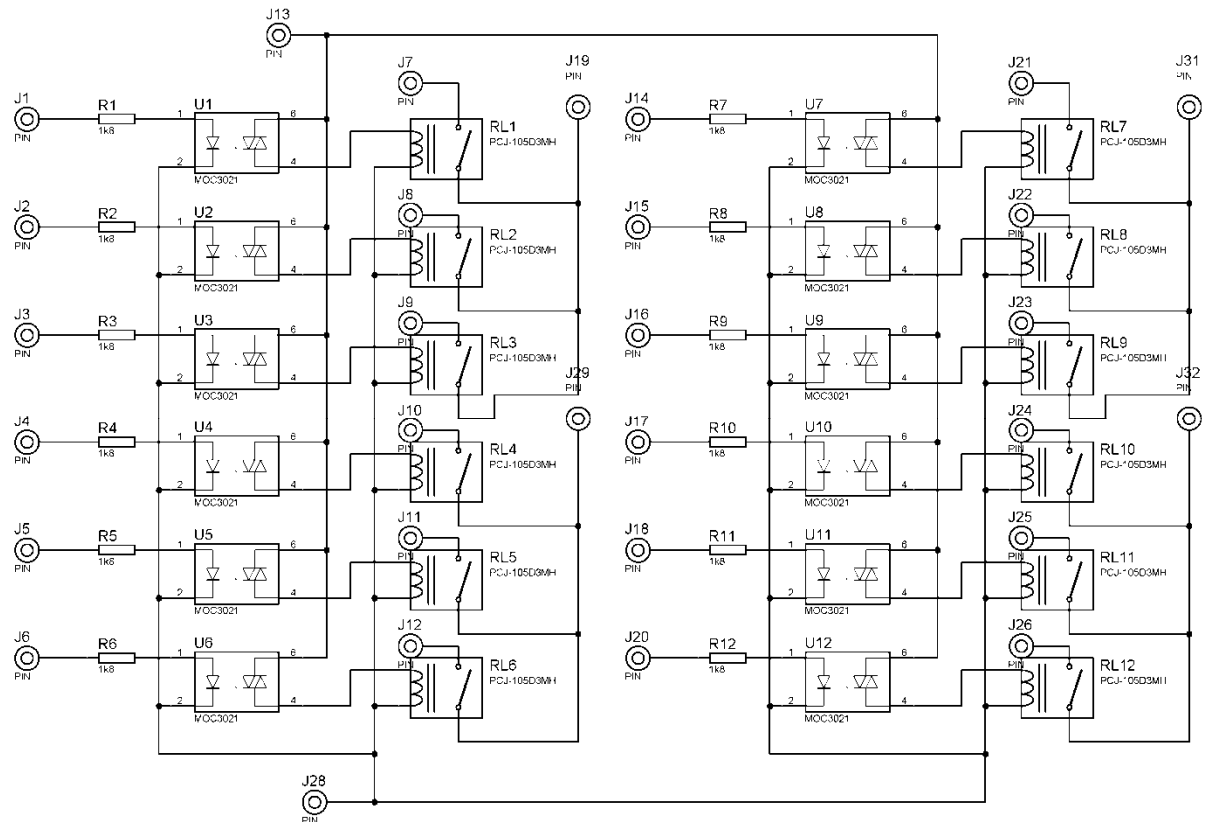


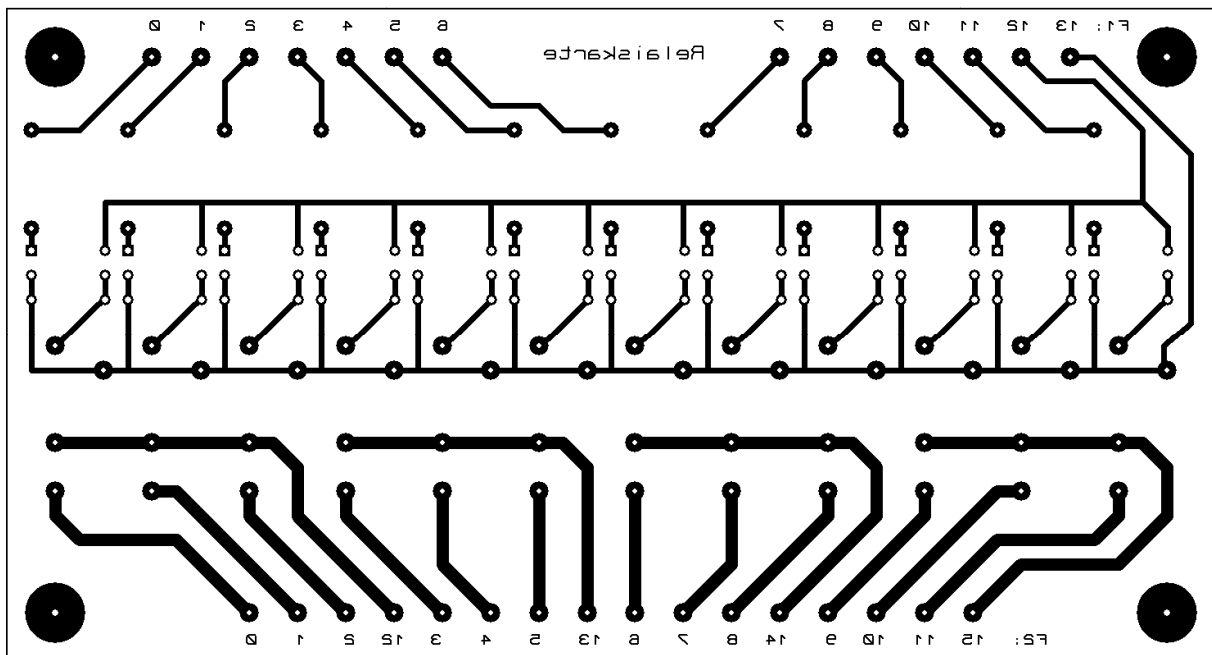
Abbildung 39: Schaltplan Relaiskarte

3.4.2.2 Funktionsprinzip

Wird auf einem der Eingänge (J1-J6, J14-J20) eine Spannung von 5V angelegt, so schaltet der Optokoppler dieses Strangs durch und bestromt so die Wicklung des entsprechenden Relais. Dies verbindet den Ausgang dieses Stranges (J7-J12, J21-J26) mit der dazugehörigen Wurzel. Der Pin J13 wird mit der zentralen Masse verbunden.

3.4.2.3 Entflechtung

Das Entflechten dieser Platine gestaltete sich relativ einfach, prinzipiell wurde ein Strang 12 mal kopiert und mit den entsprechenden Pins verbunden.



3.4.2.4 Bestückung

Wiederum wurde die Platine mit den dafür vorgesehenen Bauteilen bestückt.

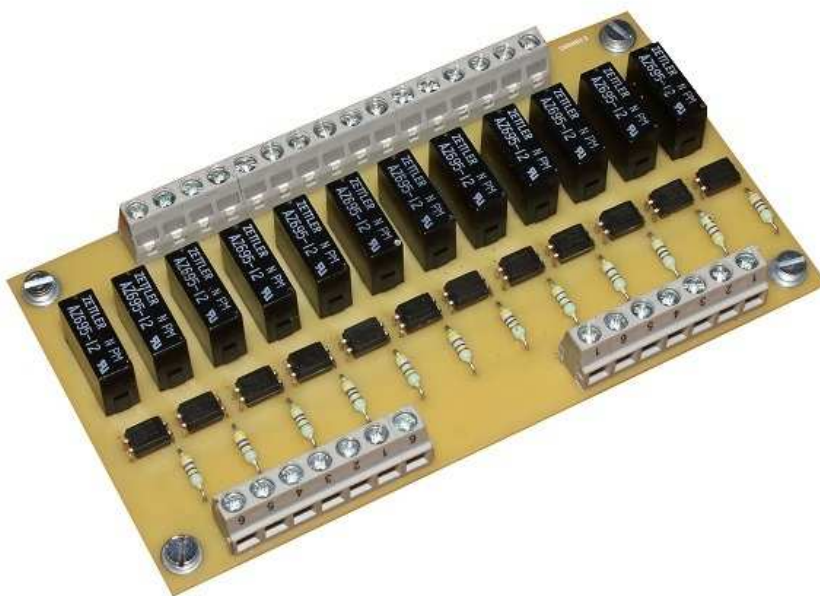


Abbildung 40: Fertig bestückte Relaiskarte

3.4.2.5 Anschlussbelegung

Klemmleiste	Anschluss	Bezeichnung
F1:	0	Input 0
	1	Input 1
	2	Input 2
	3	Input 3
	4	Input 4
	5	Input 5
	6	Input 6
	7	Input 7
	8	Input 8
	9	Input 9
	10	Input 10
	11	Input 11
	12	+12V
	13	0V
F2:	0	Output 0
	1	Output 1
	2	Output 2
	3	Output 3
	4	Output 4
	5	Output 5
	6	Output 6
	7	Output 7
	8	Output 8
	9	Output 9
	10	Output 10
	11	Output 11
	12	Wurzel 1
	13	Wurzel 2
	14	Wurzel 3
	15	Wurzel 4

Tabelle 21: Anschlussbelegung Relaiskarte

3.5 Vollständige Verkabelung

Dieses Kapitel zeigt Bilder der vollständigen Verkabelung auf der Rückseite des Fahrsimulators.



Abbildung 41: Rückseite des Fahrsimulators

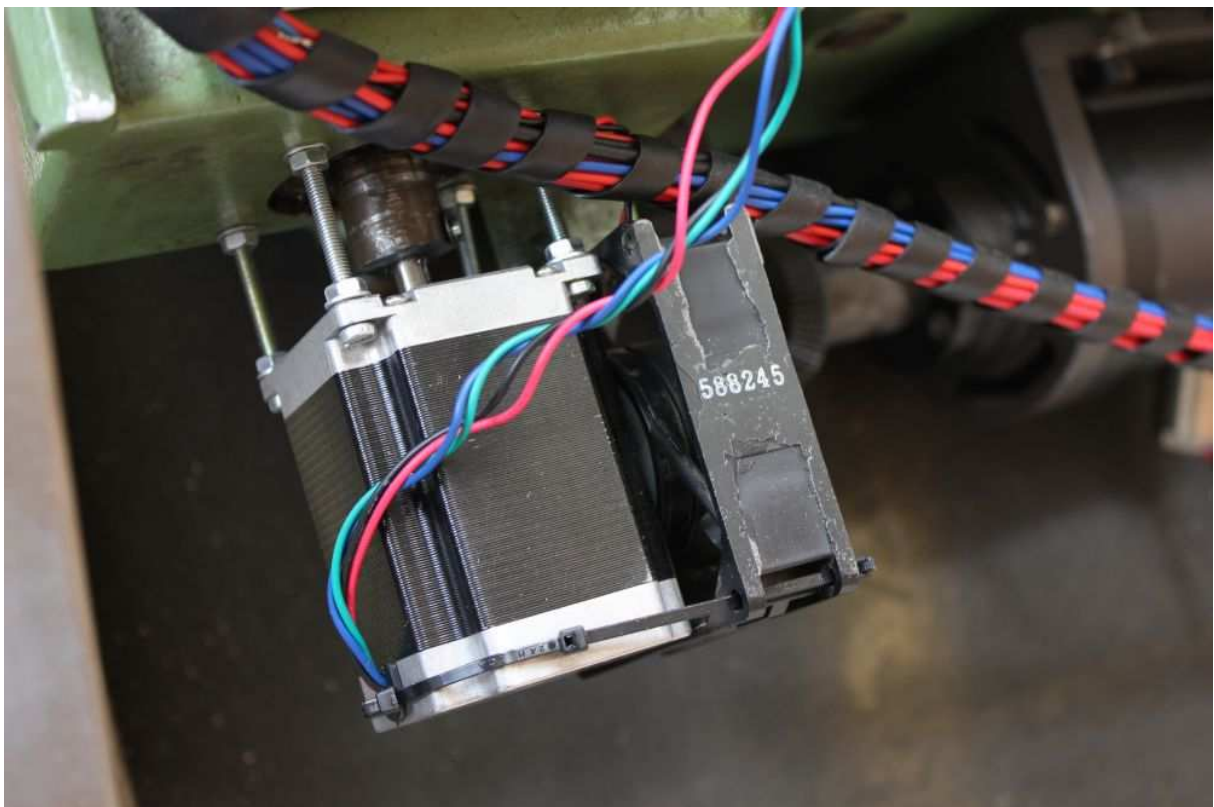


Abbildung 42: Schrittmotor mit Lüfter



Abbildung 43: Fahr Simulator mit Abdeckplatte auf der Rückseite

Die Abdeckplatte wurde zugeschnitten und an das Stahlgerüst angeschraubt. Desweiteren waren eine Kabelausführung sowie Lüftungseinlässe für die Luftzirkulation notwendig.

4 Software

Zum einen werden in diesem Kapitel jene Programme erklärt, die für diese Diplomarbeit eigens entwickelt oder adaptiert wurden, was im Wesentlichen auf die Programme zur Datenübertragung zutrifft, in welche ein wesentlicher Teil an Arbeitszeit investiert wurde. Weiters werden jene Programme im Detail vorgestellt, welche für das Projekt erworben wurden und für die Funktion benötigt werden. Eine detaillierte Anleitung zur Inbetriebsetzung findet sich in Kapitel 5.

4.1 Zugsimulation "ZUSI"

Als Simulationssoftware kommt das deutsche Zugsimulationsprogramm "Zusi" in der Version 2.4 zum Einsatz. Diese Software wurde vom Braunschweiger Carsten Hölscher programmiert und ist im Hinblick auf die akkurate Simulation des Fahrverhaltens eines Zuges das beste für private Anwender erhältliche Programm. Zusi legt größten Wert auf eine möglichst realitätsnahe Bedienung des simulierten Triebfahrzeuges, die genaue Nachbildung von Signal- und Zugsicherungssystemen, sowie eine vorbildgerechte Gleisgeometrie.

Das Wichtigste stellt im Zusi wie auch in der Realität der Fahrplan dar. Dieser kann mehrere Hundert Züge umfassen und wird anfangs geladen. Dabei sind alle Züge in der Simulation gleichwertig und werden von einer künstlichen Intelligenz (KI) geführt. Es kann nun ein Zug ausgewählt werden, der vom Anwender gefahren wird und sich ebenfalls in den Fahrplan eingliedert. Es herrscht also reger Betrieb mit Gegenverkehr, Überholungen, Zugkreuzungen - aber auch Baustellen, defekte Weichen und andere besondere Vorkommnisse werden simuliert. Aufgabe ist es nun, den eigenen Zug anhand des sogenannten Buchfahrplans sicher und pünktlich durch die virtuelle Landschaft zu steuern - ohne dabei Fahrfehler wie zu schnelles Fahren oder Missachtung von Signalen zu begehen. Es besteht jedoch jederzeit die Möglichkeit den derzeit aktiven Zug wieder an die KI zu übergeben und einen anderen Zug im Fahrplan auszuwählen. Zusi ist hier also sehr flexibel.

Standardmäßig zeigt Zusi auch den Führerstand der Lok als Bitmap mit animierten Bedienelementen und Anzeigeelementen an. Dieser kann für die hier benötigten Zwecke jedoch ausgeblendet werden, da dieselben Anzeigen mechanisch am Fahrpult vorhanden sind.

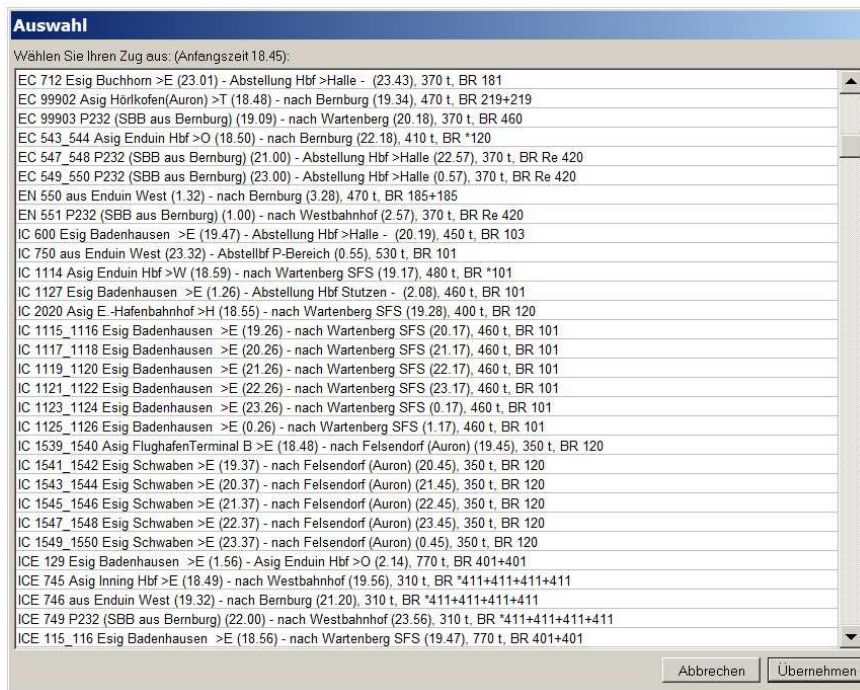


Abbildung 44: Beispiel einer Zugauswahl in Zusi nach dem Laden eines Fahrplans

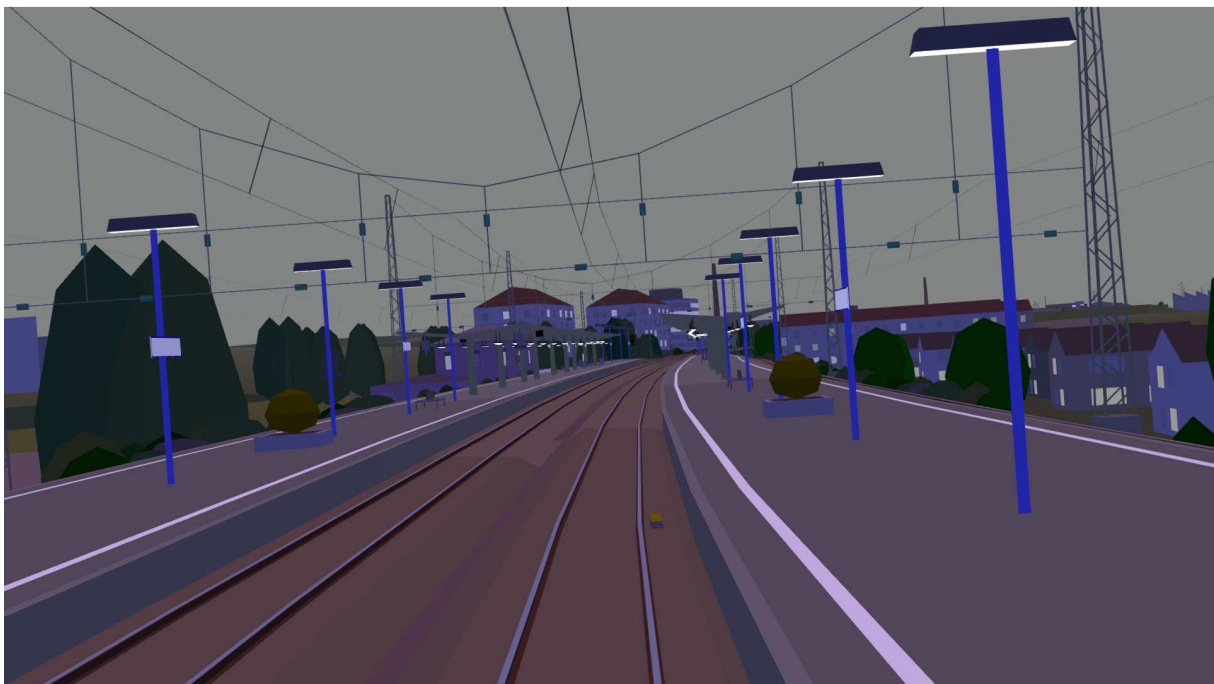


Abbildung 45: Zugfahrt auf der Linken Rheinstrecke



Abbildung 46: Aufnahme der Linken Rheinstrecke

Weitere Informationen zum Programm unter <http://www.zusi.de>

4.2 Übertragungssoftware

4.2.1 Programm am PC

Am PC läuft auf der Windows - Oberfläche ein Visual Basic Programm. Dieses dient zur Übertragung und Werteanpassung für die Datei Ein- und Ausgabe. Außerdem können für die Bedienung relevante Werte direkt abgelesen werden und so die Fahrweise des Lokführers überwacht werden.

4.2.1.1 Programmoberfläche

Die Programmoberfläche wurde visuell in 3 Teile eingeteilt, links befindet sich das Panel zur Verbindung des Programms mit Zusi und mit dem Fahrpult.



Abbildung 47: Übertragungsprogramm am PC

4.2.1.1.1 Verbindungsteil



Abbildung 48: Linker Programmteil

In den Feldern links ist die standardmäßige Adresse des TCP-Servers eingetragen, als IP "localhost", da der Server am selben Rechner läuft, als Port "1435". Prinzipiell wäre es also auch möglich, Zusi auf einem anderen Rechner laufen zu lassen und diesen via Ethernet mit dem Fahrpult - Rechner zu verbinden - dann ist hier natürlich die entsprechende IP-Adresse einzutragen.

In der rechten Dropdown-Liste sind die am PC vorhandenen COM-Anschlüsse aufgelistet, wobei hier der COM-Port des Arduino Mega vom Fahrpult ausgewählt werden muss.

Unten ist noch die CheckBox "Daten an Zusi senden" auswählbar. Ist dieser Punkt aktiviert, werden die Tastendrücke an Zusi gesendet. Dieser Punkt wird sinngemäß als letzter aktiviert (Näheres dazu im Kapitel "Inbetriebnahme").

4.2.1.1.2 Überwachungsteil

Fahrpult	
<input type="checkbox"/> PZB Wachsam	<input type="checkbox"/> Sifa
<input type="checkbox"/> PZB Frei	<input type="checkbox"/> Lüfter
<input type="checkbox"/> PZB Befehl	<input type="checkbox"/> Zbv Los
<input type="checkbox"/> Sand	<input type="checkbox"/> Zbv An
<input type="checkbox"/> Makrophon	<input type="checkbox"/> Stab
	Fahrschalter FSSoll
	Führerbremssventil
Zusi	
<input type="checkbox"/> LM PZB O	Primärstrom: <input type="text" value="000"/>
<input type="checkbox"/> LM PZB M	Motorstrom: <input type="text"/>
<input type="checkbox"/> LM 1000 Hz	Hauptluftleitung: <input type="text"/>
<input type="checkbox"/> LM HS aus	<input type="text"/>
<input type="checkbox"/> Spannung	Bremszylinder: <input type="text"/>
	Geschwindigkeit: <input type="text"/>
	Bremsstrom: <input type="text"/>

Abbildung 49: Mittlerer Programmteil

Hier werden die aktuellen Zustände vom Pult anschaulich dargestellt. Im Bereich "Fahrpult" sind alle Schalter und Taster als "Check-Boxes" dargestellt. Wird ein Taster gedrückt, erscheint der Haken der CheckBox. Dies kann zur Überwachung der korrekten Funktion der Schalter und der Datenverbindung genutzt werden. Im Bereich "Zusi" werden die Zustände der Leuchtmelder und Analogwerte angezeigt.

4.2.1.1.3 Überwachungs- und Debugteil

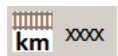
Fahrtüberwachung:



Abbildung 50: Fahrtüberwachung

Hier werden diverse Informationen über die aktuelle Fahrt angezeigt.

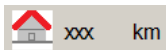
Anzeigen der Fahrtüberwachung:



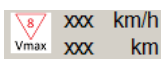
Anzeige des aktuellen Strecken - km: Hierdurch kann die aktuelle Position des Zuges auf der Strecke ermittelt werden.



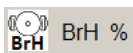
Anzeige der aktuellen Höchstgeschwindigkeit: Hier wird die aktuelle Streckenhöchstgeschwindigkeit angezeigt, somit kann ermittelt werden ob gerade zu langsam oder zu schnell gefahren wird.



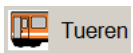
Anzeige der Entfernung zum nächsten Bahnhof: Anhand dieser Anzeige kann abgelesen werden, wie weit der nächste planmäßige Halt entfernt ist.



Anzeige der nächsten Geschwindigkeitsreduktion: Hier wird die Höhe und die Entfernung zur nächsten Geschwindigkeitsbeschränkung angezeigt.



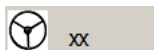
Anzeige der Bremswertstel des Zuges: Anhand dieses Wertes kann der Bremsweg des Zuges eingeschätzt werden. Der Zug brems besser, wenn dieser Wert höher ist.



Anzeige Türen: Hier wird bei einem Reisezug angezeigt, ob die Türen geöffnet oder geschlossen sind.



Anzeige der Istgeschwindigkeit: Hier wird die derzeitige Fahrgeschwindigkeit des Zuges angezeigt.



Anzeige der Schaltwerkstufe: Hier wird die aktuelle Stellung des Stufenschaltwerks angezeigt.

Debugteil:

Hier werden einige Infos über die Datenübertragung angezeigt, die zur Ermittlung von Fehlern dienen können. Dieser Teil wird im Normalbetrieb nicht benötigt. Über die 2 Schieberegler kann ein Offset des Fahrschalters und der Geschwindigkeitsanzeige eingestellt werden.

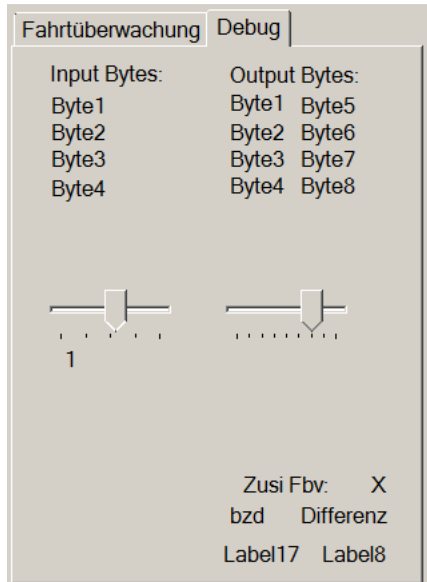


Abbildung 51: Debugteil

4.2.2 Programm am Arduino

Auf dem Arduino befindet sich ein Programm zum Empfang der Daten vom PC, sowie zur Anpassung dieser für die Ansteuerung der Platinen. Weiters werden die Analog- und Digitalwerte vom Pult eingelesen und in die andere Richtung übertragen. Eine nähere Beschreibung des Programms findet sich direkt im Quellcode im Anhang A2.

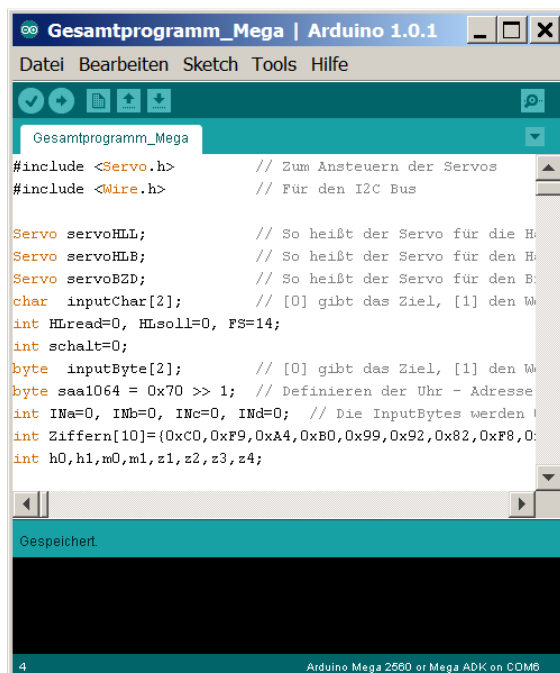


Abbildung 52: Programm in der Arduino-Programmieroberfläche

5 Inbetriebsetzung und Betrieb

Für eine fehlerfreie und sichere Inbetriebsetzung muss eine bestimmte Reihenfolge beim Start und der Verbindung des Fahrpults mit dem Simulationsprogramm eingehalten werden und sich vergewissert werden, dass alle Teilprogramme und die Hardware fehlerfrei gestartet bzw. aufgebaut sind.

5.1 Aufbau und mechanischer Test

Zuerst wird die Hardware aufgebaut, das Pult also fix aufgestellt und ein Anzeigergerät dahinter platziert. Der PC/Laptop wird neben dem Pult aufgestellt und muss die nötige Software für das Betreiben des Fahrpults installiert haben:


Zusi Fahr Simulator	Simulation des Zuges und Ausgabe von Bild und Ton.
TCP Server	Dient zur Übertragung der Istwerte zum Pult.
Connector Fahrpult 1042	Das auf der CD enthaltene Übertragungsprogramm für das Fahrpult.
<i>Optional: ZUSI Soundthesizer</i>	Dient zur realitätsnäheren Audioausgabe.

Tabelle 22: Nötige Programme

Die USB-Verbindung des Pults mit dem PC kann nun bereits hergestellt werden. Nachdem die Stellung des Netzschalters des Pultes auf "0" überprüft wurde (linke Seite des Fahrpults), kann der Netzstecker mit einer 230V~ Versorgung verbunden werden. Der Nennstrom liegt bei 500mA

5.2 Starten der Programme

Zuerst wird der Zusi Fahr Simulator gestartet (=> Zusi.exe) und eine Strecke geladen:

 Zusi.exe

Nun wird eine Strecke geladen, üblicherweise im Vollbildmodus:

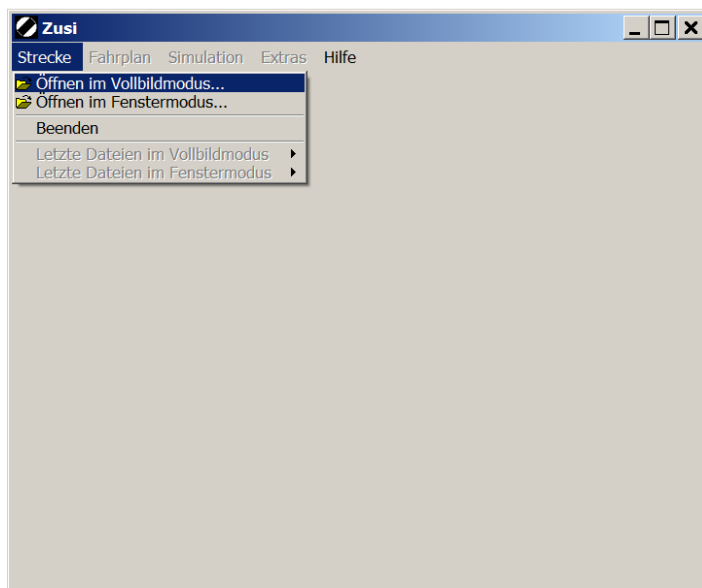


Abbildung 53: Strecke laden

Anschließend wird ein Fahrplan geladen:

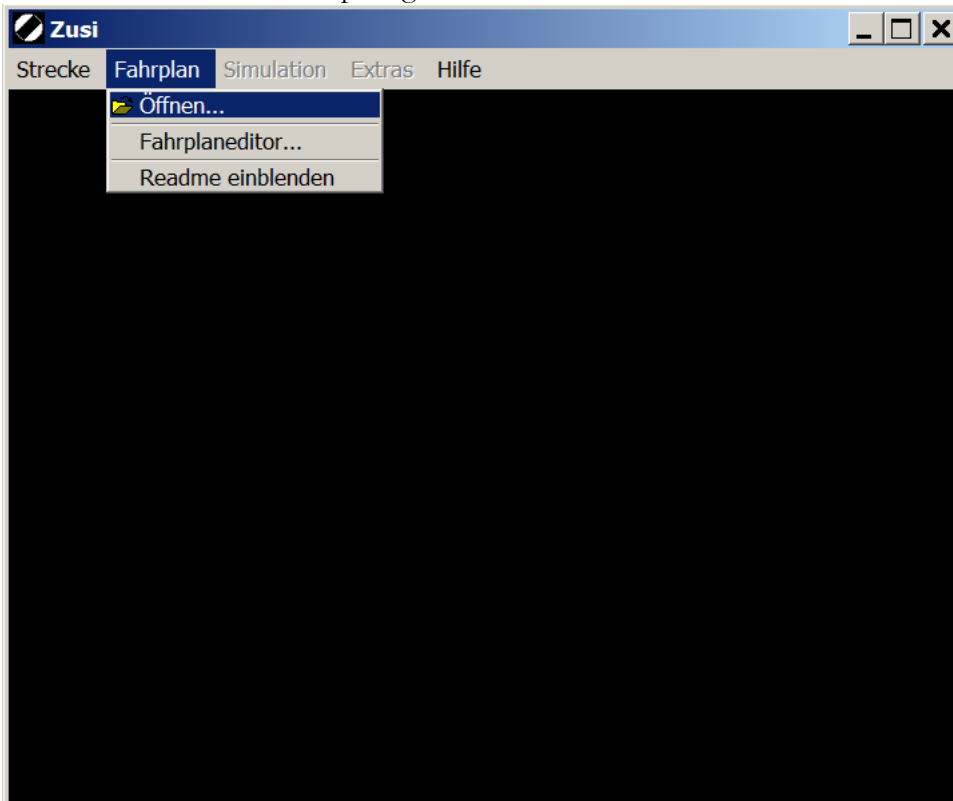


Abbildung 54: Fahrplan laden

Nun wird ein Zug mit der entsprechenden Lokomotive 1042 ausgewählt und gestartet:



Abbildung 55: Gestartete Zugfahrt

Nun werden in Zusi die Einstellungen aufgerufen (STRG + E) und kontrolliert. Die Tastenbelegung muss folgendermaßen konfiguriert werden: (Von der Standardbelegung abweichende Einträge wurden orange markiert)

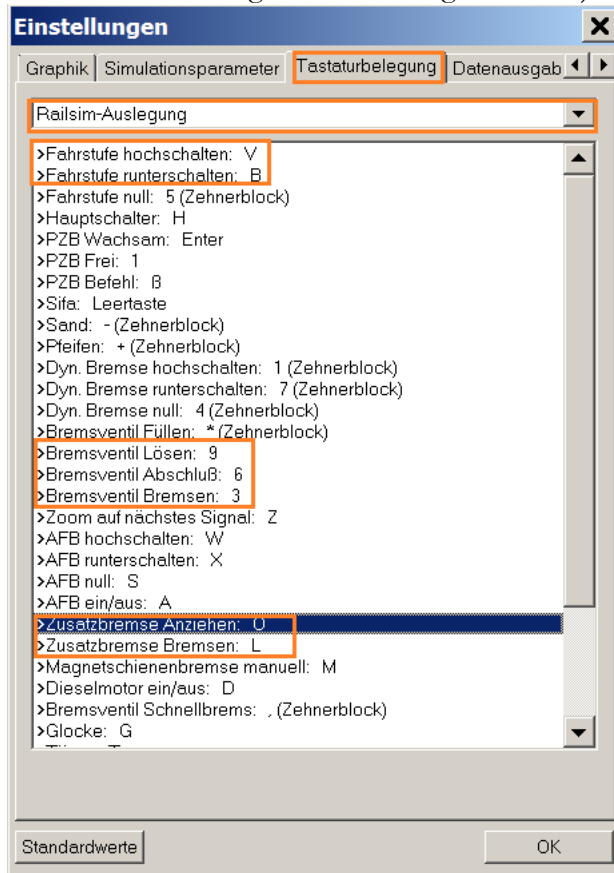


Abbildung 56: Zusi-Einstellungen

Nun kann Zusi minimiert werden (Windows - Taste oder STRG + TAB) und die Zusatzprogramme werden gestartet. Zuerst wird der TCP-Server initialisiert, mittels Ausführen von TCPserver.exe.

Nun wird das Übertragungsprogramm "Connector Fahrpult 1042" gestartet und zuerst mit dem Fahrpult (rechter Verbinden - Button), und anschließend mit dem TCP-Server verbunden:



Abbildung 57: Verbinden des Übertragungsprogramms

Weitere Zusatzprogramme die über TCP Infos aus Zusi beziehen, müssen nun gestartet und mit dem TCP-Server verbunden werden.

Nun wird wieder zurück zum Zusi Fahr Simulator gewechselt und bis zur Abfahrtszeit vorgespult, falls der Zug noch nicht aufgegleist wurde (F11). Ist der Zug aufgegleist, wird wieder in die Einstellungen gewechselt und der Reiter "Datenausgabe" ausgewählt.

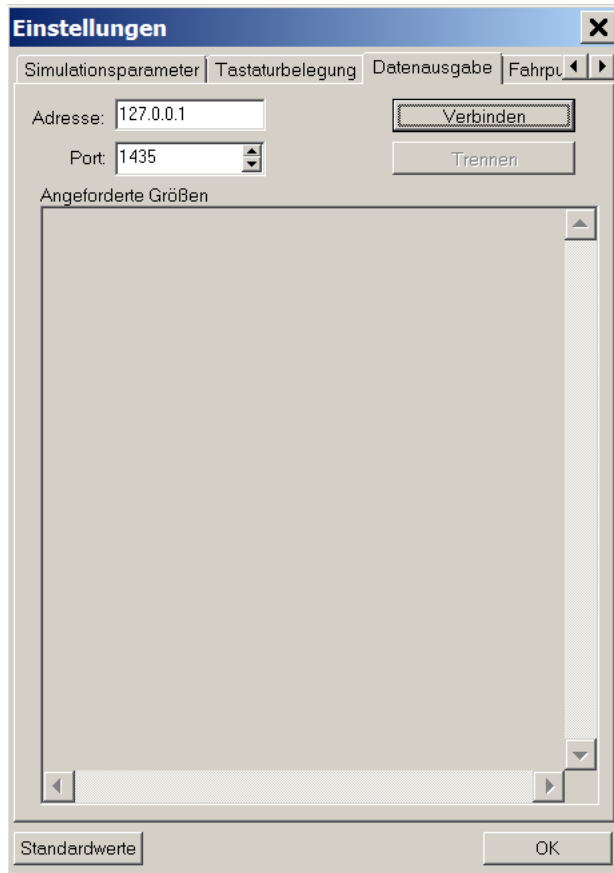


Abbildung 58: Aktivierung der Zusi-Datenausgabe

Nun werden die Einstellungen geschlossen und das Übertragungsprogramm auf korrekte Verbindung überprüft. Ist die Verbindung unterbrochen oder eine Fehlermeldung aufgetreten, muss die Routine ab dem Start des Übertragungsprogrammes wiederholt werden.

Ist die Verbindung erfolgreich hergestellt, so kann nun der Netzschalter des Fahrpults in die Stellung "1" gebracht werden. Die Leuchtmelder und Analoginstrumente sollten nun in die Sollpositionen wie auf dem Zusi-Führerstand übergehen. Sollten Fehler auftreten oder Messgeräte ungewollt ausschlagen, ist die Verbindung wieder zu trennen und Fachpersonal zu kontaktieren. Im Normalfall jedoch wird nun wieder zum Übertragungsprogramm gewechselt und der Haken bei "Senden der Daten an Zusi" gesetzt und anschließend der Fokus auf den Zusi-Fahr Simulator gesetzt (Mausklick oder STRG+TAB). Die Bedienelemente im Simulator sollten nun den Handlungen am Pult folgen. Nun wird noch der Führerstand vom Zusi-Programm ausgeblendet (Taste F4), um einen freien Streckenblick zu erhalten. Der Simulator ist nun einsatzbereit und der ausgewählte Zug kann laut Fahrplan gefahren werden.



Abbildung 59: Fahrpult verbunden und einsatzbereit

5.3 Fahren eines Zuges

5.3.1 Praxissituationen anhand von Beispielen

Im Folgenden finden sich einige Beispiele von Fahrsituationen in Zusi, wie sie bei einer Zugfahrt vorkommen können, anhand von Screenshots. Anschließend wird erklärt was in der angegebenen Situation zu tun ist und wie man sich als Triebfahrzeugführer des virtuellen Zuges zu verhalten hat. Voraussetzung zum Verständnis dieser Beispiele sind die Erläuterungen zu Signal- und Zugsicherungstechnik in Kapitel 1.

5.3.2 Österreichische Signaltechnik

Beispiel I - Fahren nach Signalgeschwindigkeit:

Sie verlassen gerade den Bahnhof Mürzzuschlag mit einem Eilzug von 150m Länge und nähern sich in der Dämmerung mit ca. 40km/h folgendem Signal an:



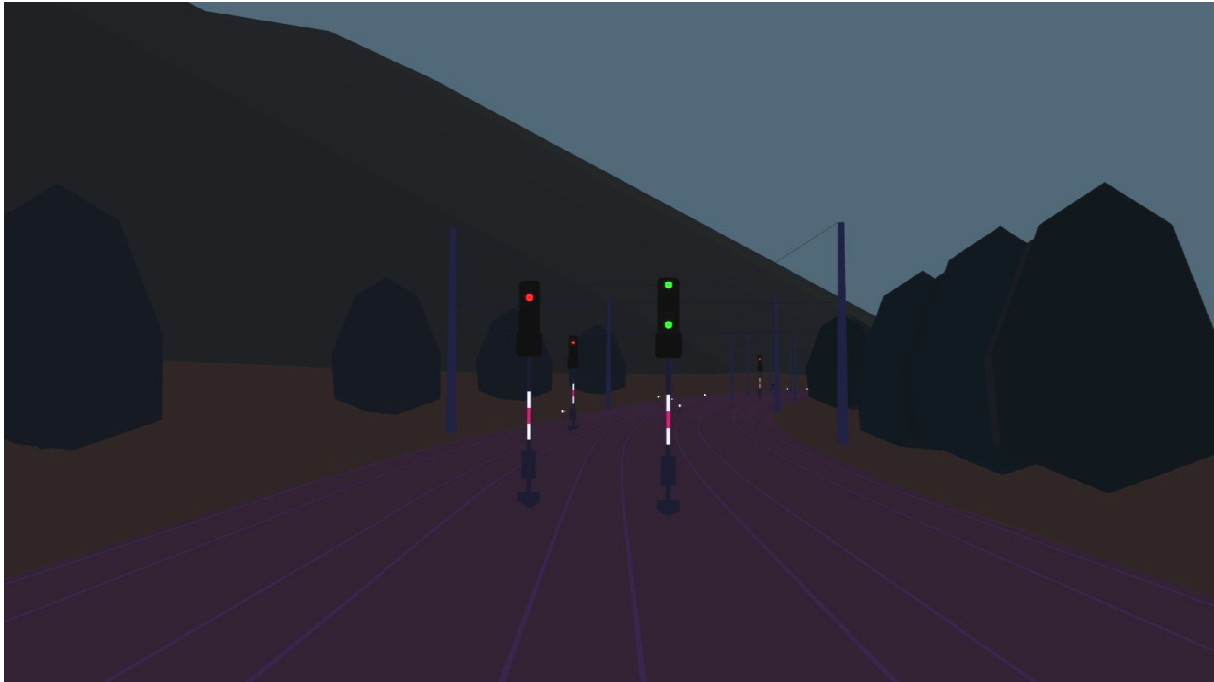
Es ist ein **Hauptsignal** (oben) mit kombiniertem **Vorsignal** (unten) erkennbar. Das Hauptsignal zeigt **Freie Fahrt** (ein grünes Licht) Dies bedeutet **Beschleunigen auf Streckengeschwindigkeit**.

Ein Blick in den Buchfahrplan verrät, dass diese hier 80km/h beträgt:

80	Mürzzuschlag	>116,2		5.16
	Sbl Spital a.S.	113,9		5.19
	Spital am Semmering	110,9	5.22	5.22
	Steinhaus Hst	108,1	5.25	5.26
	Sbl Semmering 1	105,4		5.29

Das Vorsignal zeigt jedoch **"Frei mit 60km/h erwarten"**. Dies bedeutet also, dass ab ca. 700-1000m nach diesem Signal nur mit 60km/h gefahren werden darf. Sobald Sie das Signal passieren drücken Sie **Indusi Wachsam** und achten darauf, dass der Leuchtmelder **1000Hz** zu leuchten beginnt. Nach dem Signal kann die angekündigte Geschwindigkeit von 60km/h angepeilt werden.

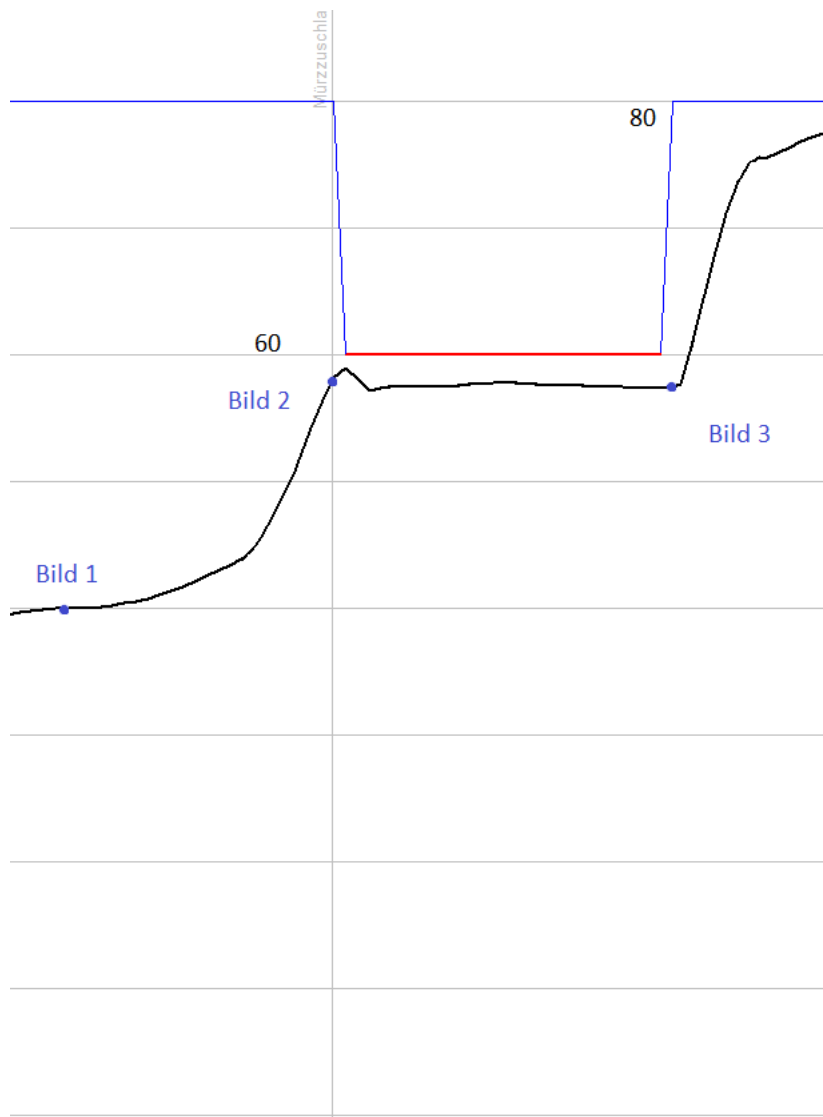
Bald wird das nächste Signal erreicht, welches wie angekündigt **"Frei mit 60km/h"** anzeigt.



Dies bedeutet also eine Weiterfahrt mit **60km/h** bis zum nächsten Hauptsignal oder bei einem **Ausfahrtsignal** (hier der Fall) bis zum **Ende des Weichenbereichs**. Warten Sie bis Sie die **letzte Weiche überfahren** haben und fahren dann noch **150m** (die Zuglänge) weiter, bis auch der letzte Wagen diesen Punkt überschritten hat.



Nun kann auf Streckengeschwindigkeit beschleunigt werden, also **80km/h**.

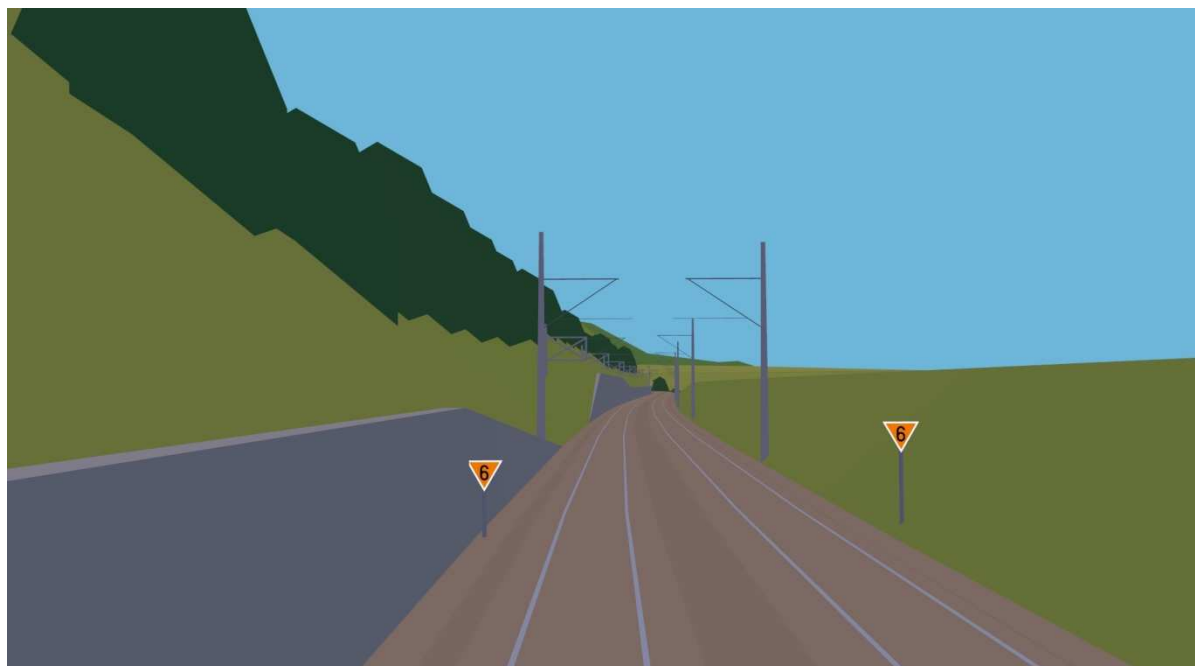


Ein Blick auf die Situation im Fahrtenschreiber zeigt wie eine mögliche Geschwindigkeitskurve aussehen könnte. Anfangs mit 40km/h, Beschleunigung auf 60km/h bis zum Ende des Weichenbereichs, anschließend beschleunigen auf Streckengeschwindigkeit.

Beispiel II - Passieren einer Langsamfahrstelle:

Bei Baustellen oder schlechter Gleislage kann es vorkommen, dass die im Buchfahrplan vermerkte Streckengeschwindigkeit nicht durchgehend gefahren werden kann. Dann wird eine sogenannte Langsamfahrstelle über einen Streckenabschnitt verhängt. Diese sind mit Ankündigungen gesichert, damit der Lokführer auch sicher bremsen kann.

Im folgenden Bild ist eine solche **Geschwindigkeitsankündigung** zu sehen:



Es ist eine "6" aufgedruckt, was eine Geschwindigkeit von **60km/h** bedeutet. Diese wird in ca. 500 bis 1000m gelten. Beim passieren der Geschwindigkeitsankündigung ist die Taste **Indusi Wachsam** zu betätigen. Anschließend kann langsam auf 60km/h abgebremst werden.

Nun ist der Anfang der Langsamfahrstelle erreicht, durch eine eckige Tafel mit der Aufschrift "6" signalisiert, welche mit 60km/h passiert wird.



Bereits wenige Meter danach ist das **Ende der Langsamfahrstelle** erreicht, was durch ein "E" signalisiert wird:



Ab dieser Tafel muss noch solange 60km/h gefahren werden, bis der **komplette Zug** diese Tafel überschritten hat, bei einem 150m langen Zug also noch 150m.



Ein Blick auf den Fahrtenschreiber zeigt eine mögliche Geschwindigkeitskurve.

6 Wirtschaftlicher Teil

6.1 Arbeitsaufteilung

Arbeit	Person
Dokumentation	Nestinger, Rudolf
Stahlgerüst gefertigt	Nestinger
Führerstandsteile befestigt	Nestinger
Abdeckplatten gefertigt	Nestinger
Abdeckplatten befestigt	Nestinger
Instandsetzung Schaltergruppe	Nestinger
Anschluss Schaltergruppe	Rudolf
Adaptierung Fahrschaltereinheit	Nestinger
Anschluss Fahrschaltereinheit	Rudolf
Adaptierung Tachograph	Nestinger, Rudolf
Anschluss Tachograph	Rudolf
Adaptierung Führerbremsventil	Nestinger
Anschluss Führerbremsventil	Rudolf
Adaptierung Zusatzbremsventil	Rudolf
Anschluss Zusatzbremsventil	Rudolf
Adaptierung Zugfunkgerät	Rudolf
Anschluss Zugfunkgerät	Rudolf
Adaptierung Nachlaufsteuerung	Nestinger
Anschluss Nachlaufsteuerung	Nestinger, Rudolf
Anschluss dig. Anzeigen	Rudolf
Anschluss analoge Anzeigen	Rudolf
Adaptierung Druckluftanzeigen	Nestinger
Anschluss Druckluftanzeigen	Rudolf
Relaiskarte entworfen	Rudolf
Relaiskarte bestückt	Rudolf
PWM Platine entworfen	Nestinger
PWM Platine bestückt	Nestinger
Elektr. Versorgung	Rudolf
Arduino MEGA Programm	Nestinger
Arduino UNO Programm	Nestinger
Visual Basic Programm	Nestinger
Verbindung mit ZUSI Anwendung	Nestinger

Tabelle 23: Arbeitsaufteilung

6.2 Kostenaufstellung

Firma	Datum	Bezeichnung	Betrag
Privat	01.07.2012	Führerstandsteile 1042	950,00
Hornbach	19.07.2012	Lenkrollen & Stahlprofile	70,90
Forstinger	25.07.2012	Farbe	17,99
Hornbach	14.08.2012	Lenkrollen	7,80
RS Components	10.12.2012	Arduino UNO & MEGA	67,98
Conrad	08.10.2012	Zahnräder	27,11
Hornbach	10.10.2012	Lenkrollen	3,90
RS Components	22.10.2012	IC IR2104	13,50
André Helbig	30.10.2012	Platinensteckverbinder	25,60
Conrad	31.10.2012	Gehäuse, Taster usw	50,44
Conrad	17.11.2012	Div.Kleinmaterial	87,36
RS Components	Div.	Verschiedene Elektronikbauteile	44,65
CNCshop.at	26.04.2013	Schrittmotor	37,90
Conrad	20.02.2013	Servos	15,97
			1.421,12

Tabelle 24: Kostenaufstellung

Literatur- und Quellenverzeichnis

de.wikipedia.org	ÖBB 1042
	Eisenbahnsignale in Österreich
	Druckluftbremse (Eisenbahn)
	Induktive Zugsicherung
www.lokifahrer.ch	Bremse Eisenbahntechnik
www.bremsenbude.de	Indirekte Bremse
www.zusi.de	Simulationsprogramm "ZUSI 2"

Abbildungsverzeichnis

Abbildung 01:	Kurbel zur Betätigung der Handbremse	1
Abbildung 02:	Oerlikon FDS2 Zusatzbremsventil.....	2
Abbildung 03:	Stellungen Zusatzbremsventil.....	2
Abbildung 04:	Bremsanlage.....	3
Abbildung 05:	Bremsschläuche zwischen Lok und Waggon	4
Abbildung 06:	Sifa Pedal auf der Bodenplatte.....	17
Abbildung 07:	Indusi Fahrzeugmagnet an einer Lokomotive (Quelle: S. Nestinger).....	18
Abbildung 08:	1042 44 in den 1960er Jahren (Quelle: Digitales Eisenbahnfotoarchiv) ...	20
Abbildung 09:	Die modernisierte 1142 598 in St. Pölten Hbf 2007 (Quelle: L. Rudolf)...	20
Abbildung 10:	Die modernisierte 1142 596 am 10.08.2012 (Quelle: S. Nestinger)	21
Abbildung 11:	Der Führerstand einer Lok der Reihe 1142 (Quelle: bahnvideos.net).....	21
Abbildung 12:	1142 680 in Tulln im August 2011 (Quelle: S. Nestinger).....	21
Abbildung 13:	Übersicht der mechanischen Bauteile.....	22
Abbildung 14:	Schaltergruppe vor Instandsetzung.....	23
Abbildung 15:	Schaltergruppe nach Instandsetzung	23
Abbildung 16:	Beschreibung der Schalterfunktionen.....	24
Abbildung 17:	Schalterstellungen Zugheizung.....	24
Abbildung 18:	Fahrschaltereinheit mit Schaltwalzen und Stufenanzeige	26
Abbildung 19:	Potentiometer mit Zahnrädern am Gestänge des Fahrschalters	26
Abbildung 20:	Hasler Tachograph	28
Abbildung 21:	Führerbremsventil am Fahrsimulator.....	29
Abbildung 22:	Zusatzbremsventil am Fahrsimulator.....	30
Abbildung 23:	Grundbild des analogen Zugfunkgerätes ZFM 90	31
Abbildung 24:	Das Stahlgerüst zu einem frühen Fertigungszeitpunkt (20.07.2012)	32
Abbildung 25:	Übertragungsweg der Dateneingabe.....	33
Abbildung 26:	Beschaltung der digitalen Eingänge	34
Abbildung 27:	Leuchtmelder Indusi 1000Hz	38
Abbildung 28:	Leuchtmelder Indusi Zugart P70-50.....	38
Abbildung 29:	Leuchtmelder Indusi Zugart D95-65.....	39
Abbildung 30:	Motorstromanzeigen 1 bis 4	40
Abbildung 31:	Bremsstromanzeigen.....	41
Abbildung 32:	Antrieb der Druckmessgeräte	42
Abbildung 33:	Die rotierende Scheibe der Stufenanzeige	43
Abbildung 34:	Schrittmotor an der Welle der Fahrschaltereinheit.....	43
Abbildung 36:	Schematische Darstellung des Datenflusses	44
Abbildung 37:	Schaltplan PWM-Platine.....	46
Abbildung 38:	Fertig bestückte PWM - Platine.....	48
Abbildung 39:	Schaltplan Relaiskarte.....	49
Abbildung 40:	Fertig bestückte Relaiskarte.....	50
Abbildung 41:	Rückseite des Fahrsimulators.....	52
Abbildung 42:	Schrittmotor mit Lüfter	52
Abbildung 43:	Fahrsimulator mit Abdeckplatte auf der Rückseite	53
Abbildung 44:	Beispiel einer Zugauswahl in Zusi nach dem Laden eines Fahrplans.....	55
Abbildung 45:	Zugfahrt auf der Linken Rheinstrecke	55
Abbildung 46:	Aufnahme der Linken Rheinstrecke	56
Abbildung 47:	Übertragungsprogramm am PC	57
Abbildung 48:	Linker Programmteil	58

Abbildung 49:	Mittlerer Programmteil	59
Abbildung 50:	Fahrtüberwachung.....	60
Abbildung 51:	Debugteil.....	61
Abbildung 52:	Programm in der Arduino-Programmieroberfläche	61
Abbildung 53:	Strecke laden.....	62
Abbildung 54:	Fahrplan laden.....	63
Abbildung 55:	Gestartete Zugfahrt.....	63
Abbildung 56:	Zusi-Einstellungen	64
Abbildung 57:	Verbinden des Übertragungsprogramms	64
Abbildung 58:	Aktivierung der Zusi-Datenausgabe	65
Abbildung 59:	Fahrsimulator verbunden und einsatzbereit	66

Tabellenverzeichnis

Tabelle 01:	Österreichische Signale.....	6
Tabelle 02:	Ersatzsignal Österreich	6
Tabelle 03:	Schutzsignale Österreich.....	7
Tabelle 04:	Formsignale Österreich.....	9
Tabelle 05:	Geschwindigkeitstafeln Österreich.....	9
Tabelle 06:	Geschwindigkeitsanzeiger Österreich	10
Tabelle 07:	Langsamfahrstellen Österreich.....	10
Tabelle 08:	Lichtsignale Deutschland.....	11
Tabelle 09:	Ersatzsignal Deutschland.....	12
Tabelle 10:	Formsignale Deutschland	13
Tabelle 11:	Geschwindigkeitstafeln Deutschland	13
Tabelle 12:	Geschwindigkeitsanzeiger Deutschland	14
Tabelle 13:	Langsamfahrstelle Deutschland	15
Tabelle 14:	Haltetafel	15
Tabelle 15:	Praxisbeispiele Signale	16
Tabelle 16:	Verbindung der Zugsammelschiene (vorne).....	19
Tabelle 17:	Technische Daten ÖBB 1042	22
Tabelle 18:	Beschreibung der Anzeige des Zugfunks	31
Tabelle 19:	Liste Digitaleingänge.....	35
Tabelle 20:	Anschlussbelegung PWM-Platine.....	48
Tabelle 21:	Anschlussbelegung Relaiskarte.....	51
Tabelle 22:	Nötige Programme	62
Tabelle 23:	Arbeitsaufteilung	72
Tabelle 24:	Kostenaufstellung	73

Glossar

Arduino

Mikroprozessor, der Werte vom Fahrpult einliest und Werte an das Fahrpult ausgibt.

Aufrüsten

Als Aufrüstvorgang wird der Startvorgang einer Lokomotive bezeichnet.

Ausfahrtsignal

Als Ausfahrtsignal wird ein Hauptsignal im Bahnhof bezeichnet. Es regelt die Ausfahrt des Zuges aus dem Bahnhof.

Blocksignal

Blocksignale sind Hauptsignale, welche die freie Strecke in Blockabschnitte unterteilen. Ist im Abschnitt gerade kein Zug, dann zeigt das Blocksignal üblicherweise Frei. Alle anderen Signale zeigen im Ruhezustand Halt.

Buchfahrplan

Der Buchfahrplan versorgt den Lokführer mit allen relevanten Streckendaten (km, Zeit, v-max, Bahnhöfe, Zugfunknummern, etc.).

DB

Eisenbahnunternehmen in Deutschland, Deutsche Bahn AG

Direkte Bremse

Dies ist eine Druckluftbremse und wird auch noch Zusatzbremse genannt. Diese umfasst ausschließlich das Bremsen an der Lokomotive selbst.

Einfahrtsignal

Als Einfahrtsignal wird ein Hauptsignal bezeichnet, welches vor einem Bahnhof steht. Finden zum Beispiel im Bahnhof Verschubarbeiten statt, so kann der Fahrdienstleiter einen kommenden Zug vor dem Bahnhof anhalten.

Formsignal

Alte Signale, die rein mechanisch funktionieren, werden Formsignale genannt.

Hauptsignal

Das Signal zeigt dem Lokführer an, ob und mit welcher Geschwindigkeit in einen Gleisabschnitt gefahren werden darf.

Indirekte Bremse

Die indirekte Bremse bremst den kompletten Zug über die Hauptluftleitung. Sie wird auch als Zugbremse bezeichnet.

Indusi

Indusi ist die Kurzform für Induktive Zugsicherung. Passive Schwingkreise auf den Gleisen und aktive Schwingkreise auf der Lokomotive mit jeweils unterschiedlichen Resonanzfrequenzen beeinflussen sich gegenseitig und sorgen für Informationsübertragung.

ÖBB

Eisenbahnunternehmen in Österreich, Österreichische Bundesbahnen.

PZB

Punktförmige Zugbeeinflussung, ein anderer Begriff für die Indusi. Heutige Systeme verwenden diesen Begriff als Bezeichnung.

PZB Befehl

Dieser Taster darf nur bei Vorbeifahrt an einem gestörten oder HALT-zeigenden Signal und auf besondere Anordnung betätigt werden.

PZB Frei

Dieser Taster befreit den Zug aus einer Zwangsbremmung, sobald diese mindestens 8 Sekunden lang aktiv war.

PZB Wachsam

Dieser Taster muss betätigt werden, wenn ein aktiver 1000Hz Magnet überfahren wird.

Schnellbremsung

Zieht der Lokführer das Führerbremmsventil bis zum Anschlag zu sich, wird eine Schnellbremsung (Notbremsung) eingeleitet.

Schutzsignal

Ein Schutzsignal unterteilt ein Gleis im Bahnhof in mehrere Gleisabschnitte. Dadurch können an einem Bahnsteig mehrere Züge einfahren.

Sifa

Die Sicherheitsfahrerschaltung (kurz: Sifa) ist eine Sicherheitseinrichtung in Triebfahrzeugen. Durch regelmäßiges Loslassen des Pedals erhält die Sifa die Rückmeldung, dass der Lokführer noch reaktionsfähig ist.

Signalgrundbegriffe

Die Signalgrundbegriffe lassen sich rein mit den Lichtern der Signale anzeigen. In Österreich sind diese Halt, Frei mit 40km/h, Frei mit 60km/h und Frei mit Streckenhöchstgeschwindigkeit.

Soundthesizer

Der Soundthesizer ist eine externe Anwendung von ZUSI, die über den TCP Server mit dem Hauptprogramm verbunden wird. Dieses Programm sorgt für die Ausgabe der realistischen Fahrgeräusche und Loksounds.

Streckenhöchstgeschwindigkeit

Die ohne Beschränkung maximal zulässige Geschwindigkeit für einen Streckenabschnitt wird Streckenhöchstgeschwindigkeit genannt. Sie ist im Buchfahrplan angegeben.

Stromabnehmer

Der Stromabnehmer (kurz: Bügel) befindet sich auf dem Dach der Lokomotive und greift die Fahrdrabtspannung ab. Im Normalfall wird immer der hintere Stromabnehmer ausgefahren, nur in Ausnahmefällen (z.B. Sondertransporte) wird der Vordere ausgefahren. Die Stromabnehmer sind zusätzlich mit Eisbrechern ausgestattet um bei Eisbildung den Kontakt zur Fahrleitung nicht zu verlieren.

TCP Server

Server des Programms ZUSI, welches den Datenaustausch von externen Programmen mit ZUSI ermöglicht.

Tfz

Kurzform für Triebfahrzeug

Totmannknopf

Totmannknopf ist eine andere Bezeichnung für das Sifa Pedal der Sicherheitsfahrerschaltung.

Triebfahrzeugführer

Die offizielle Bezeichnung des Berufes Lokführer lautet Triebfahrzeugführer (kurz Tfzf).

Ursprungslackierung

Als Ursprungslackierung bezeichnet man jenes Farbschema einer Lokomotive, welches sie bei der Auslieferung hatte.

Valousek Design

Lackierungsvariante für Triebfahrzeuge der ÖBB in den 1990er Jahren.

Vorsignal

Ein Vorsignal hat eine quadratische Form und steht im Bremswegabstand zum nächsten Signal. Dabei kündigt es jenes Signalbild an, welches beim Hauptsignal zu erwarten ist.

ZUSI

ZUSI ist eine Zugsimulationssoftware für Privatanwender. Es wurde vom Braunschweiger Carsten Hölscher programmiert und kann im Internet erworben werden.

Anhang A. Quellcodes

Anhang A1: Visual Basic Programm "Connector Fahrpult 1042"

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 1
'Die Grundstruktur des Programms geht auf Andreas Karg (TCP-Übertragung) und Jens Eggert aus dem Zusi-Forum zurück
'und wurde für die Diplomarbeit um einige Funktionen, sowie über eine Übertragung vom Arduino zum PC erweitert.

Imports System
Imports System.IO.Ports
Imports System.Threading
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Drawing
Imports System.Text
Imports System.Windows.Forms
Imports Zusi_Datenausgabe

' ZusiTCPDemoApp
' This example shows basic usage of Andreas Karg's Zusi TCP interface for .Net.
' It is published under the GNU General Public License v3.0. Base your own work on it, play around, do what you want. :-)
,
,
' Using the interface requires three steps:
' - Write one or more handler methods
' - Create a ZusiTcpConn object, passing basic parameters
' - Tell your ZusiTcpConn object what measures you want to receive
,
' Everything else is explained below.
' ZusiTCPDemoAppVBA
' Translated C#-Project into VisualBasic.
'Declaration

Public Class CMainWindow
'Deklaration der Bytes für die Leuchtmelder
Dim LmPzb1000 As Byte
Dim LmPzbu As Byte
Dim LmPzbo As Byte
Dim LmHS As Byte
Dim LmTuer As Byte
Dim LmElBr As Byte
Dim Span As Byte

'Es folgen die Bytes für die Wertausgabe an den Arduino:
Dim V As Byte
Dim L As Byte
Dim B As Byte
Dim H As Byte
Dim M As Byte
Dim N As Byte
Dim X As Byte
Dim Y As Byte
Dim O As Byte
Dim S As Byte
Dim Lm1 As Byte

Dim ElBr As Integer

'Variablen für die Steuerung des Führerbremventils
Dim HLsoll As Integer
Dim HList As Integer
Dim Differenz As Integer

'Hilfsvariablen für die Soundausgabe der Bremsanlage
Dim Abspiel As Integer
Dim Abspiell As Integer
Dim Abspiellos As Integer
Dim Fbv As Integer

Dim Motorstrom As Integer
Dim Fs As Integer

```

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 2
Dim zeitdiff As Integer
Dim OutChar(2) As Byte
Dim tastendruck As Integer
Dim letztestunde As Integer

'Input Bytes a,b,c,d "ax", "bx" stellen die Werte des vorhergehenden Zyklus dar und werden zur Erfassung
von
'Zustandsänderungen genutzt.
Dim INa As Byte
Dim INax As Byte
Dim INb As Byte
Dim INbx As Byte
Dim INc As Byte
Dim INd As Byte
Dim Regler As Integer
Dim Cnt As Integer
Dim Zugkraft As Integer
Dim Bremsstrom As Integer
Dim Tueren As Integer
Dim FSsoll As Integer
Dim FSist As Integer
Dim Everz As Integer
Dim count As Integer

' Deklaration des "ZusiTcpConn" Objekts:
Private MyTCPConnection As ZusiTcpConn

Public CMainWindow()
Public Sub New()
    InitializeComponent()

    'Die Connection Class wird erstellt:
    'Hier wird der Name des Programms definiert, wie es im TCP Server aussieht.

    MyTCPConnection = New ZusiTcpConn( _
    "Connector Fahrpult 1042", _
    ClientPriority.High, _
    New ReceiveEvent(Of Single)(AddressOf HandleIncomingData), _
    Nothing, _
    Nothing _
    )

    ' The name of this application (Shows up on the server's list)
    ' The priority with which the server should treat you
    ' A delegate method for the connection class to call when it receives float data (may be null)
    ' A delegate method for the connection class to call when it receives string data (may be null)
    ' A delegate method for the connection class to call when it receives byte[] data (may be null)

    'Hier werden nun jene Informationen aufgelistet, die von Zusi angefordert werden sollen.
    'Eine Liste mit den Zuordnungen der IDs findet sich im Anhang.

    MyTCPConnection.RequestData(2561) 'Geschwindigkeit
    MyTCPConnection.RequestData(2562) 'Druck Hauptluftleitung
    MyTCPConnection.RequestData(2563) 'Druck Bremszylinder
    MyTCPConnection.RequestData(2564) 'Druck HL Behälter
    MyTCPConnection.RequestData(2565) 'Zugkraft gesamt
    MyTCPConnection.RequestData(2567) 'Strom
    MyTCPConnection.RequestData(2570) 'Stunde
    MyTCPConnection.RequestData(2571) 'Minute
    MyTCPConnection.RequestData(2576) 'Schaltwerk
    MyTCPConnection.RequestData(2580) 'LM PZB 1000Hz
    MyTCPConnection.RequestData(2581) 'LM PZB 500Hz
    MyTCPConnection.RequestData(2585) 'M PZB Zugart 0
    MyTCPConnection.RequestData(2597) 'LM Hauptschalter
    MyTCPConnection.RequestData(2607) 'LM Türen
    MyTCPConnection.RequestData(2611) 'Fahrschalter
    MyTCPConnection.RequestData(2612) 'Fbv
    MyTCPConnection.RequestData(2645) 'Strecken km
    MyTCPConnection.RequestData(2646) 'Tueren
    MyTCPConnection.RequestData(2654) 'BrH
    MyTCPConnection.RequestData(2660) 'aktuelle Höchstgeschwindigkeit

```

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 3
    MyTCPConnection.RequestData(2661) 'nächste Höchstgeschwindigkeit
    MyTCPConnection.RequestData(2662) 'nächste Höchstgeschwindigkeit km
    MyTCPConnection.RequestData(2663) 'nächster Halt km
    MyTCPConnection.RequestData(2664) 'nächster Halt

End Sub

'Es folgt die Verbindungsroutine, nach dem Klicken auf den "Verbinden" Button
Private Declare Sub keybd_event Lib "user32" ( _
    ByVal bVk As Byte, _
    ByVal bScan As Byte, _
    ByVal dwFlags As Long, _
    ByVal dwExtraInfo As Long _
)

Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnConnect.Click
    'Ist die Verbindung getrennt,...
    If (MyTCPConnection.ConnectionState = Zusi_Datenausgabe.ConnectionState.Disconnected) Then
        '...wird versucht...
        Try
            '...eine Verbindung mit den im Hauptfenster eingetragenen Daten für Hostname und Port
            herzustellen:
            MyTCPConnection.Connect(TbServer.Text, Convert.ToInt32(TbPort.Text))

            ' Ist die Verbindung erfolgreich, ändert sich die Beschriftung des Buttons auf "Trennen".
            BtnConnect.Text = "Trennen"

            ' Kann keine Verbindung hergestellt werden,
            Catch ex As ZusiTcpException
                ' ... wird eine Fehlermeldung angezeigt.
                MessageBox.Show(String.Format("Ein Fehler trat bei der Verbindung auf: {0}", ex.Message))
            End Try

            ' Wenn bereits verbunden ist oder ein Error auftrag, wird die Verbindung zurückgesetzt:
        Else
            MyTCPConnection.Disconnect()

            ' und der Button wird wieder auf "Verbinden" gesetzt
            BtnConnect.Text = "Verbinden"
        End If
    End Sub

'Wenn Daten ankommen, wird dieses Subprogramm aufgerufen:
Private Sub HandleIncomingData(ByVal dataSet As DataSet(Of Single))

    If Serial.IsOpen Then 'Die Werte werden über die TCP Verbindung ausgelesen und in lokale Variablen
    gespeichert:

        Select Case dataSet.Id
            Case 2561 'Geschwindigkeit
                V = CByte(Math.Round(0.44 * vkorrr.Value / 100 * dataSet.Value, 0))
                vistsig.Text = Math.Round(dataSet.Value)
                vsig.Value = Math.Round(dataSet.Value)

            Case 2562 'HL (Hauptluftleitung)
                HList = Math.Round(100 * dataSet.Value, 0) + 1
                H = CByte(Math.Round(9.5 * dataSet.Value, 0))

            Case 2563 'Druck Bremszylinder
                B = CByte(Math.Round(20 * dataSet.Value, 0))
                bzd.Text = Math.Round(B)

            Case 2564 'HB (Hauptluftbehälter)
                L = CByte(Math.Round(10 * dataSet.Value, 0))

            Case 2565 'Zugkraft/Motorstrom
                If Math.Round(dataSet.Value, 0) > 1 Then
                    M = CByte(Math.Round(0.15 * dataSet.Value, 0))
                    Motorstromsig.Value = Math.Round(dataSet.Value, 0)
                Else
                    M = 0
                End If
        End Select
    End If

```

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 4
    If Math.Round(dataSet.Value, 0) < -1 Then
        0 = CByte(Math.Round(-0.4 * dataSet.Value, 0))
        Bremsstromsig.Value = 0
    Else
        0 = 0
    End If

    Case 2567 'Primaerstrom
        If Math.Round(dataSet.Value, 0) > 1 Then 'Von Zusi wird die Zugkraft
übertragen, welche auch negativ werden kann. Daher nur für positive Werte.
            N = CByte(Math.Round(0.07 * dataSet.Value, 0))

            Primstromsig.Value = N
            primstromtxtsig.Text = N
        Else
            N = CByte(Math.Round(1)) 'Primärstrom
            Primstromsig.Value = Math.Round(0)
        End If

        'Case 2568 'Fahrdrabtspannung
        'If dataSet.Value < 2 Then
        'If Stab = 1 Then Stab = 0
        'End If

    Case 2570 'Stunde
        If (dataSet.Value > 0) Then
            X = CByte(Math.Round(zeitdiff + dataSet.Value, 0))
            Label4.Text = CByte(Math.Round(zeitdiff + dataSet.Value, 0))

            If CheckBox1.Checked = True Then 'Da Zusi die Stunde im 12h Format ausgibt, muss für
Uhrzeiten nach 12h die Zeit manuell angepasst werden.
                zeitdiff = 11.5
            Else
                zeitdiff = -0.5 'kompensiert die Rundungsdifferenz
            End If
        End If

    Case 2571 'Minute
        Y = CByte(Math.Round(5 * dataSet.Value, 0))
        Label3.Text = CByte(Math.Round(5 * dataSet.Value, 0))

    Case 2576 'Schaltwerk 2576 (FS2611)
        If (Math.Round(dataSet.Value, 0) >= 0) Then
            S = CByte(Math.Round(dataSet.Value, 0))
            swsig.Text = S
        End If

    Case 2585 'M PZB Zugart 0
        lmpsig.Checked = dataSet.Value
        LmPzb0 = CByte(dataSet.Value)

    Case 2583 'LM PZB Zugart U
        lmgsig.Checked = dataSet.Value
        LmPzbU = CByte(dataSet.Value) << 1
    Case 2580 'LM PZB 1000Hz
        lm1000sig.Checked = dataSet.Value
        LmPzb1000 = CByte(dataSet.Value) << 2
    Case 2597 'Lm Hauptschalter
        lmHssig.Checked = 1 - dataSet.Value
        LmHS = 1 - CByte(dataSet.Value) << 3
    Case 2607 'Lm Türen
        LmTuer = CByte(dataSet.Value) << 4

    Case 2611 'Fahrschalter
        FSist = Math.Round(dataSet.Value, 0)
        Fsig.Text = FSist
        If CByte(Math.Round(dataSet.Value, 0)) < 15 Then
            E1Br = 1
        Else
            E1Br = 0
        End If
        LmE1Br = CByte(E1Br) << 5

    Case 2612 'Fbv

```

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 5
    If dataSet.Value > 0 Then
        Fbv = CByte(Math.Round(dataSet.Value, 0))
        Label5.Text = Fbv
    End If

    Case 2645 'Strecken km
        If (dataSet.Value > 0) Then
            kmsig.Text = (Math.Round(dataSet.Value) / 1000)
        End If

    Case 2646 'Tueren
        Tueren = Math.Round(dataSet.Value * 1.0E+45)
        If (Tueren = 1) Then
            tuersig.Text = "offen"
        ElseIf (Tueren = 4) Then
            tuersig.Text = "schließbereit"
        ElseIf (Tueren = 6) Then
            tuersig.Text = "schließen"
        ElseIf (Tueren = 7) Then
            tuersig.Text = "zu"
        ElseIf (Tueren = 8) Then
            tuersig.Text = "Abfahrt"
        ElseIf (Tueren = 10) Then
            tuersig.Text = "verriegelt"
        End If

    Case 2654 'BrH
        brhsig.Text = dataSet.Value

    Case 2660 'Aktuelle HG
        vmaxsig.Text = dataSet.Value

    Case 2661 'Nächste HG
        If (dataSet.Value >= 0) Then
            vmaxnsig.Text = dataSet.Value
        Else
            vmaxnsig.Text = "-"
        End If

    Case 2662 'Nächste HG km
        If (vmaxnsig.Text <> "-") Then
            vmaxnkmsig.Text = (Math.Round(dataSet.Value) / 1000)
        Else
            vmaxnkmsig.Text = "-"
        End If

    Case 2663 'Nächster Bhf km
        nbhfkmsig.Text = (Math.Round(dataSet.Value) / 1000)

    End Select
    Lm1 = LmPzb0 Or LmPzbU Or LmPzb1000 Or LmHS Or LmTuer Or LmElBr Or Span 'Die
    Digitalwerte werden in ein Byte gepackt

    End If
End Sub
'Das Form_Load-Ereignis
Private Sub CMainWindow_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.
Load
    AddCom2Combo(CbCom)
End Sub
'Hier wird eine Liste der vorhandenen COM-Ports erstellt
Sub AddCom2Combo(ByVal cbPort As ComboBox)
    ' Eine Liste der vorhandenen COM Ports wird angefordert
    Dim ports As String() = SerialPort.GetPortNames()
    ' Oben wird folgendes angezeigt:
    cbPort.Text = "The following serial ports were found:"
    ' Jedes Element kommt in ein Feld der ComboBox
    Dim port As String

```

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 6
    For Each port In ports
        cbPort.Items.Add(port)
    Next port
    ' Zuerst wird das oberste Element ausgewählt:
    cbPort.SelectedIndex = 0
End Sub
'Hier wird die Verbindung zu unserer Ausgabehardware hergestellt.
Private Sub ComCon_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComCon. Click
    If Not Serial.IsOpen Then
        Serial.PortName = CbCom.SelectedItem
        Serial.Open()
        ComCon.Text = "Trennen"
        Serial.WriteLine("ww")
        System.Threading.Thread.Sleep(500)
        If Serial.BytesToRead > 0 Then
            If Not Serial.ReadLine() = "I am a drivers desk" & System.Text.Encoding.UTF8.GetString({13})
            Then
                MessageBox.Show("An dem gewählten Port scheint das falsche Gerät angeschlossen zu sein!")
            )
                Serial.Close()
                ComCon.Text = "Verbinden"
            End If
        Else
            MessageBox.Show("An dem gewählten Port scheint kein Gerät angeschlossen zu sein!")
            Serial.Close()
            ComCon.Text = "Verbinden"
        End If
    Else
        Serial.Close()
        ComCon.Text = "Verbinden"
    End If
End Sub

'Hier werden die Informationen an den Arduino weitergeleitet.
'Alle 215ms werden die Daten ausgegeben:
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

    FSadj.Text = Analogwert.Value

    If Serial.IsOpen Then

        Serial.DiscardOutBuffer() 'Der Puffer wird geleert
        Serial.Write("P") 'Es wird ein Buchstabe zur Identifikation gesendet
        Serial.Write(Convert.ToChar(Lm1)) 'Die Information wird gesendet.
        01.Text = Lm1 'Hier wird die Info zusätzlich auf der Programmoberfläche
angezeigt

        System.Threading.Thread.Sleep(8) 'Zwischen den Bytes findet eine kurze Verzögerung statt.

        Serial.DiscardOutBuffer()
        Serial.Write("V")
        Serial.Write(Convert.ToChar(V))
        02.Text = V

        System.Threading.Thread.Sleep(8)

        Serial.DiscardOutBuffer()
        Serial.Write("H")
        Serial.Write(Convert.ToChar(H))
        03.Text = H

        System.Threading.Thread.Sleep(8)

        Serial.DiscardOutBuffer()
        Serial.Write("B")
        Serial.Write(Convert.ToChar(B))
        04.Text = B

        System.Threading.Thread.Sleep(8)

        Serial.DiscardOutBuffer()
        Serial.Write("L")
    End If
End Sub

```

```
E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb 7
    Serial.Write(Convert.ToChar(L))
    O5.Text = L

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("M")
    Serial.Write(Convert.ToChar(M))
    O6.Text = M

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("O")
    Serial.Write(Convert.ToChar(O))
    O7.Text = O

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("N")
    Serial.Write(Convert.ToChar(N))
    O8.Text = N

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("X")
    Serial.Write(Convert.ToChar(X))

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("Y")
    Serial.Write(Convert.ToChar(Y))

    System.Threading.Thread.Sleep(8)

    Serial.DiscardOutBuffer()
    Serial.Write("S")
    Serial.Write(Convert.ToChar(S))

    System.Threading.Thread.Sleep(10)

    'Einlesen der Werte vom Arduino
    Serial.WriteLine("ii")    "'ii" für "input"
    Serial.DiscardInBuffer()
    System.Threading.Thread.Sleep(20)

    If Serial.ReadLine() = "readytosend" & System.Text.Encoding.UTF8.GetString({13}) Then
        System.Threading.Thread.Sleep(15)

        If (Serial.ReadByte() = 255) Then
            INa = Serial.ReadByte()    'Byte 0 - Digitaleingabe I
            INb = Serial.ReadByte()    'Byte 1 - Digitaleingabe II
            INc = Serial.ReadByte()    'Byte 2 - Analogwert FbV
            INd = Serial.ReadByte()    'Byte 3 - Analogwert Fahrschalter
        Else
            Serial.DiscardInBuffer()
        End If

        Byte1.Text = INa
        Byte2.Text = INb
        Byte3.Text = INc * 2.2 - 20
        zwei.Text = INc                'Ausgabe am Bildschirm FbV
        Byte4.Text = INd                'Ausgabe am Bildschirm FS
        Bremsventilsig.Value = INc     'Ausgabe graphisch FbV
        Fahrschaltersig.Value = FSsoll 'Ausgabe graphisch FS
    End If
```


E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainwindow.vb

8

```

Else
    System.Threading.Thread.Sleep(4)
    Serial.DiscardInBuffer()
End If

If (zusi.Checked = True) Then

    'Hier werden die digitalen Inputs an Zusi weitergeleitet. Dies erfolgt durch virtuelle
    Tastendrücke,
    'da Zusi keine andere Form der Eingabe ermöglicht.
    'Da es im Programm sowohl tastende, als auch rastende Schalter gibt, gibt es verschiedene
    Senderoutinen.
    'Bei Tastern im Programm wird die virtuelle Taste solange gehalten, wie der Taster am Pult,
    bei Schaltern
    'in Zusi wird bei jedem Zustandswechsel am Pult ein Impuls gesendet.
    'Länger gedrückte Taster müssen über die Funktion "keybd_event" abgewickelt werden.

    If (INa And 1) > 0 Then
        'Bit 0.0: Wenn der aktuelle Zustand größer als 0 ist,
        If (INax And 1) = 0 Then
            'und der Zustand im vorherigen Durchlauf = 0 ist,
            SendKeys.Send("K")
            'dann wird ein Impuls gesendet, Key "K", welcher den
            Zustand des Schalters in Zusi umschaltet.
            lueftersig.Checked = 1
            'Damit auch im Übertragungsprogramm erkannt werden kann,
            welcher Zustand aktiv ist, wird dies auch auf dem Bildschirm ausgegeben.
        End If
    End If
    If (INa And 1) = 0 Then
        'Hier wird auf den Zustandswechsel in die andere
        Richtung abgefragt, von 1 auf 0
        If (INax And 1) > 0 Then
            SendKeys.Send("K")
            'Lüfter 1=>0
            lueftersig.Checked = 0
            'Zustandswechsel in der Visualisierung
        End If
    End If

    If (INa And 2) > 0 Then
        'Bit 0.1
        If (INax And 2) = 0 Then
            SendKeys.Send("B")
            'Befehl 0=>1, Key "NUM 0"
            befehlsig.Checked = 1
        End If
    End If
    If (INa And 2) = 0 Then
        If (INax And 2) > 0 Then
            SendKeys.Send("B")
            'Befehl 1=>0
            befehlsig.Checked = 0
        End If
    End If

    If (INa And 4) > 0 Then
        'Bit 0.2
        If (INax And 4) = 0 Then
            keybd_event(&H31, 0, 0, 0)
            'Frei 0=>1 - Key "NUM ,"
            freisig.Checked = 1
        End If
    End If
    If (INa And 4) = 0 Then
        If (INax And 4) > 0 Then
            keybd_event(&H31, 0, &H2, 0)
            'Frei 1=>0
            freisig.Checked = 0
        End If
    End If

    If (INa And 8) > 0 Then
        'Bit 0.3
        If (INax And 8) = 0 Then
            keybd_event(&HD, 0, 0, 0)
            'Wachsam 0=>1 - Key "Enter"
            wachsamsg.Checked = 1
        End If
    End If
    If (INa And 8) = 0 Then
        If (INax And 8) > 0 Then
            keybd_event(&HD, 0, &H2, 0)
            'Wachsam 1=>0
            wachsamsg.Checked = 0
        End If
    End If
End If

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb

9

```

If (INa And 16) > 0 Then If (INax And 16) = 0 Then SendKeys.Send("H") 'HSum Bit 0.4

If (INa And 32) > 0 Then          'Bit 0.5 NEU
  If (INax And 32) = 0 Then
    SendKeys.Send(" ")          'SIFA 0 => 1 - Key "Leer" senden
    sifasig.Checked = 1
  End If
End If
If (INa And 32) = 0 Then
  If (INax And 32) > 0 Then
    SendKeys.Send(" ")          'SIFA 1 => 0 - Key "Leer" senden
    sifasig.Checked = 0
  End If
End If

If (INa And 64) = 0 Then
  If (INax And 64) > 0 Then      'Bit 0.6
    keybd_event(&H6B, 0, 0, 0) 'Horn 0=>1 - Key "NUM +"
    hornsig.Checked = 1
  End If
End If
If (INa And 64) > 0 Then
  If (INax And 64) = 0 Then
    keybd_event(&H6B, 0, &H2, 0) 'Horn 1=>0
    hornsig.Checked = 0
  End If
End If

If (INa And 128) = 0 Then
  If (INax And 128) > 0 Then    'Bit 0.7
    keybd_event(&H48, 0, 0, 0) 'Sifa 0=>1 - Key "Leer"
    hornsig.Checked = 1
  End If
End If
If (INa And 128) > 0 Then
  If (INa And 128) = 0 Then
    keybd_event(&H48, 0, &H2, 0) 'Sifa 1=>0
    hornsig.Checked = 0
  End If
End If

If (INb And 1) > 0 Then        'Bit 1.0
  If (INbx And 1) = 0 Then
    SendKeys.Send("O")
    loesig.Checked = 1          'Abfrage auf = 0
                                'Bremsen lösen 0=>1, Key "+"
                                'Zustandswechsel in der Visualisierung
  End If
End If
If (INb And 1) = 0 Then
  If (INbx And 1) > 0 Then
    SendKeys.Send("L")
    loesig.Checked = 0          'Abfrage auf > 0
                                'Bremsen in Mittelstellung bringen, Key "#"
                                'Zustandswechsel in der Visualisierung
  End If
End If

If (INb And 2) > 0 Then        'Bit 1.1
  If (INbx And 2) = 0 Then
    keybd_event(&H4C, 0, 0, 0) 'Anlegen 0=>1 - Key "L"
    ansig.Checked = 1
  End If
End If
If (INb And 2) = 0 Then
  If (INbx And 2) > 0 Then
    keybd_event(&H4C, 0, &H2, 0) 'Anlegen 1=>0
    ansig.Checked = 0
  End If
End If

If (INb And 4) > 0 Then        'Bit 1.2
  If (INbx And 4) = 0 Then
    keybd_event(&H6D, 0, 0, 0) 'Sand 0=>1 - Key "NUM ,"
    sandsig.Checked = 1
  End If
End If

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainwindow.vb

10

```

End If
End If
If (INb And 4) = 0 Then
  If (INbx And 4) > 0 Then
    keybd_event(&H6D, 0, &H2, 0) 'Sand 1=>0
    sandsig.Checked = 0
  End If
End If

If (INb And 8) > 0 Then 'Stromabnehmer Bit 1.3
  stabsig.Checked = True
  If (count < 25) Then 'Einschaltverzögerung
    count = count + 1
  Else
    Span = CByte(1) << 6
    spansig.Checked = True
  End If
Else
  Span = CByte(0) << 6
  count = 0
  stabsig.Checked = False
  spansig.Checked = False
End If

If INd > 50 Then INd = 0
INax = INa
INbx = INb

hlistsig.Value = HList 'Die Steuerung des FbVs
Hlsoll = INC * 2.2 - 20
If (Hlsoll < hlsollsig.Maximum) Then
  'hlsollsig.Value = Hlsoll
End If

Differenz = Hlsoll - HList
Label16.Text = Differenz

If INC < 250 Then
  If Differenz > 5 And Fbv <> 1 Then
    SendKeys.Send("9")

    ElseIf Differenz < (-5) And Fbv <> 4 Then
      SendKeys.Send("3")
    ElseIf Differenz < 5 And Differenz > (-5) And Fbv <> 3 Then
      SendKeys.Send("6")
      Abspiel = 0
      Abspiell = 0
    End If
Else
  'SendKeys.Send("0")
End If

'Wertegelättung für Fahrschalter:
'If (INd <= 16) And (INd >= 14) Then
'FSsoll = 15
'Else
'FSsoll = INd + 1
'End If

If FSist < FSsoll Then 'Fahrschalter
  'SendKeys.Send("{UP}")
  SendKeys.Send("V")
End If

If FSist > FSsoll Then
  'SendKeys.Send("{DOWN}")
  SendKeys.Send("B")
End If

FSsollsig.Text = FSsoll

```

E:\Zusi\Zusi\ZusiTCPDemoAppVBA_Timer\CMainWindow.vb

11

```
    If Differenz < -8 And Abspiel = 0 Then          'Soundausgabe für das FbV
        My.Computer.Audio.Play("C:\Daten\zugbr_an_2.wav")
        Abspiel = 1
    End If

    If Differenz < -40 And Abspiel1 = 0 Then
        My.Computer.Audio.Play("C:\Daten\zugbr_an_1.wav")
        Abspiel1 = 1
    End If

    If Differenz > 40 And Abspiel = 0 Then
        My.Computer.Audio.Play("C:\Daten\zugbr_lose_2.wav")
        Abspiel = 1
    End If

    If Hlsoll > 500 And Abspiellos = 0 Then
        My.Computer.Audio.Play("C:\Daten\zugbr_lose_1.wav")
        Abspiellos = 1
    End If
    If Hlsoll < 495 Then Abspiellos = 0
End If
End Sub

Private Sub Motorstromsig_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Motorstromsig.Click
End Sub

Public Shared Function ToChar( _
    ByVal value As Byte _
) As Char

End Function

End Class
```



```

{
  servoHLB.attach(12);          // Den 3 Servos wird der Pin zugewiesen, an dem sie angeschlossen sind
  servoBZD.attach(10);
  servoHLL.attach(11);

  for(i=40;i<51;i++)
    pinMode(i, OUTPUT);      // Alle Pins für Ausgänge werden als Output gesetzt

  for(i=24;i<39;i++)          // Die PULL-UP Widerstände für die Inputs werden aktiviert, da gegen Masse
  geschaltet wird.
    {pinMode(i, INPUT);
     digitalWrite(i, HIGH);
    }

  Serial.begin(9600);          // Der serial port soll mit 9600 bps starten
  Wire.begin(1);              // Starten des I2C Bus
  delay(500);
  //initDisplay();

  // Pins für die Amperemeter und den Tacho werden als Ausgang festgelegt
  pinMode(5,OUTPUT);          // Tacho
  pinMode(6,OUTPUT);          // Bremsstrom
  pinMode(7,OUTPUT);          // Motorstrom
  pinMode(8,OUTPUT);          // Primärstrom
  analogWrite(5,255);         // Ausgänge der Amperemeter und den Tacho werden auf 5V gesetzt
  analogWrite(6,255);
  analogWrite(7,255);
  analogWrite(8,255);

  for (int thisReading = 0; thisReading < numReadings; thisReading++) // Startroutine für die
  Werteglättung
  readings[thisReading] = 0;
}

void initDisplay()
// Aktiviert den Dynamic mode der Digitaluhr und stellt dessen
Leuchtkraft ein. ( nicht benutzt )
{
  Wire.beginTransmission(saal064);
  Wire.write(B00000000);      // Instruction Byte
  Wire.write(B01000111);      // Control Byte (Dynamic Mode und LED-Strom)
  Wire.endTransmission();
}

void displayDigits()          //Anzeigen der Ziffern der Uhr
{
  for (int z=0; z<16; z++)
  {
    Wire.beginTransmission(saal064);
    Wire.write(1);            // Instruction Byte
    Wire.write(~Ziffern[h0]); // Stunde Ziffer 1
    Wire.write(~Ziffern[h1]+128); // Stunde Ziffer 2
    Wire.write(~Ziffern[m0]); // Minute Ziffer 1
    Wire.write(~Ziffern[m1]); // Minute Ziffer 2
    Wire.endTransmission();
  }
}

void loop()
{
  if (inputChar[0]=='w'){      // Wenn wir an erster Stelle ein "w" empfangen, möchte der PC
  wissen, ob wir wirklich
  Serial.println("I am a drivers desk"); // der Arduino für das Fahrpult sind. Das bestätigen wir mit
  "I am a drivers desk"
  if (inputChar[1]=='w'){      // Wenn an zweiter Stelle ebenfalls ein "w" kommt, sollen wir
  akustisch quittieren
    tone(5, 1000, 100);
    delay(200);
    tone(5, 1000, 100);
  }
  }
  if(inputChar[1] != -1)
  {
    if (inputChar[0] == 'P'){   // "P" für PZB, stellvertretend für alle Leuchtmelder
    LmPZB(inputChar[1]);        // Funktion siehe unten
    }

    if (inputChar[0] == 'V'){   // "V" für Geschwindigkeit
    n=1.85*inputChar[1];        // Da die halbe Geschwindigkeit übertragen wird, wird der
    übertragene Wert mit 2 multipliziert. Für den Tacho reicht eine Genauigkeit von 2 km/h aus.
    analogWrite(5,speed[n]);
    }

    if (inputChar[0] == 'H'){   // "H" für Hauptlleitung
    if (inputChar[1] < 75)
    servoHLL.write(163-2*inputChar[1]); // Wir weisen dem entsprechenden Servoobjekt seinen Wert zu
    }

    if (inputChar[0] == 'B'){   // "B" Bremszylinder
    servoBZD.write(185-0.815*inputChar[1]); // Wir weisen dem entsprechenden Servoobjekt seinen Wert zu
    }

    if (inputChar[0] == 'L'){   // "L" für Hauptluftbehälter
    servoHLB.write(265-2*inputChar[1]); // Wir weisen dem entsprechenden Servoobjekt seinen Wert zu
    }
  }
}

```

```

if (inputChar[0] == 'M'){ // "M" für Motorstrom
  analogWrite(7,255-(2*inputChar[1])); // Wert von 0 - 255 PWM an Pin 2 ausgeben
}

if (inputChar[0] == 'O'){ // "O" für Bremsstrom
  analogWrite(6,255-(2*inputChar[1])); // Wert von 0 - 255 PWM an Pin 4 ausgegeben
}

if (inputChar[0] == 'N'){ // "N" für Primärstrom
  analogWrite(8,255-2*inputChar[1]); // Wert von 0 - 255 PWM an Pin 7 ausgegeben
}

if (inputChar[0] == 'X') // "X" - Stunde
{
  h0 = inputChar[1]/10; // Aufspalten in 2 Ziffern
  h1 = inputChar[1]%10;
}

if (inputChar[0] == 'Y') // "Y" - Minute
{
  m0 = inputChar[1]/10; // Aufspalten in 2 Ziffern
  m1 = inputChar[1]%10;
}

//displayDigits();
}

if (inputChar[0] == 'S'){ // "S" für Schaltwerk
  //Wire.beginTransmission(3);
  //Wire.write(inputChar[1]);
  //Wire.endTransmission();
}

}

if (inputChar[0]=='i'){ // Wenn wir an erster Stelle ein "w" empfangen, möchte der PC
wissen, ob wir wirklich
Serial.println("readytosend");
if (inputChar[1]=='i')
{Input();
}

} //Es folgt die Werteglättung des Fahrschalters:

// subtract the last reading:
total= total - readings[index];
// read from the sensor:
readings[index] = analogRead(14)+5;
// add the reading to the total:
total= total + readings[index];
// advance to the next position in the array:
index = index + 1;

// if we're at the end of the array...
if (index >= numReadings)
// ...wrap around to the beginning:
index = 0;

// calculate the average:
average = total / numReadings;
}

void serialEvent() { // Es sind Daten vom PC zugespielt worden
for (byte i=0; i<2;i++){
inputChar[i]=Serial.read(); // inputChar[0] Ziel und inputChar[1] Wert zuweisen
delay(5); // Da es seine Zeit braucht, bis der PC alles sendet, wird
zwischen den Bytes etwas abgewartet
}
// Serial.flush(); // Falls aus irgend einem Grund nun noch Daten im Lesepuffer
sind, wird dieser gelöscht
}

void Input() { // Übertragen der Bytes zum PC
Serial.write(255); // Überprüfungsbyte
Serial.write(INa); // Digital Input I
Serial.write(INb); // Digital Input II
Serial.write(INc); // Analog Input I (Führerbremventil)
Serial.write(INd); // Analog Input II (Fahrschalter)

// Nach der Übertragung werden die Digital Input Bytes =0 gesetzt und
neu "befüllt":
INa=0;
if (digitalRead(24)==LOW)INa|=1; //Lüfter
if (digitalRead(25)==LOW)INa|=2; //PZB Befehl
if (digitalRead(26)==HIGH)INa|=4; //PZB Frei
if (digitalRead(27)==HIGH)INa|=8; //PZB Wachsam
if (digitalRead(28)==LOW)INa|=16; //HSein
if (digitalRead(33)==LOW)INa|=32; //SIFA
if (digitalRead(30)==HIGH)INa|=64; //Horn
if (digitalRead(29)==LOW)INa|=128; //Hauptschalter

INb=0;
if (digitalRead(34)==LOW)INb|=1; //Lokbremse Lösen
if (digitalRead(35)==LOW)INb|=2; //Lokbremse Anlegen
if (digitalRead(31)==LOW)INb|=4; //Sand
if (digitalRead(32)==LOW)INb|=8; //Stromabnehmer
if (digitalRead(33)==LOW)INb|=16; //unbelegt
if (digitalRead(37)==LOW)INb|=32; //unbelegt
if (digitalRead(38)==LOW)INb|=64; //unbelegt
if (digitalRead(39)==LOW)INb|=128; //unbelegt

```

```
HLread = round(150 + 0.15*(1024-analogRead(15))); //Sollbremsdruck vom FbV wird berechnet
HLsoll = HLread;
if(HLread>228)HLsoll=235; //Anpassung für Lösestellung
if(HLread>245)HLsoll=255; //Anpassung für Füllstellung
if(HLread<30)HLsoll=0; //Anpassung für Schnellbremsstellung

INc=HLsoll; //Sollbremsdruck wird als drittes Byte übertragen

if(FS>80)FS=0;
//FS=(1024-average)/4; //Stellung des Fahrschalters wird berechnet
FS=analogRead(14);

INd=fahr[average];
//INd=analogRead(14);

if(fahr[average]>=14)
schalt=fahr[average]-14;

Wire.beginTransmission(3);
Wire.write(schalt);
Wire.endTransmission();
}

void LmPZB(byte LM){ // Leuchtmelder vom PC werden empfangen
if ((LM&1) > 0){digitalWrite(41, HIGH);} //P70
if ((LM&1) == 0){digitalWrite(41, LOW);}
if ((LM&2) > 0){digitalWrite(42, HIGH);} //D95
if ((LM&2) == 0){digitalWrite(42, LOW);}
if ((LM&4) > 0){digitalWrite(43, HIGH);} //1000
if ((LM&4) == 0){digitalWrite(43, LOW);}
if ((LM&8) > 0){digitalWrite(45, HIGH);} //LM Hauptschalter aus
if ((LM&8) == 0){digitalWrite(45, LOW);}
if (n<1){digitalWrite(46, HIGH);} //LM Lokomotive steht
else{digitalWrite(46,LOW);}
if ((LM&16) > 0){digitalWrite(47, HIGH);} //LM Türen I
if ((LM&16) == 0){digitalWrite(47, LOW);}
if ((LM&32) > 0){digitalWrite(48, HIGH);} //LM Elektrische Bremse
if ((LM&32) == 0){digitalWrite(48, LOW);}
if ((LM&64) > 0){digitalWrite(50, HIGH);} //Fahrdratspannung
if ((LM&64) == 0){digitalWrite(50, LOW);}

}
```


Anhang A3: C Programm des Arduino UNO

```

#include <Wire.h>

float verz = 23;           // Verzögerung
int x=0;
long FSsoll=0;           // Sollwert der Stufenanzeige (Schritte)
long FSist = 0;         // Istwert der Stufenanzeige (Schritte)
long previousMillis = 0; // will store last time LED was updated

void setup() {
  pinMode(12,OUTPUT);    // Richtungspin wird als Ausgang gesetzt
  pinMode(13,OUTPUT);    // Schrittpin wird als Ausgang gesetzt
  FSist=0;               // Istwert wird auf 0 gesetzt. (Stufenanzeige ist im
  Ausgangszustand auf 0)
  FSsoll=0;             // Sollwert wird eingangs ebenfalls auf 0 gesetzt
  delay(50);

  Wire.begin(3);        // Join auf I2C Bus mit Adresse 03
  Wire.onReceive(receiveEvent); // Einbinden des recieveEvent() beim Empfangen von Bytes am
  Bus
  Serial.begin(9600);   // Starten des Bus mit einer Baudrate von 9600
}

void receiveEvent(int howMany) // Dieses Unterprogramm wird durchlaufen, wenn etwas empfangen
  wird
{
  FSsoll = Wire.read()*200; // Sollwert wird vom Bus empfangen. Da der Motor 200 Schritte
  pro Umdrehung hat,      // wird dieser mit 200 multipliziert.
}

void Runup() // Hochlaufen bis 30 Schritte vor dem Sollwert mit fahrgeschwindigkeit
{
  do{
    Stepup();
    delay(2.5);
  }
  while(FSist+30<FSsoll);
  Stopup();
}

void Rundown() // Herunterlaufen bis 30 Schritte vor dem Sollwert mit Fahrgeschwindigkeit
{
  do{
    Stepdown();
    delay(2.5);
  }
  while(FSist-30>FSsoll);
  Stopdown();
}

void Startup() // Aufwärts beschleunigen - 22 Schritte lang wird die Geschwindigkeit
  verlaufend erhöht, damit aufgrund // der Trägheit der Achse kein Schrittfehler entsteht.
{
  for(verz=25; verz>2.6; verz=verz/1.1){ // die Totzeit zwischen den Schritten startet mit
    // 25ms und wird pro Durchlauf
    // durch 1.11 dividiert, bis ein Wert von < 2.6
    // erreicht ist, also die Fahr-
    // geschwindigkeit.
    Stepup(); // Schritt aufwärts
    delay(verz); // Verzögerung zwischen den Schritten
  }
}

void Startdown() // Gleiches Prinzip wie Startup, nur in die andere Richtung
{
  for(verz=25; verz>2.6; verz=verz/1.1){
    Stepdown();
    delay(verz);
  }
}

void Stopup() // Damit beim stehenbleiben auch kein Schrittfehler stattfindet und die
  mechanischen Teile geschont werden, // wird die letzten 30 Schritte lang verlaufend abgebremst.
{
  verz=2.5; // Die Verzögerung startet also bei Fahrgeschwindigkeit (2.5ms Verzögerung) und
  // wird pro Schritt mit 1.09
  // multipliziert, somit wird auch wieder verlaufend angehalten. Sobald FSist =
  // FSsoll, wird das Unterpgorgramm
  // verlassen.
  do{
    if(verz<25)
      verz=verz*1.09;
    Stepup();
    delay(verz);
  }while(FSist<FSsoll);
}

void Stopdown() // Gleich wie Stopup, nur in die andere Richtung
{
  verz=2.5;
  do{
    if(verz<25)
      verz=verz*1.09;
    Stepdown();
    delay(verz);
  }
}

```

```
}while(FSist>FSsoll);
}
void Stepup()      //Schritt aufwärts
{
    digitalWrite(12,HIGH); //So wird die Richtung auf RL eingestellt (PIN 12 HIGH),...
    digitalWrite(13,LOW); //...und am PIN 13 ein IMPULS gesendet (Auf LOW...
    digitalWrite(13,HIGH); //...und wieder auf HIGH).
    FSist++;
}
void Stepdown()   //Schritt abwärts
{
    digitalWrite(12,LOW);
    digitalWrite(13,LOW);
    digitalWrite(13,HIGH);
    FSist--;
}
void loop()
{
    if(FSist<FSsoll) // Ist der Istwert kleiner als der Sollwert,...
    {
        Startup(); // ... so werden die entsprechenden Unterprogramme für die Fahrt
        Runup();   // zum Sollwert aufgerufen.
    }
    if(FSist>FSsoll)
    {
        Startdown();
        Rundown();
    }
}
```

Anhang B. Datenblätter

Anhang B1 Mosfet:

intersil **RFD14N05L, RFD14N05LSM, RFP14N05L**

Data Sheet **April 1999** **File Number 2246.3**

14A, 50V, 0.100 Ohm, Logic Level, N-Channel Power MOSFETs

These are N-channel power MOSFETs manufactured using the MegaFET process. This process, which uses feature sizes approaching those of LSI integrated circuits, gives optimum utilization of silicon, resulting in outstanding performance. They were designed for use in applications such as switching regulators, switching converters, motor drivers and relay drivers. This performance is accomplished through a special gate oxide design which provides full rated conductance at gate bias in the 3V-5V range, thereby facilitating true on-off power control directly from logic level (5V) integrated circuits.

Formerly developmental type TA09870.

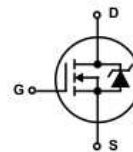
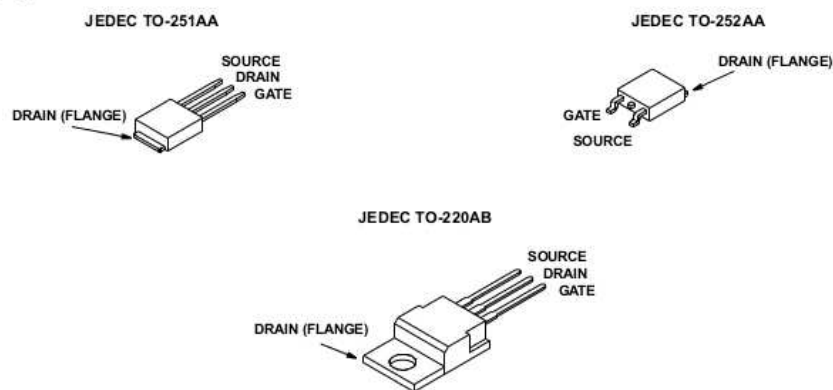
Ordering Information

PART NUMBER	PACKAGE	BRAND
RFD14N05L	TO-251AA	14N05L
RFD14N05LSM	TO-252AA	14N05L
RFP14N05L	TO-220AB	FP14N05L

NOTE: When ordering, use the entire part number. Add the suffix 9A to obtain the TO-252AA variant in the tape and reel, i.e., RFD14N05LSM9A.

Features

- 14A, 50V
- $r_{DS(ON)} = 0.100 \Omega$
- Temperature Compensating PSPICE™ Model
- Can be Driven Directly from CMOS, NMOS, and TTL Circuits
- Peak Current vs Pulse Width Curve
- UIS Rating Curve
- 175°C Operating Temperature
- Related Literature
 - TB334 "Guidelines for Soldering Surface Mount Components to PC Boards"

Symbol**Packaging**

6-135

CAUTION: These devices are sensitive to electrostatic discharge; follow proper ESD Handling Procedures.
PSPICE™ is a trademark of MicroSim Corporation.
<http://www.intersil.com> or 407-727-9207 | Copyright © Intersil Corporation 1999

RFD14N05L, RFD14N05LSM, RFP14N05L**Absolute Maximum Ratings** $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

	RFD14N05L, RFD14N05LSM, RFP14N05L	UNITS
Drain to Source Voltage (Note 1)	V_{DSS}	V
Drain to Gate Voltage ($R_{GS} = 20k\Omega$) (Note 1)	V_{DGR}	V
Gate to Source Voltage	V_{GS}	± 10 V
Continuous Drain Current	I_D	14 A
Pulsed Drain Current (Note 3)	I_{DM}	Refer to Peak Current Curve
Pulsed Avalanche Rating	E_{AS}	Refer to UIS Curve
Power Dissipation	P_D	48 W
Derate above 25°C		0.32 W/ $^\circ\text{C}$
Operating and Storage Temperature	T_J, T_{STG}	-55 to 175 $^\circ\text{C}$
Maximum Temperature for Soldering		
Leads at 0.063in (1.6mm) from Case for 10s	T_L	300 $^\circ\text{C}$
Package Body for 10s, See Techbrief 334	T_{pkg}	260 $^\circ\text{C}$

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

- $T_J = 25^\circ\text{C}$ to 150°C .

Electrical Specifications $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS	
Drain to Source Breakdown Voltage	BV_{DSS}	$I_D = 250\mu\text{A}, V_{GS} = 0\text{V}$, Figure 13	50	-	-	V	
Gate Threshold Voltage	$V_{GS(TH)}$	$V_{GS} = V_{DS}, I_D = 250\mu\text{A}$, Figure 12	1	-	2	V	
Zero Gate Voltage Drain Current	I_{DSS}	$V_{DS} = 40\text{V}, V_{GS} = 0\text{V}$	-	-	1	μA	
		$V_{DS} = 40\text{V}, V_{GS} = 0\text{V}, T_C = 150^\circ\text{C}$	-	-	50	μA	
Gate to Source Leakage Current	I_{GSS}	$V_{GS} = \pm 10\text{V}$	-	-	± 100	nA	
Drain to Source On Resistance (Note 2)	$r_{DS(ON)}$	$I_D = 14\text{A}, V_{GS} = 5\text{V}$, Figures 9, 11	-	-	0.100	Ω	
Turn-On Time	$t_{i(ON)}$	$V_{DD} = 25\text{V}, I_D = 7\text{A}, R_L = 3.57\Omega, V_{GS} = 5\text{V}, R_{GS} = 0.6\Omega$	-	-	60	ns	
Turn-On Delay Time	$t_{d(ON)}$		-	13	-	ns	
Rise Time	t_r		-	24	-	ns	
Turn-Off Delay Time	$t_{d(OFF)}$		-	42	-	ns	
Fall Time	t_f		-	16	-	ns	
Turn-Off Time	$t_{i(OFF)}$		-	-	100	ns	
Total Gate Charge	$Q_g(TOT)$		$V_{GS} = 0\text{V}$ to 10V	$V_{DD} = 40\text{V}, I_D = 14\text{A}, R_L = 2.86\Omega$ Figures 20, 21	-	-	40
Gate Charge at 5V	$Q_g(5)$	$V_{GS} = 0\text{V}$ to 5V	-		-	25	nC
Threshold Gate Charge	$Q_g(TH)$	$V_{GS} = 0\text{V}$ to 1V	-		-	1.5	nC
Input Capacitance	C_{ISS}	$V_{DS} = 25\text{V}, V_{GS} = 0\text{V}, f = 1\text{MHz}$ Figure 14	-	670	-	pF	
Output Capacitance	C_{OSS}		-	185	-	pF	
Reverse Transfer Capacitance	C_{RSS}		-	50	-	pF	
Thermal Resistance Junction to Case	$R_{\theta JC}$		-	-	3.125	$^\circ\text{C}/\text{W}$	
Thermal Resistance Junction to Ambient	$R_{\theta JA}$	TO-251 and TO-252	-	-	100	$^\circ\text{C}/\text{W}$	
	$R_{\theta JA}$	TO-220	-	-	80	$^\circ\text{C}/\text{W}$	

Source to Drain Diode Specifications

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Source to Drain Diode Voltage (Note 2)	V_{SD}	$I_{SD} = 14\text{A}$	-	-	1.5	V
Diode Reverse Recovery Time	t_{rr}	$I_{SD} = 14\text{A}, dI_{SD}/dt = 100\text{A}/\mu\text{s}$	-	-	125	ns

NOTES:

- Pulse Test: Pulse Width " 300ms, Duty Cycle " 2%.
- Repetitive Rating: Pulse Width limited by max junction temperature. See Transient Thermal Impedance Curve (Figure 3) and Peak Current Capability Curve (Figure 5).

RFD14N05L, RFD14N05LSM, RFP14N05L

Typical Performance Curves Unless Otherwise Specified

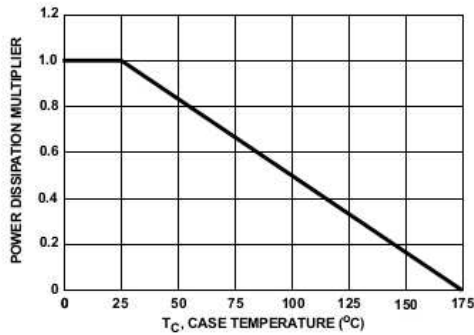


FIGURE 1. NORMALIZED POWER DISSIPATION vs CASE TEMPERATURE

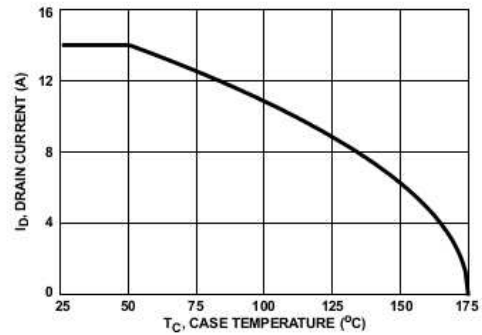


FIGURE 2. MAXIMUM CONTINUOUS DRAIN CURRENT vs TEMPERATURE

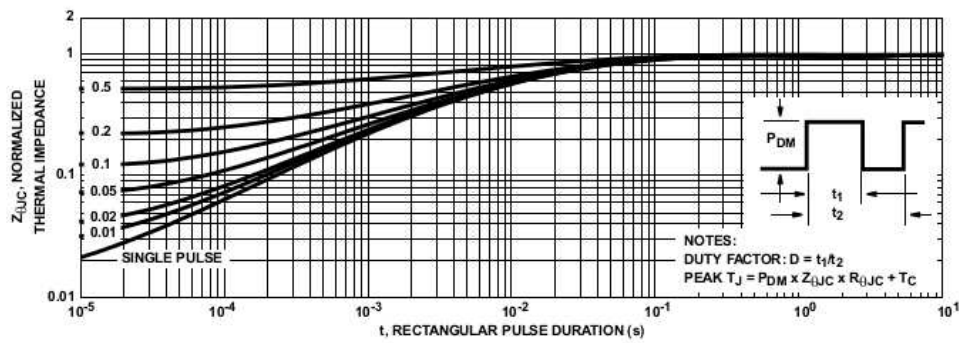


FIGURE 3. NORMALIZED MAXIMUM TRANSIENT THERMAL IMPEDANCE

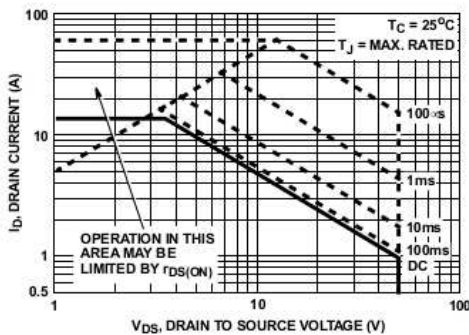


FIGURE 4. FORWARD BIAS SAFE OPERATING AREA

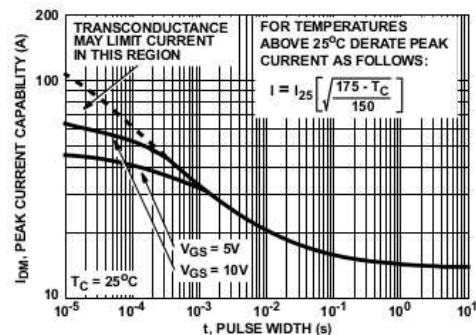
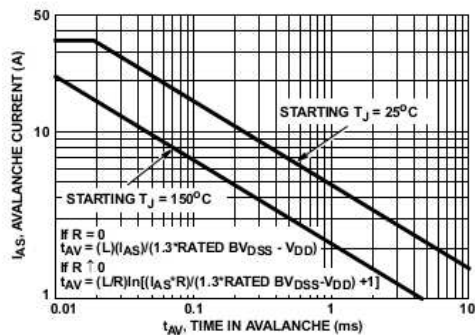


FIGURE 5. PEAK CURRENT CAPABILITY

RFD14N05L, RFD14N05LSM, RFP14N05L

Typical Performance Curves Unless Otherwise Specified (Continued)



NOTE: Refer to Intersil Application Notes AN9321 and AN9322.

FIGURE 6. UNCLAMPED INDUCTIVE SWITCHING

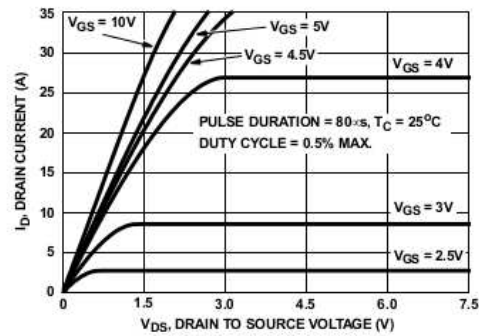


FIGURE 7. SATURATION CHARACTERISTICS

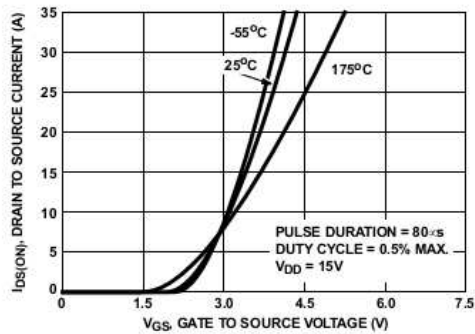


FIGURE 8. TRANSFER CHARACTERISTICS

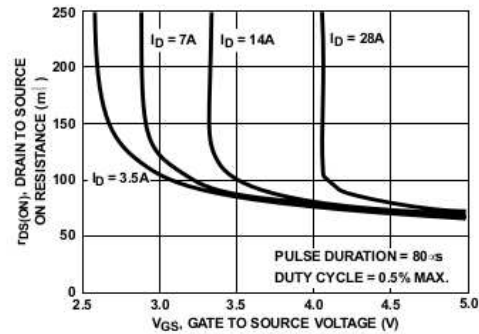


FIGURE 9. DRAIN TO SOURCE ON RESISTANCE vs GATE VOLTAGE AND DRAIN CURRENT

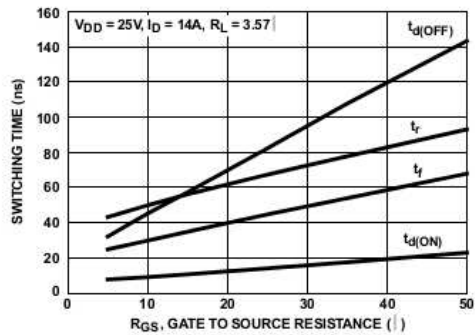


FIGURE 10. SWITCHING TIME vs GATE RESISTANCE

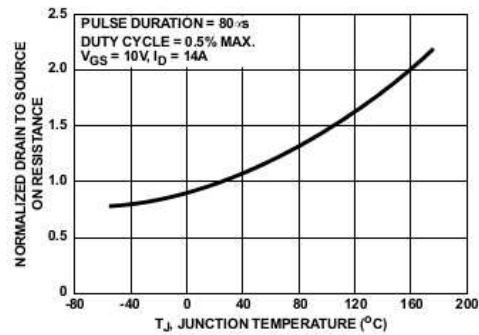


FIGURE 11. NORMALIZED DRAIN TO SOURCE ON RESISTANCE vs JUNCTION TEMPERATURE

RFD14N05L, RFD14N05LSM, RFP14N05L

Typical Performance Curves Unless Otherwise Specified (Continued)

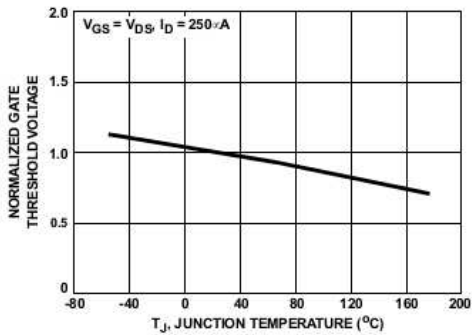


FIGURE 12. NORMALIZED GATE THRESHOLD VOLTAGE vs JUNCTION TEMPERATURE

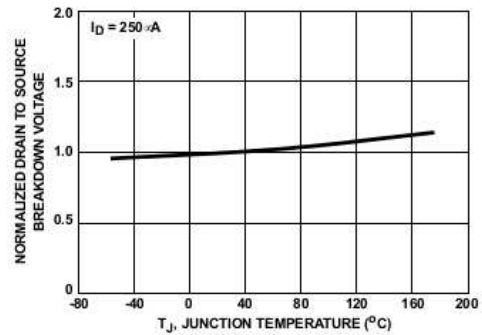


FIGURE 13. NORMALIZED DRAIN TO SOURCE BREAKDOWN VOLTAGE vs JUNCTION TEMPERATURE

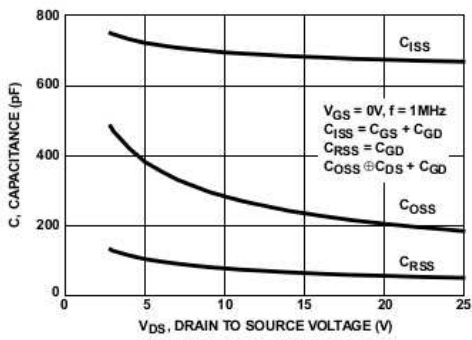
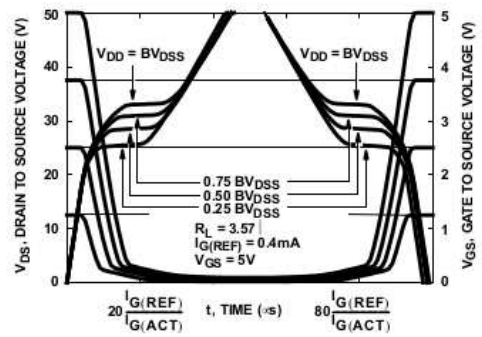


FIGURE 14. CAPACITANCE vs DRAIN TO SOURCE VOLTAGE



NOTE: Refer to Intersil Application Notes AN7254 and AN7260, FIGURE 15. TRANSCONDUCTANCE vs DRAIN CURRENT

RFD14N05L, RFD14N05LSM, RFP14N05L

Test Circuits and Waveforms

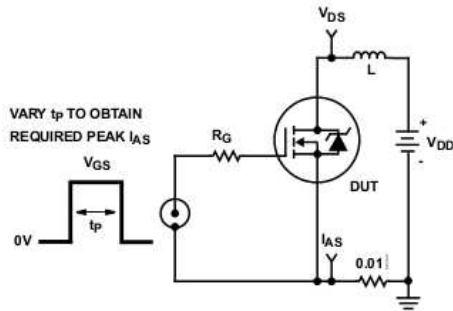


FIGURE 16. UNCLAMPED ENERGY TEST CIRCUIT

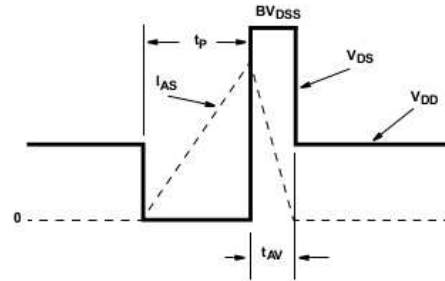


FIGURE 17. UNCLAMPED ENERGY WAVEFORMS

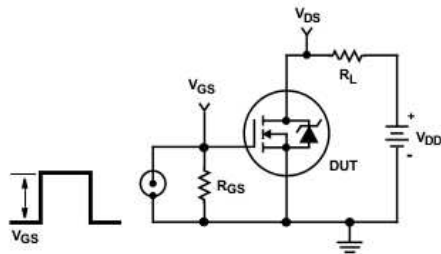


FIGURE 18. SWITCHING TIME TEST CIRCUIT

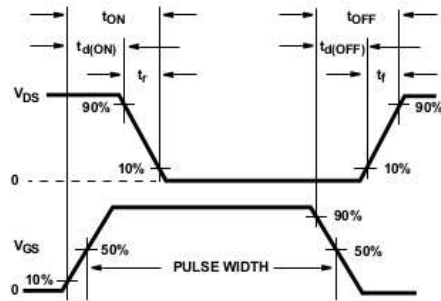


FIGURE 19. RESISTIVE SWITCHING WAVEFORMS

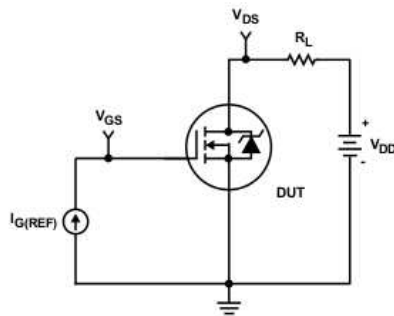


FIGURE 20. GATE CHARGE TEST CIRCUIT

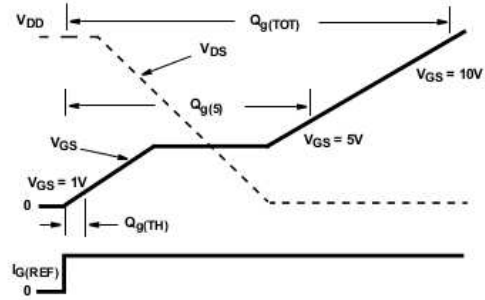


FIGURE 21. GATE CHARGE WAVEFORMS

RFD14N05L, RFD14N05LSM, RFP14N05L

PSPICE Electrical Model

.SUBCKT RFP14N05L 2 1 3; rev 9/15/94

CA 12 8 1.464e-9
CB 15 14 1.64e-9
CIN 6 8 6.17e-10DBODY 7 5 DBDMOD
DBREAK 5 11 DBKMOD
DPLCAP 10 5 DPLCAPMODEBREAK 11 7 17 18 65.35
EDS 14 8 5 8 1
EGS 13 8 6 8 1
ESG 6 10 6 8 1
EVTO 20 6 18 8 1

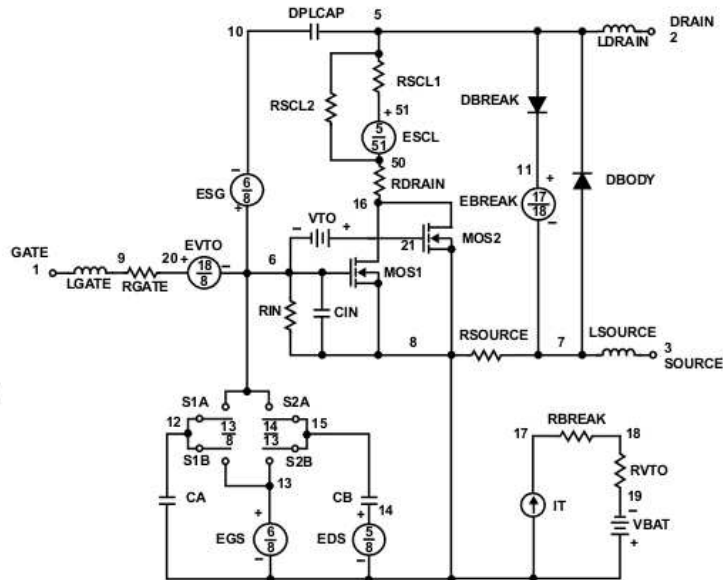
IT 8 17 1

LDRAIN 2 5 1e-9
LGATE 1 9 5.68e-9
LSOURCE 3 7 5.35e-9MOS1 16 6 8 8 MOSMOD M=0.99
MOS2 16 21 8 8 MOSMOD M=0.01RBREAK 17 18 RBKMOD 1
RDRAIN 50 16 RDSMOD 33.1e-3
RGATE 9 20 5.85
RIN 6 8 1e9
RSCL1 5 51 RSCLMOD 1e-6
RSCL2 5 50 1e3
RSOURCE 8 7 RDSMOD 14.3e-3
RVTO 18 19 RVTOMOD 1S1A 6 12 13 8 S1AMOD
S1B 13 12 13 8 S1BMOD
S2A 6 15 14 13 S2AMOD
S2B 13 15 14 13 S2BMODVBAT 8 19 DC 1
VTO 21 6 0.485

ESCL 51 50 VALUE = ((V(5,51)/ABS(V(5,51)))^(PWR(V(5,51))*1e6/46.7))

.MODEL DBDMOD D (IS = 2.23e-13 RS = 1.15e-2 TRS1 = 1.64e-3 TRS2 = 7.89e-6 CJO = 6.83e-10 TT = 3.68e-8)
.MODEL DBKMOD D (RS = 3.8e-1 TRS1 = 1.89e-3 TRS2 = 1.13e-5)
.MODEL DPLCAPMOD D (CJO = 25.7e-11 IS = 1e-30 N = 10)
.MODEL MOSMOD NMOS (VTO = 1.935 KP = 18.89 IS = 1e-30 N = 10 TOX = 1 L = 1u W = 1u)
.MODEL RBKMOD RES (TC1 = 7.18e-4 TC2 = 1.53e-6)
.MODEL RDSMOD RES (TC1 = 4.45e-3 TC2 = 2.9e-5)
.MODEL RSCLMOD RES (TC1 = 2.8e-3 TC2 = 6.0e-6)
.MODEL RVTOMOD RES (TC1 = -1.7e-3 TC2 = -2.0e-6)
.MODEL S1AMOD VSWITCH (RON = 1e-5 ROFF = 0.1 VON = -3.55 VOFF = -1.55)
.MODEL S1BMOD VSWITCH (RON = 1e-5 ROFF = 0.1 VON = -1.55 VOFF = -3.55)
.MODEL S2AMOD VSWITCH (RON = 1e-5 ROFF = 0.1 VON = -2.55 VOFF = 2.45)
.MODEL S2BMOD VSWITCH (RON = 1e-5 ROFF = 0.1 VON = 2.45 VOFF = -2.55)

.ENDS

NOTE: For further discussion of the PSPICE model, consult **A New PSPICE Sub-circuit for the Power MOSFET Featuring Global Temperature Options**; authored by William J. Hepp and C. Frank Wheatley.

RFD14N05L, RFD14N05LSM, RFP14N05L

All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

Intersil semiconductor products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see web site <http://www.intersil.com>

Sales Office Headquarters**NORTH AMERICA**

Intersil Corporation
P. O. Box 883, Mail Stop 53-204
Melbourne, FL 32902
TEL: (407) 724-7000
FAX: (407) 724-7240

EUROPE

Intersil SA
Mercure Center
100, Rue de la Fusee
1130 Brussels, Belgium
TEL: (32) 2.724.2111
FAX: (32) 2.724.22.05

ASIA

Intersil (Taiwan) Ltd.
7F-6, No. 101 Fu Hsing North Road
Taipei, Taiwan
Republic of China
TEL: (886) 2 2716 9310
FAX: (886) 2 2715 3029

6-142

The Intersil logo consists of the word "intersil" in a lowercase, bold, sans-serif font. The letter "i" is lowercase, while "ntersil" are uppercase. The logo is positioned to the right of the page number "6-142".

Anhang B2 Schrittmotor:**1.8° 57mm Hybrid Stepper Motor-NEMA23****General Specification:**

Step Angle ----- 1.8°±5%
 Temperature Rise ----- 80°Cmax
 Ambient Temperature ----- -20°C~+50°C
 Insulation Resistance----- 100 MΩ Min. ,500VDC
 Dielectric Strength: ----- 500VAC for 1minute
 Shaft Radial Play ----- 0.02Max. (450g-load)
 Shaft Axial Play ----- 0.08Max. (450g-load)
 Max. radial force----- 75N (20mm from the flange)
 Max. axial force----- 15N

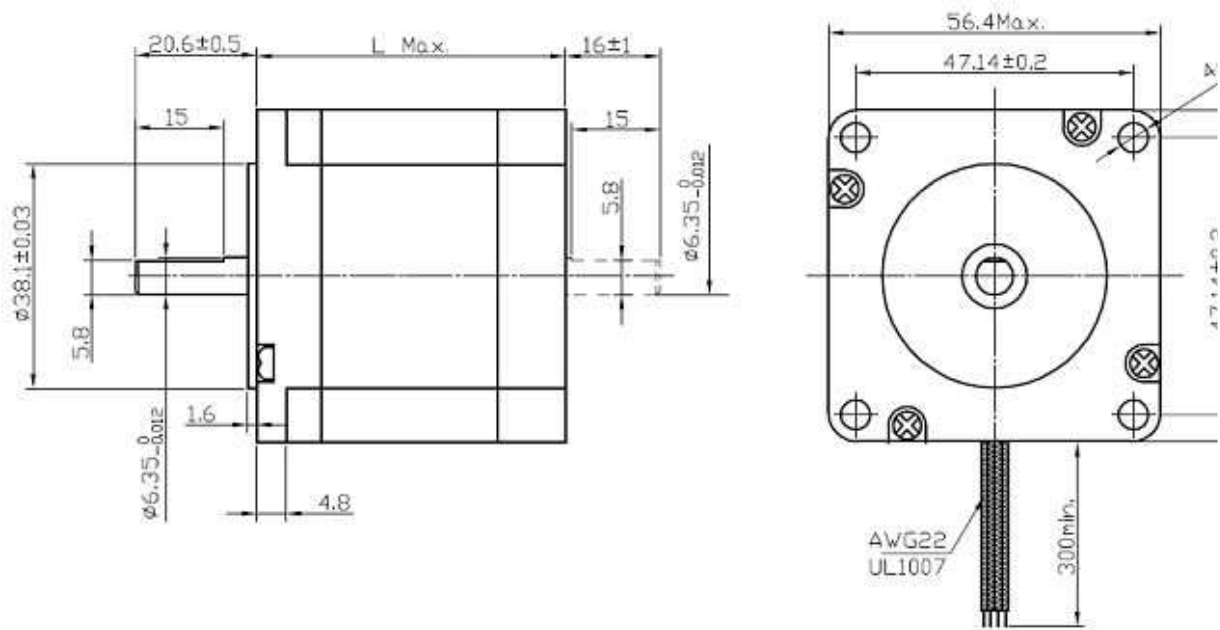
Electrical Specification:

Model No.	Step Angle (°)	Motor Length (L)mm	Current /Phase A	Resistance /Phase Ω	Inductance /Phase mH	Holding Torque N.m	# of Leads No.	Detent Torque g.cm	Rotor Inertia g.cm	Mass Kg
JK57H41-1006	1.8	41	1	5.2	5.5	0.4	6	250	150	0.47
JK57H41-0604	1.8	41	0.6	12	24	0.55	4	250	150	0.47
JK57H41-2004	1.8	41	2	1.2	2.5	0.55	4	250	150	0.47
JK57H45-0604	1.8	45	0.6	12	26	0.8	4	280	190	0.52
JK57H45-2504	1.8	45	2.5	1.0	2.2	0.8	4	280	190	0.52
JK57H51-0604	1.8	51	0.6	13	28	1.1	4	300	230	0.59
JK57H51-2504	1.8	51	2.5	1.2	3.2	1.1	4	300	230	0.59
JK57H51-0806	1.8	51	0.8	6.8	9.2	0.62	6	300	230	0.59
JK57H56-	1.8	56	1.5	3.2	5.5	0.9	6	350	280	0.68

1506										
JK57H56-2504	1.8	56	2.5	1.3	4.2	1.1	4	350	280	0.68
JK57H56-3004	1.8	56	3	0.8	2.4	1.1	4	350	280	0.68
JK57H56-4204	1.8	56	4.2	0.4	1.2	1.1	4	350	280	0.68
JK57H64-1004	1.8	64	1.0	7.5	20	1.5	4	500	380	0.85
JK57H64-2504	1.8	64	2.5	1.5	4.5	1.5	4	500	380	0.85
JK57H64-3004	1.8	64	3.0	0.8	2.3	1.5	4	500	380	0.85
JK57H64-4204	1.8	64	4.2	0.55	1.2	1.5	4	500	380	0.85
JK57H76-1506	1.8	76	1.5	4.5	9.2	1.4	6	600	440	1.1
JK57H76-2504	1.8	76	2.5	1.8	6.5	1.8	4	600	440	1.1
JK57H76-3004	1.8	76	3.0	1.0	3.5	1.8	4	600	440	1.1
JK57H76-4204	1.8	76	4.2	0.6	1.8	1.8	4	600	440	1.1
JK57H112-3004	1.8	112	3.0	1.6	6.8	2.8	4	1200	800	1.4
JK57H112-4204	1.8	112	4.2	0.9	3.8	2.8	4	1200	800	1.4

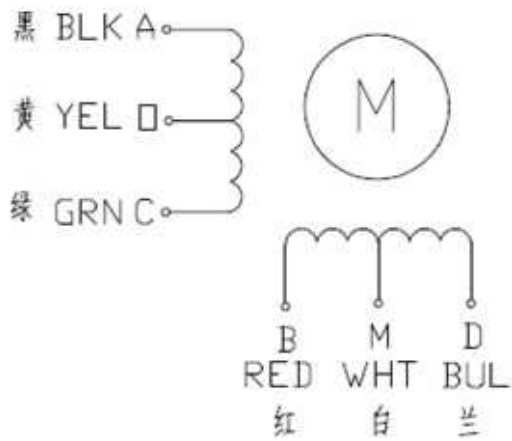
*We can produce it according your requests.

Dimensions:(Unit=mm)

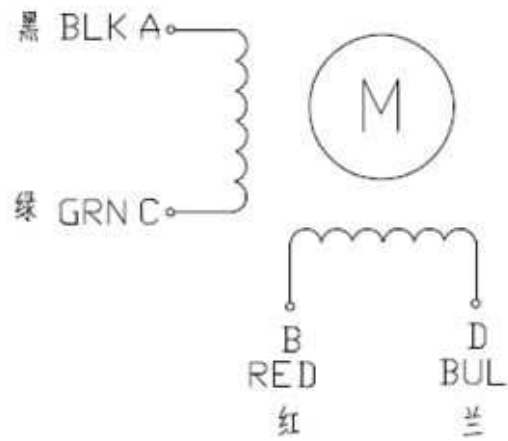


Wiring Diagram:

6 LEADS



4 LEADS



Anhang B3 Schrittmotoransteuerung:

Item specifics

Place of Origin:

China (Mainland)

Model Number:

M335-B

Product Description

Features:

High integration;high reliability

High speed optocoupler isolation interface

High resistance to the high-frequency interruption

Max. input voltage: DC 35V(peak)

Main functions:

Excitation choices: whole excitation/2 excitation/8 excitation/16 excitation

Output current: 7 choices

Auto over-heat protection

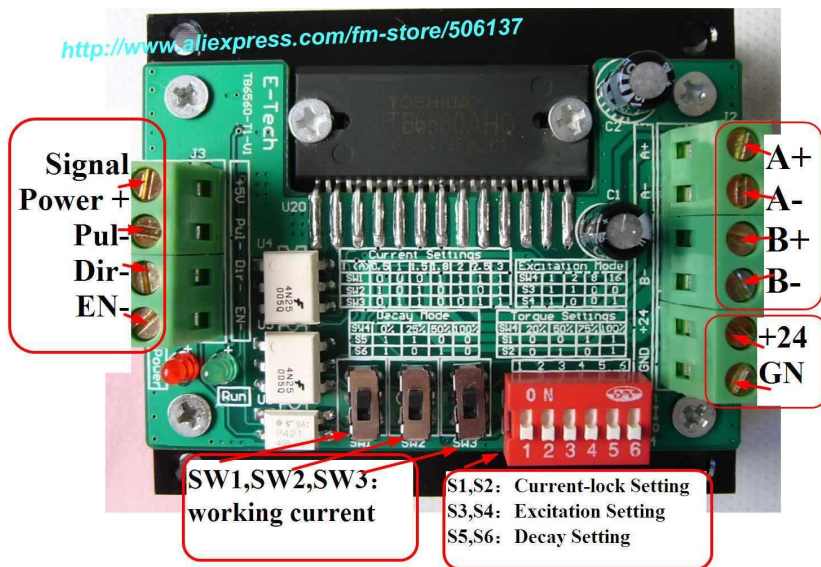
Auto semi-current locking, semi-current: 4 choices

Decay mode: 4 choices

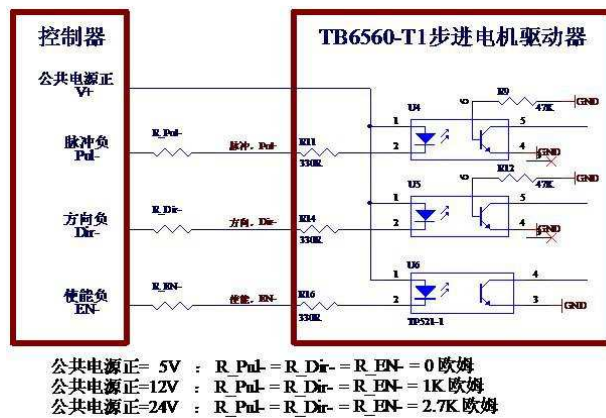
Working Factors:

Contents	Min.	Rated	Max.	Unit	
Temperature	-30	/	80	⁰ C	
DC Input Voltage(DC)	7	24	32	V	
Out Input Frequency	0	/	16000	Hz	
Inner Frequency	640	/	20000	Hz	
Output Current	0.6	/	3	A	
Interface Voltage	H	4.5	5	5.5	V
	L	0	0	0.5	V

Interface Definition:



Wiring Way:



- Controller interface voltage=5V, no resistance;
- Controller interface voltage=12V, series connect a 1K resistance;
- Controller interface voltage=24V, series connect a 2.7K resistance

Ps:

EN- inputs high level or being hanged, optocoupler U6 light off, step motor=working

EN- inputs low level, optocoupler U6 light on, step motor= not working

Working Current Setting

(Only can be adjustable when no current is passed through motor)

Current Setting							
I(A)	0.5	1	1.5	1.8	2	2.5	3
SW1	0	0	0	1	1	1	1
SW2	1	0	1	0	1	0	1
SW3	0	1	1	0	0	1	1
Resistance Ω	1	0.51	0.34	0.33	0.25	0.2	0.167

Excitation Setting:

Excitation Mode Setting				
SW4	1	2	8	16
S3/M2	1	1	0	0
S4/M1	1	0	0	1

Decay Setting:

Decay Mode Setting				
SW4	0%	25%	50%	100%
S5/DY2	1	1	0	0
S6/DY1	1	0	1	0

Decay Setting:

Decay Mode Setting				
SW4	0%	25%	50%	100%
S5/DY2	1	1	0	0
S6/DY1	1	0	1	0

Users can lower the noises and promote the stability by decay mode setting.

Semi-current Lock Torque Setting:

Semi-current Torque Setting				
SW4	20%	50%	75%	100%
S1/TQ2	0	0	1	1
S2/TQ1	0	1	0	1