

DSD 2010

(Drehscheiben-Dekoder 2010) Entwicklungsgrundlage / TecDoc

Dies ist keine Aufbau-Anleitung oder Bedienungs-Anleitung. Die hier beschriebenen Informationen sind für den Aufbau und den Betrieb des Moduls normalerweise nicht nötig.

Inhaltsverzeichnis

1 - Grundsätzliches	3
1.1 - Zweck dieses Doc.....	3
1.2 - System-Übersicht DSD2010.....	3
2 - Haupt-Platine (Grube)	5
2.1 - Aufgaben.....	5
2.2 - Funktionsblock „Eingang Digital-Spannung“.....	5
2.3 - Funktionsblock „Spannungsversorgung“.....	6
2.4 - Funktionsblock „Gleis umpolen“.....	6
2.5 - Funktionsblock „RS232“.....	7
2.6 - Funktionsblock „Rückmeldung“.....	7
2.6.1 - mechanische Schnittstelle für externe Rückmelde-Aufsätze.....	7
2.6.2 - Elektrische Schnittstelle für externe Rückmelde-Aufsätze.....	9
2.6.3 - Software-Schnittstelle für externe Rückmelde-Aufsätze.....	11
2.6.3.1 - Übertragung in Richtung Zentrale.....	11
2.6.3.2 - Übertragung in Richtung Drehscheibe.....	12
2.7 - Ablaufbeschreibung.....	13
2.7.1 - Soll / Ist-Position.....	13
3 - Bühnen-Platine (Bühne)	14
3.1 - Aufgaben.....	14
3.2 - Funktionsblock „Spannungsversorgung“.....	14
3.3 - Funktionsblock „Hall-Sensor“.....	14
3.4 - Funktionsblock „Motor-Ansteuerung“.....	15
3.5 - Funktionsblock „Erkennung Position“.....	16
3.6 - Funktionsblock „Rückmeldung“.....	18
3.7 - Funktionsblock „SUSI Schnittstelle“.....	19
3.8 - Ablaufbeschreibung.....	22
3.8.1 - State Machine.....	22
3.9 - Kommunikation mit der Haupt-Platine.....	23
3.9.1 - Bitübertragung (Physical Layer).....	23
3.9.2 - Gruben-Platine an Bühnen-Platine.....	23
3.9.2.1 - Info 1, Flags_01.....	27
3.9.2.2 - Info 2, Soll-Position.....	27
3.9.2.3 - Info 3, Flags_03.....	27
3.9.2.4 - Info 5, Write EEprom Bühne.....	28

3.9.2.5 - Info 6, Request EEprom Bühne.....	28
3.9.2.6 - Info 4, Direct Drive.....	28
3.9.3 - Bühnen-Platine an Haupt-Platine.....	29
3.9.3.1 - Info 1, Flags_02.....	29
3.9.3.2 - Info 2, Error-Code.....	30
3.9.3.3 - Info 3, Ist-Position.....	30
3.9.3.4 - Info 4, Sensor-Spannung.....	30
3.9.3.5 - Info 5, Strom-Wert.....	30
3.9.3.6 - Info 6, Value EEprom Bühne.....	31
4 - Anzeige-Platine / PC.....	32
4.1 - Aufgaben.....	32
4.2 - Kommunikation mit der Gruben-Platine.....	33
4.2.1 - Bitübertragung (Physical Layer).....	33
4.2.2 - Gruben-Platine an Anzeige-Platine / PC.....	33
4.2.2.1 - Info 1, Flags_01 / Flags_02.....	34
4.2.2.2 - Info 2, Errorcode_01 / _02.....	34
4.2.2.3 - Info 3, Positionen.....	34
4.2.2.4 - Info 4, Analog-Werte.....	34
4.2.2.5 - Info 5, Value EEprom Grube.....	35
4.2.2.6 - Info 6, Value EEprom Bühne.....	35
4.2.3 - Anzeige-Platine / PC an Gruben-Platine.....	36
4.2.3.1 - Info 1, Flags_01.....	37
4.2.3.2 - Info 2, Flags_03.....	37
4.2.3.3 - Info 3, Soll-Position und Ist-Position.....	38
4.2.3.4 - Info 4, Write EEprom Grube.....	38
4.2.3.5 - Info 5, Request EEprom Grube.....	38
4.2.3.6 - Info 6, Write EEprom Bühne.....	39
4.2.3.7 - Info 7, Request EEprom Bühne.....	39
4.2.3.8 - Info 8, Direct Drive.....	39
4.3 - Design: Multiplex Taster und Anzeigen.....	40
5 - Anhang.....	42
5.1 - Fehler-Codes.....	42
5.2 - Beschreibung der globalen Register.....	43
5.2.1 - R_SOLLPOSITION.....	43
5.2.2 - R_ISTPOSITION.....	43
5.2.3 - R_STAT_FLAG_01.....	44
5.2.4 - R_STAT_FLAG_02.....	44
5.2.5 - R_STAT_FLAG_03.....	45
5.2.6 - R_ERRORCODE_01.....	45
5.2.7 - R_ERRORCODE_02.....	45
5.2.8 - R_AD_SENSOR.....	46
5.2.9 - R_AD_SENSE.....	46
5.3 - EEprom Belegung - Grube.....	47
5.4 - EEprom Belegung - Bühne.....	51
5.5 - Parameter Lastregelung.....	54

1 Grundsätzliches

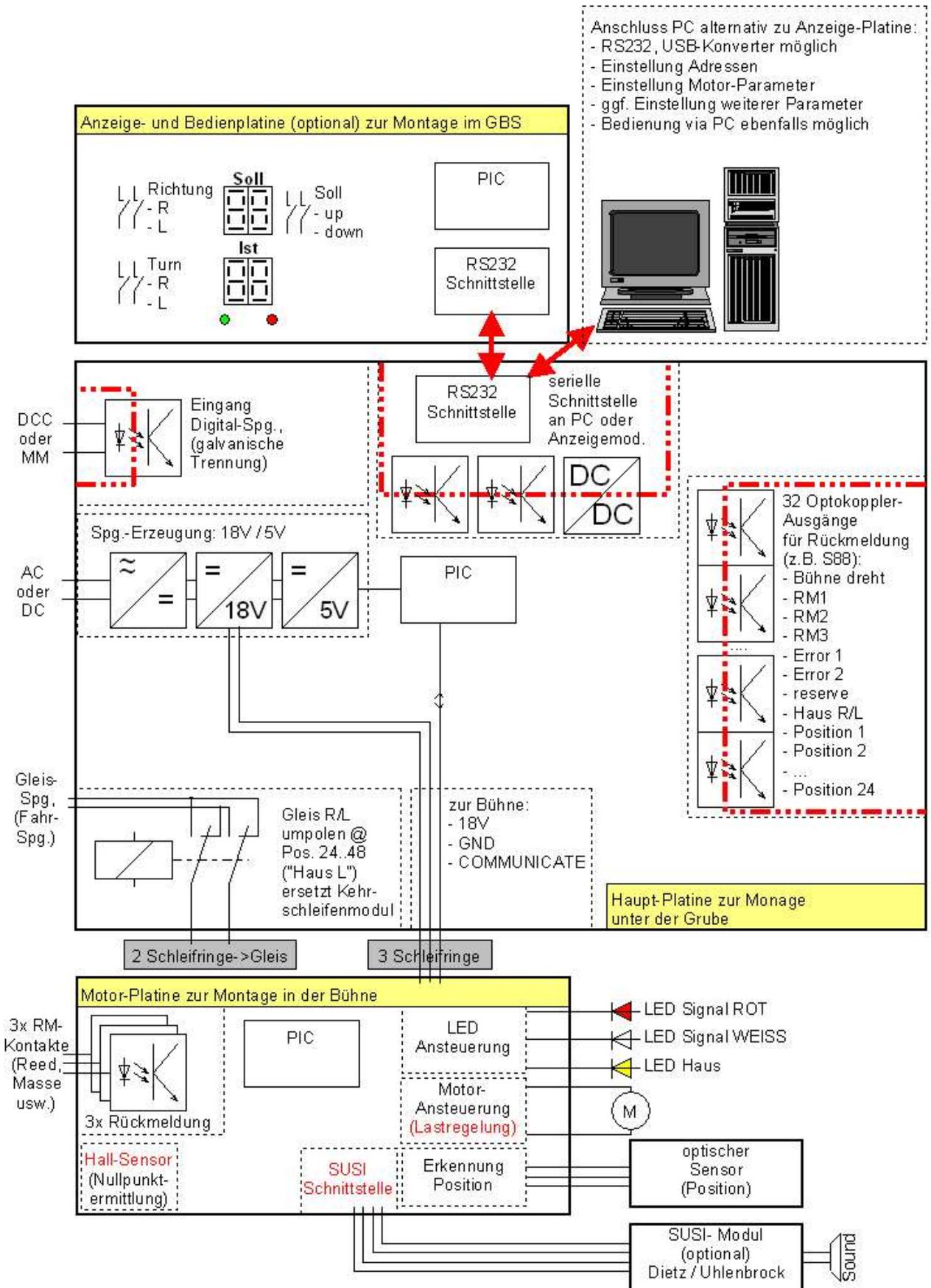
1.1 Zweck dieses Doc

Dieses Dokument dient der technischen Dokumentation für DSD2010. Für die Inbetriebnahme und den Betrieb des Systems ist dieses Dokument nicht relevant.

1.2 System-Übersicht DSD2010

Das System DSD2010 besteht aus 3 Modulen:

1. Gruben-Platine
2. Motor-Platine (= Bühnen-Platine)
3. Anzeige-Platine (diese Platine ist optional)



2 Haupt-Platine (Grube)

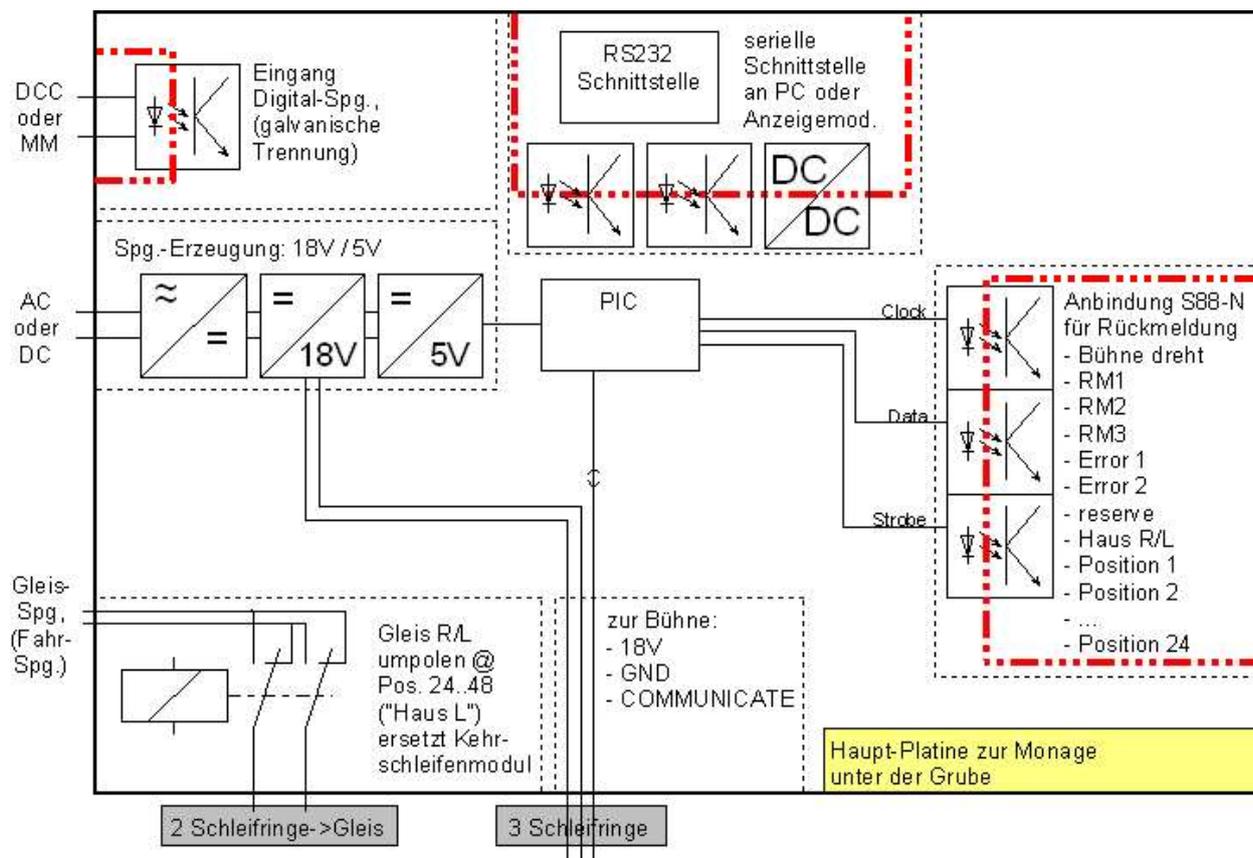


Abbildung 1: Haupt-Platine

2.1 Aufgaben

Die Hauptplatine fungiert als Zentral-Platine.

- Kommunikation mit der Anzeige-Platine
- Kommunikation mit der Bühnen-Platine
- Empfang der Befehle aus der Digital-Spannung (Gleisspannung)
- Ausgabe von Rückmeldungen.
- Bereitstellung der Versorgungs-Spannung für die Bühnen-Platine
- DIP-Schalter 1: OFF = 48 Positionen, ON = Positions-Anzahl aus EEPROM
- DIP-Schalter 2: OFF = DCC, ON = MM
- DIP-Schalter 3 OFF = Zielpositions-Berechnung bei Befehl via Digital einschalten (Märklin-kompatibel), ON = Berechnung aus (Fahre genau nach abgespeicherter Position)
- LED1 ROT: Anzeige: Kommunikation mit Bühne OK
- LED2 GELB: Anzeige: Kommunikation RS232 / Digital-Befehl (Aufleuchten bei Befehls-Empfang)
- LED3 GRÜN: Anzeige Bühne dreht (blinken)
- Berechnen der Soll-Position (wird dann an Bühnen-Platine übergeben)

2.2 Funktionsblock „Eingang Digital-Spannung“

Das digitale Signal (Gleissignal DCC oder MM) wird galvanisch getrennt auf den IR-Eingang des PICs gegeben.

Verwendete Optokoppler:

- THD: 6N137 (Reichelt)

2.3 Funktionsblock „Spannungsversorgung“

Die Haupt-Platine erhält DC oder AC-Spannung. Diese wird auf 18V und 5V herabgesetzt.

5V:

- PIC
- Optokoppler

18V:

- DCDC-Wandler für RS232-Wandler (Verbindung zum Anzeige-Modul / PC)
- Versorgung der Bühnen-Platine
- Umschalt-Relais

2.4 Funktionsblock „Gleis umpolen“

Ein Umpol-Relais wird von PIC in Abhängigkeit der Bühnen-Position angesteuert. Dadurch kann die Polarisierung der Gleise der Bühnenstellung angepasst werden (betrifft: 2-Leiter Bahn). Ein Kehrschleifen-Modul kann damit entfallen.

Es gilt für 48 Positionen Drehscheiben:

Position	Relais
1..24	abgefallen
25..48	angezogen

Es gilt für 24 Positionen Drehscheiben:

Position	Relais
1..12	abgefallen
13..24	angezogen

Berechnung: Vergleich der Ist-Position mit der durch den DIP1 definierten Positions-Anzahl.

2.5 Funktionsblock „RS232“

Die Haupt-Platine erhält einen RS232 Port (9-pol D-Sub Buchse). Über diesen kann die Anzeige-Platine angesteuert werden. Ebenfalls ist alternativ der Anschluss an den PC hierüber möglich.

Im Falle des PC-Anschlusses besteht die Gefahr einer Masse-Schleife

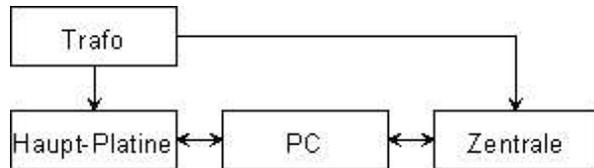


Abbildung 2: Masseschleife

→ daher erhält die RS232 Schnittstelle eine galvanische Trennung.

zum Thema USB:

das ist leider wirklich nicht so simpel. Es gibt hier fertige Bausteine, die aber ein sehr kleines Raster für die Lötpins haben und die ich daher verworfen habe. Des Weiteren kommt hinzu, dass man für USB dann wieder eine Menge neue Probleme hat. So muss man z.B. eine ID-Nummer beantragen, mit der ein PC das USB-Gerät identifizieren kann. Und das kosten wiederum...

Lange Rede, kurzer Sinn: es gibt fertige Konverter, die in diesem Fall völlig problemlos funktionieren. Die sehen fast aus wie ein normales Kabel USB--RS232, beinhalten aber die entsprechende Elektronik. Auch die PC-Treiber sind dann kein Problem. Und die Dinger kosten keine 10 Euro (bei Reichelt), da wäre der USB-Chip schon teurer.

Einziger Nachteil: Das USB-Gerät benötigt eine eigene Spannungs-Versorgung, die Spannung vom USB-Port kann nicht verwendet werden. Macht in diesem Fall aber nichts, die Platine muss ja sowieso an eine Spannung gelegt werden...

2.6 Funktionsblock „Rückmeldung“

Der PIC erzeugt ein serielles Übergabe-Signal (3 Signale), welches via Optokoppler an den Rückmelde-Part der Schaltung übertragen wird.

Auch hier kann es zu Masse-Schleifen kommen, da sie Masse der Haupt-Platine nicht mit der des S88-N Bussen identisch sein muss. Auch hier ist daher eine galvanische Trennung vorgesehen.

Es werden 3 Signale via Optokoppler (ILQ74) auf die Rückmelde Seite übertragen. Diese Signale sind so ausgelegt, dass diese dort durch ein 4094 wieder dekodiert werden könnten (also CLK / DATA / STROBE). Die Kommunikation erfolgt entsprechend langsam, sodass der ILQ74 verwendet werden kann (Schaltzeit einige µs). Zudem kann der Rückmelde-PIC dieses Signal nicht IR-gesteuert auswerten (INT ist hier zwingend dem S88-CLK zugeordnet)

Die Implementierung eines S88-N Interface erfolgt mittels eines PIC16F690, welches die Rückmelde-Daten direkt mit den 3 Daten-Signalen dekodiert. Die Versorgung des PIC / Optokoppler erfolgt dabei aus dem S88-N Bus. Ggf. könnte auch die RAILDATA-Info (also die „normale“ digitale Information des Gleis-Signals) des S88-N genutzt werden

Vorgesehen werden weiterhin Pfostenstecker, sodass eine zusätzliche Interface-Karte für alternative Rückmelde-Busse angeschlossen werden könnte (Realisierung extern!)

2.6.1 mechanische Schnittstelle für externe Rückmelde-Aufsätze

Es ist der Stecker X3 für den Anschluss an das externe Rückmelde-Modul vorgesehen. Zur mechanischen Stabilisierung kann zusätzlich auf X9 aufgesteckt werden.

Die Stecker liegen im RM2.54, Anordnung auf der Platine wie folgt:

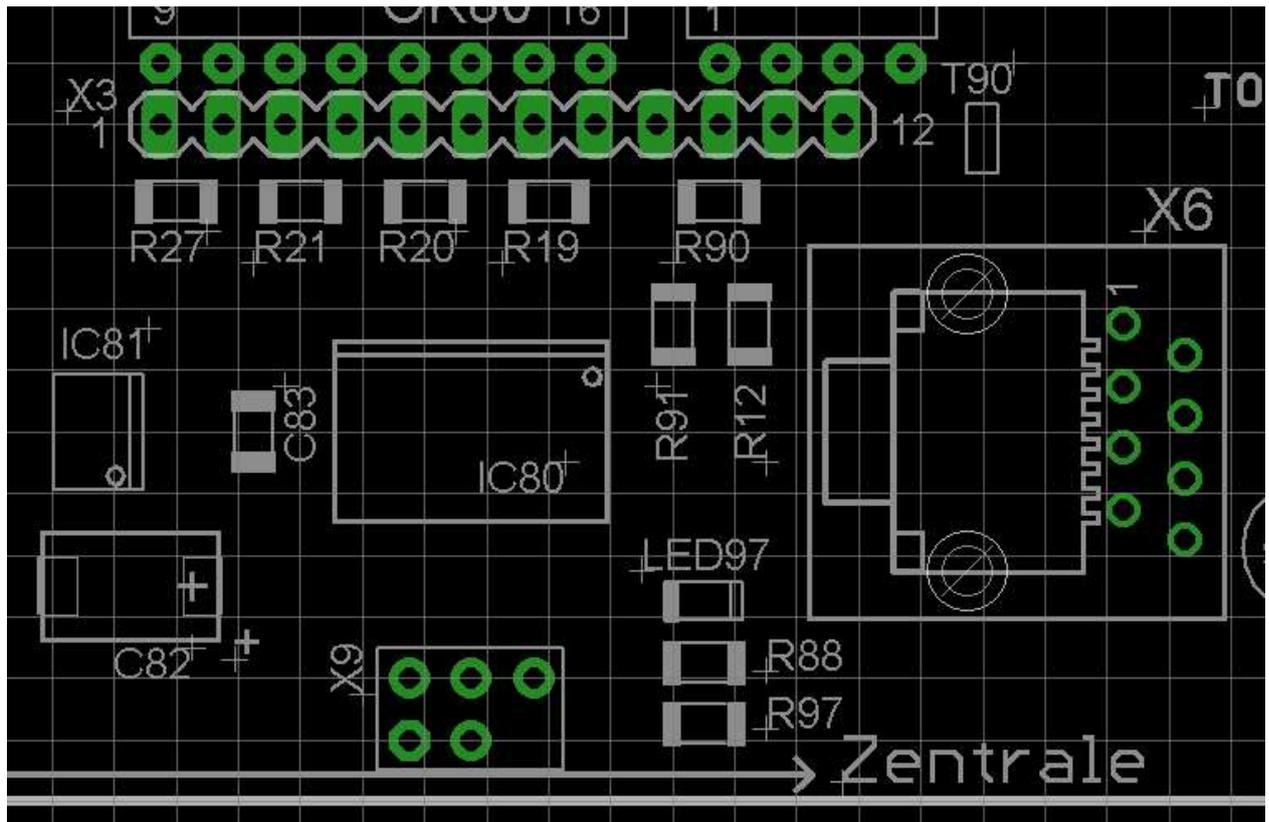


Abbildung 3: Rückmelde-Stecker mit Raster 2.54 (100mil)

2.6.2 Elektrische Schnittstelle für externe Rückmelde-Aufsätze

Die Übertragung erfolgt via Optokoppler OK80 (in Richtung Zentrale) und OK90 (in Richtung DS) X3 stellt die Schnittstelle zum Rückmelde-Modul dar, dort finden sich die Signale:

PIN #	Signal Name	Art	Funktion
1	GND		
2	RM_RESERVE_S88	Opto Ausgang, OC	reserve
3	RM_CLK_S88	Opto Ausgang, OC	Clock Signal
4	+5V		(1)
5	+5V		(1)
6	RM_DATA_S88	Opto Ausgang, OC	Data Signal
7	RM_STROBE_S88	Opto Ausgang, OC	Strobe Signal
8	GND		
9	GND		
10	GND		
11	OK90_A	Diode Anode	Übertragung der Gleis-Befehle
12	OK90_K	Diode Kathode	

(1) die 5V sind nur dann von Interesse, wenn die Pull-Up Widerstände R100ff für die OC-Ausgänge des OK80 verwendet werden. Ansonsten muss auf dem externen Modul entsprechende Maßnahmen für den OC Ausgang ergriffen werden.

Details zum Schaltplan:

Im Minimal-Fall sind nur OK80, OK90 und X3 als Interface nötig. Ggf. können die Widerstände R100..R103 als Pull-Up Widerstände verwendet werden.

IC81, C81, C82, C83, R90, T90, R91 sind dann **nicht** bestückt

R92..R96, C92 liegen auf der „anderen Seite“ des OK und sind demnach bestückt.

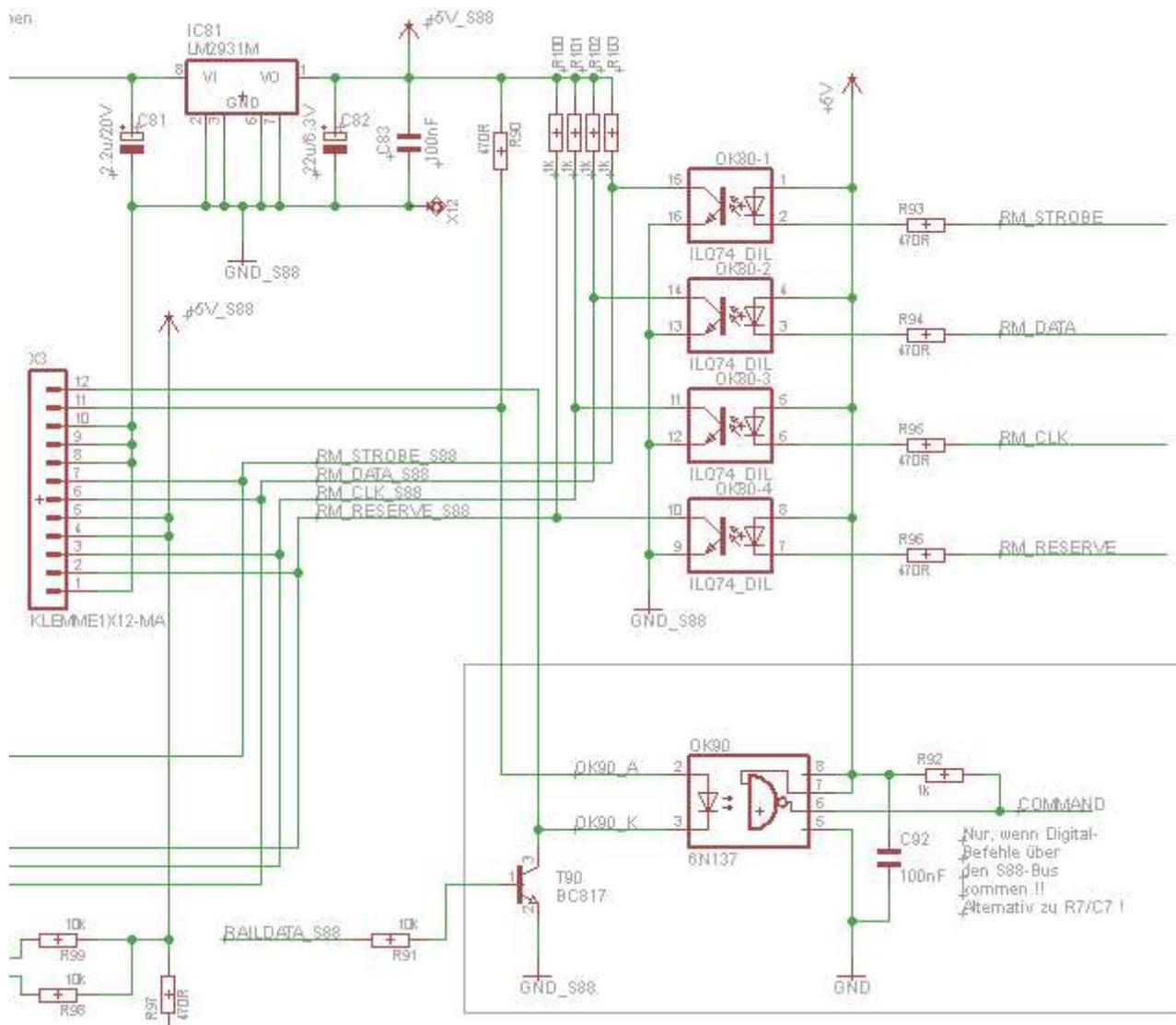


Abbildung 4: Schaltplanauszug Schnittstelle für den Anschluss von Rückmelde-Modulen

2.6.3 Software-Schnittstelle für externe Rückmelde-Aufsätze

2.6.3.1 Übertragung in Richtung Zentrale

Es werden 3 Signale Übertragen:

- Clock
- Data
- PS

Diese Signale sind schon von S88-Bus bekannt und werden auch entsprechend (mit entspanntem Timing) verwendet:

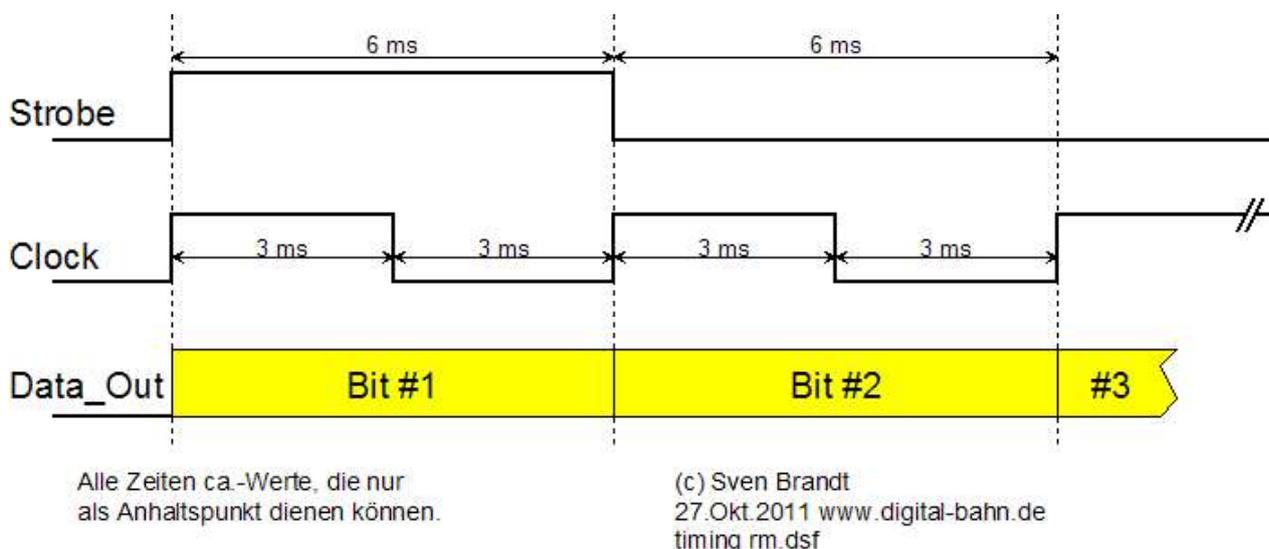


Abbildung 5: Timing an den Optokopplern für Rückmelde-Module

Wie un schwer zu erkennen ist das Timing völlig anspruchslos im ms Bereich ausgelegt. Dies ermöglicht zum einen die Verwendung eines 4er Optokoppler ohne jeden Anspruch auf schnelle Übertragung. Zum zweiten ermöglicht dies dem μC die Abarbeitung der Signale in normalem Poll-Betrieb, d.h. es werden keine INT benötigt (die bei meinem S88-PIC auch nicht mehr frei sind). Zum 3. gibt es auch gar keinen Grund für „Echtzeit-Übertragung“, da die Zentrale keine zeitkritischen Aktionen durchführen muss. Die DS fährt ja selbstständig in die Ziel-Position und muss nicht von der Zentrale / PC angehalten werden. Lediglich die Rückmelde-Signale könnten für die Lok-Positionierung hätten hier einen gewissen Geschwindigkeits-Anspruch, aber auch hier würde die Übertragungsgeschwindigkeit keine Hürde darstellen.

Beispiel: ca. 6 ms je Bit macht bei 32 Bit ganze 384 was immerhin fast 3 Übertragungen je Sekunde bedeutet.

Ich empfehle, bei der fallenden Clock-Flanke das Data einzulesen. Ist hierbei PS = HIGH, so ist die das erste Bit des Übertragungs-Zyklus.

Es werden immer die 4 Byte übertragen. Das erste Bit (Byte 1 / Bit #0) ist durch ein entsprechendes PS-Signal gekennzeichnet.

Die Übertragungs-Bits sind wie folgt definiert:

	Bit #			
Byte 1 Status-Flags	0	1 = DSD2010 aktiv, 0 = DSD2010 bereit (F_TURN_ACTIVE)		
	1	1 = Hall-Sensor aktiv (F_HALL)		
	2	1 = Rückmelde-Eingang 3 aktiv (F_RM_03)		
	3	1 = Rückmelde-Eingang 2 aktiv (F_RM_02)		
	4	1 = Rückmelde-Eingang 1 aktiv (F_RM_01)		
	5	1 = Bühne dreht sich, 0 = Bühne steht (F_TURNING)		
	6	tbd.		
	7	1 = Haus LINKS/RECHTS (1. Hälfte / 2. Hälfte)		
Byte 2 Ist-Position	0	Übergabe der IST-Position (BCD-Codiert)		
	1			
	2			
	3			
	4			
	5			
	6			
	7			
Byte 3 (Soll-Position)	0	Übergabe der SOLL-Position (BCD-Codiert)		
	1			
	2			
	3			
	4		→ derzeit nicht weiter verwendet	
	5			
	6			
	7			
Byte 4 Error-Code	0	Motor open loop	1 = Error	0 = OK
	1	Motor Kurzschluss	1 = Error	0 = OK
	2	Motor Parameter Fehler	1 = Error	0 = OK
	3	Klemmung	1 = Error	0 = OK
	4	Sensor Fehler 1	1 = Error	0 = OK
	5	Sensor Fehler 2	1 = Error	0 = OK
	6	Klemmung 2	1 = Error	0 = OK
	7	Kommunikation mit Bühnen-Platine	1 = Error	0 = OK

2.6.3.2 Übertragung in Richtung Drehscheibe

Hier werden die Gleis-Befehle (DCC oder Motorola) über einen (schnellen) Optokoppler zum Haupt-Prozessor (auf der Haupt-Platine) übertragen werden. Dies ist optional und spart den Anschluss der Gleisspannung als Befehls-Träger an der Haupt-Platine. Allerdings muss dort dennoch diejenige Gleisspannung angeschlossen werden, mit der die Lok auf der Bühne bewegt wird (und die ggf. mithilfe der Kehrschleifen-Funktion der Haupt-Platine für 2-Leiter Bahner in Abhängigkeit der Bühnen-Stellung umgepolt werden muss)

2.7 Ablaufbeschreibung

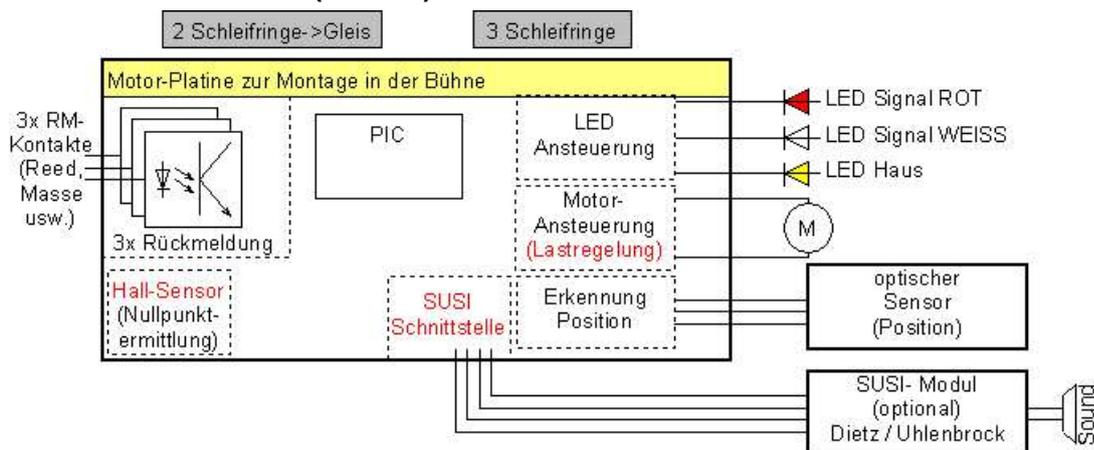
2.7.1 Soll / Ist-Position

In der Haupt-Platine wird die Soll-Position berechnet. Die nötige Info ergibt sich aus den Digital-Befehlen oder den externen Bedien-Möglichkeiten (Anzeige-Modul, PC, alter Bedienschalter).

Mit dem Flag „F_TURN_GO“ beginnt dann die Bühne mit der Drehung bis auf die Ziel-Position. Wird das Flag „F_TURN_GO“ jetzt zurückgenommen, dann gilt das wie ein „STOP“-Befehl und die Bühne hält an der nächsten Position.

Soll der Ablauf nicht abgebrochen werden sondern die Bühne bis zur Ziel-Position fahren, so bleibt das „F_TURN_GO“-Flag bis zum Ende aktiv. Erst wenn die Bühnen-Platine über das Flag „F_TURN_ACTIVE“ das Ende der Bewegung signalisiert, so wird das Flag „F_TURN_GO“ zurückgenommen. Die Bühnen-Platine regiert am Ziel nicht auf das „F_TURN_GO“-Flag, solange es nach der letzten Bewegung nicht inaktiv war. Auch wenn Soll-Position = Ziel-Position, wird keine Bewegung gestartet.

3 Bühnen-Platine (Bühne)



3.1 Aufgaben

Diese Platine wird unter der Bühne montiert und ist klein genug, um in alle Fleischmann-Drehscheiben (H0, TT, N) zu passen.

- Kommunikation mit der Bühnen-Platine
- Steuerung des Motors, Lastregelung
- Steuerung der Lichtsignale (weiches Überblenden bei Bühnenbewegung auf ROT)
- Steuerung der schaltbaren Hausbeleuchtung
- Abfrage von (bis zu) 3 Rückmelde-Kontakten (Optokoppler-Eingänge, z.B. zum Anschluss an Reed-Kontakte, Massesensoren). Der Status der Rückmeldekontakte können via Haupt-Platine auf den Rückmeldebus gegeben werden, wodurch eine Positionierung einer Lok auf der Bühne möglich wird (nötig insbesondere bei Automatik-Betrieb durch eine Steuerungs-Software)
- Ein Hall-Sensor zur Definition der Position NULL kann eingebunden werden. Dadurch kann sich die Bühne selbstständig justieren
- SUSI Schnittstelle für Sound-Modul

3.2 Funktionsblock „Spannungsversorgung“

Die Bühnen-Platine erhält 18V DC (von der Haupt-Platine). Diese wird auf 5V herabgesetzt.

5V:

- PIC
- Optokoppler

18V:

- Motor
- Umschalt-Relais für Motor (Drehrichtung)
- LED-Ausgänge

3.3 Funktionsblock „Hall-Sensor“

ermöglicht ein automatisches Erkennen der „Null-Position“

Hall-Sensor: TLE 4095 (Reichelt)

3.4 Funktionsblock „Motor-Ansteuerung“

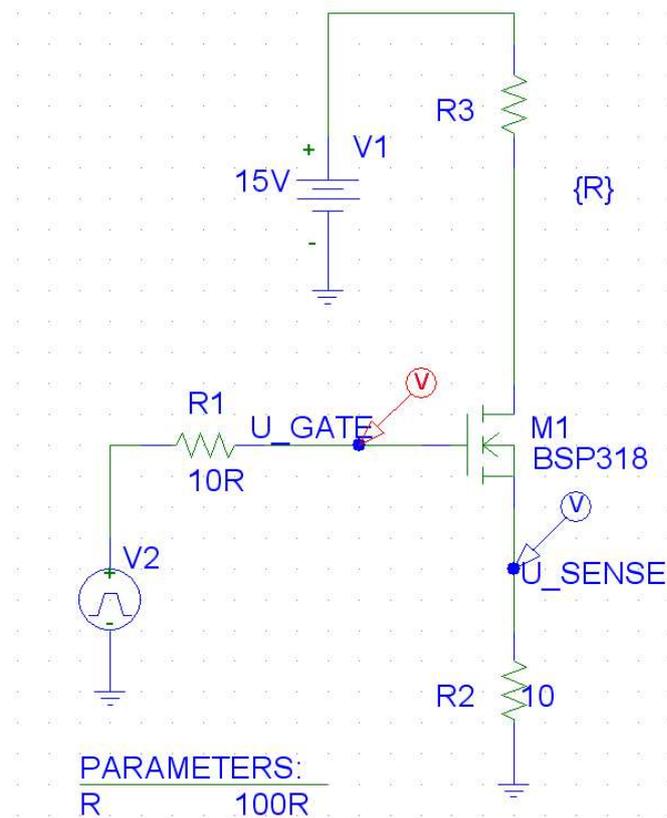
Die Drehrichtungsumkehr erfolgt wie bisher via Relais.

Neu ist jetzt eine Motor-Regelung, die ein genaueres Positionieren während der Endphase der Bewegung (Langsamfahrt!) ermöglicht.

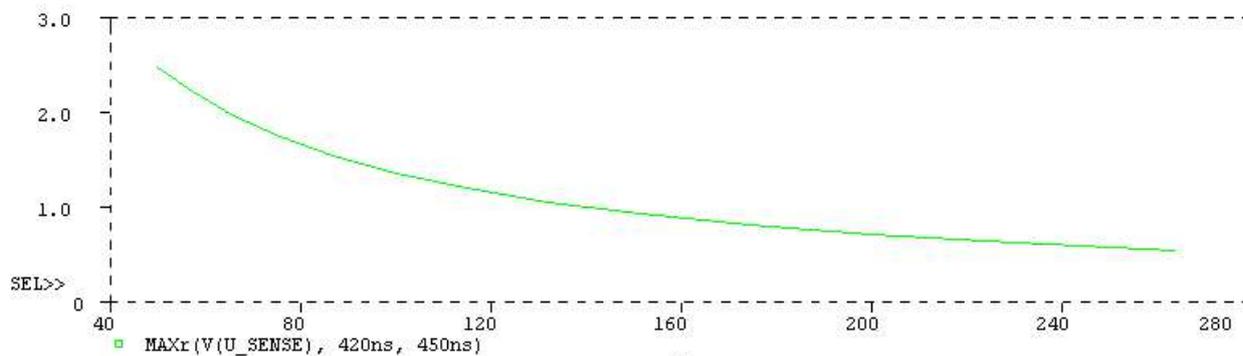
Schalt-Transistor: BSP295

Sense-R: 1W, 15 Ohm

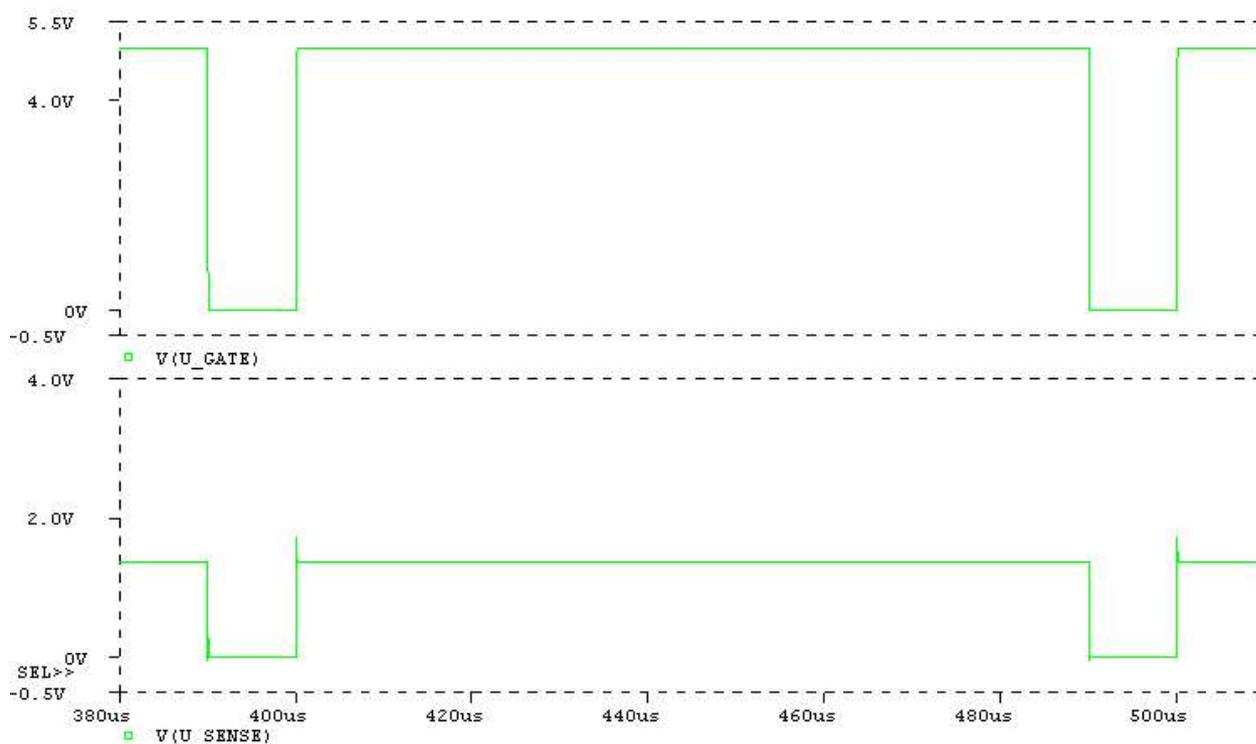
Simulations-Modell:



Es ergibt sich folgender Spannungs-Abfall (Höhe der Spannung während des Ansteuer-Impulses, entspricht y-Achse) bei fallendem Motor-Strom (Motor blockiert ->R3 wird kleiner, entspricht X-Achse)



Betrachtung im Zeitbereich:



3.5 Funktionsblock „Erkennung Position“

Filter:

Fällt die Sensor-Spannung, wird der Counter R_AD_COUNTER um **Filter_s** verkleinert, steigt die Spannung (oder bleibt gleich) wird der Counter um **Filter_b** vergrößert. Kleinere Werte für **Filter_b** und **Filter_s** machen die Erkennung träge, unterdrückt aber Störungen besser.

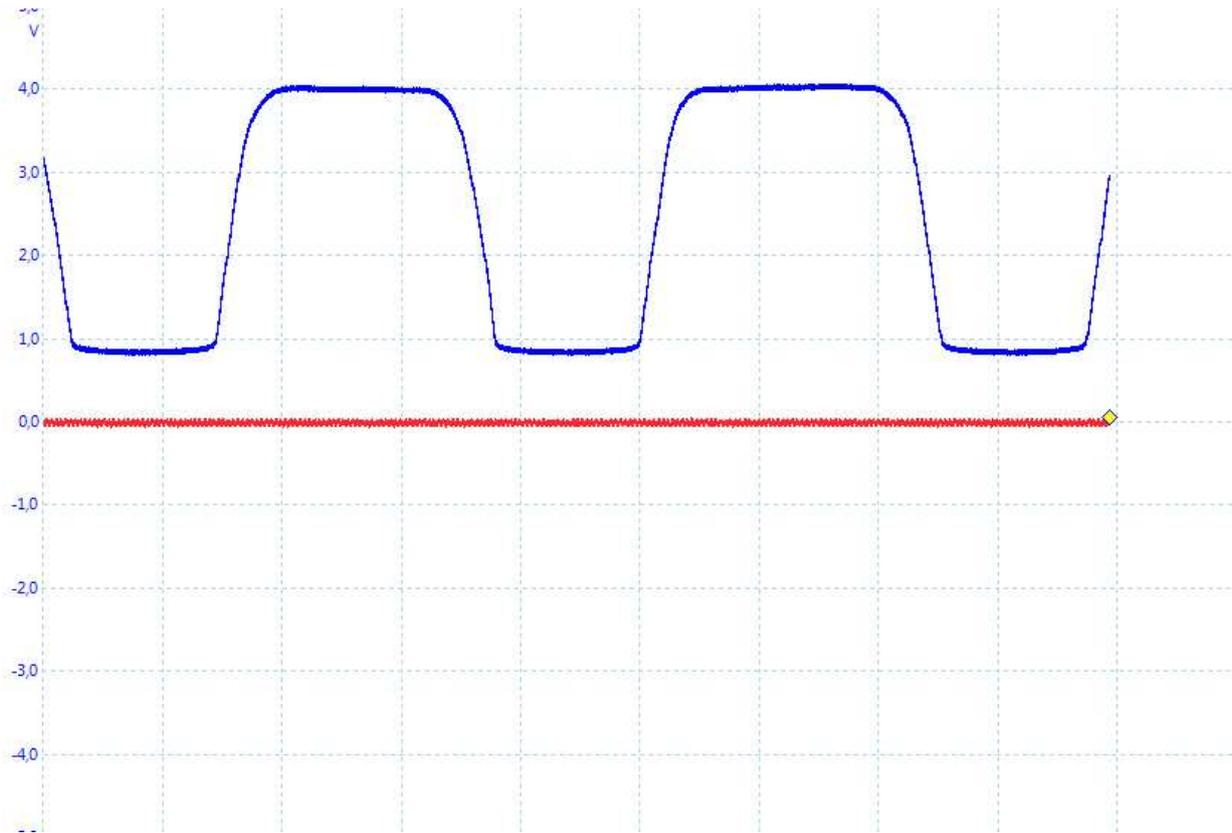


Abbildung 6: Sensor-Eingang PIC Pin 5 mit Pull-Up R5 = 20 kOhm → Sättigung!

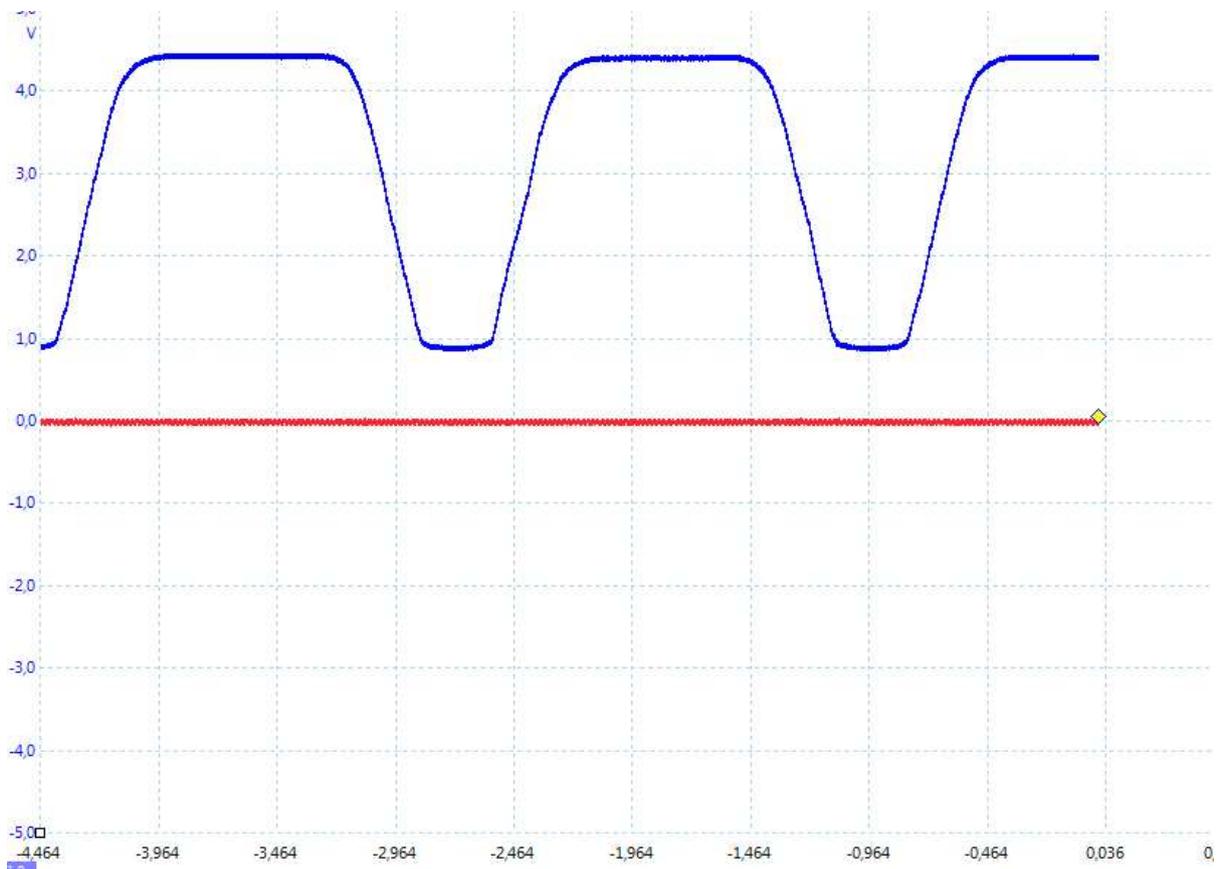


Abbildung 7: Sensor-Eingang PIC Pin 5 mit Pull-Up R5 = 10 kOhm → Sättigung!

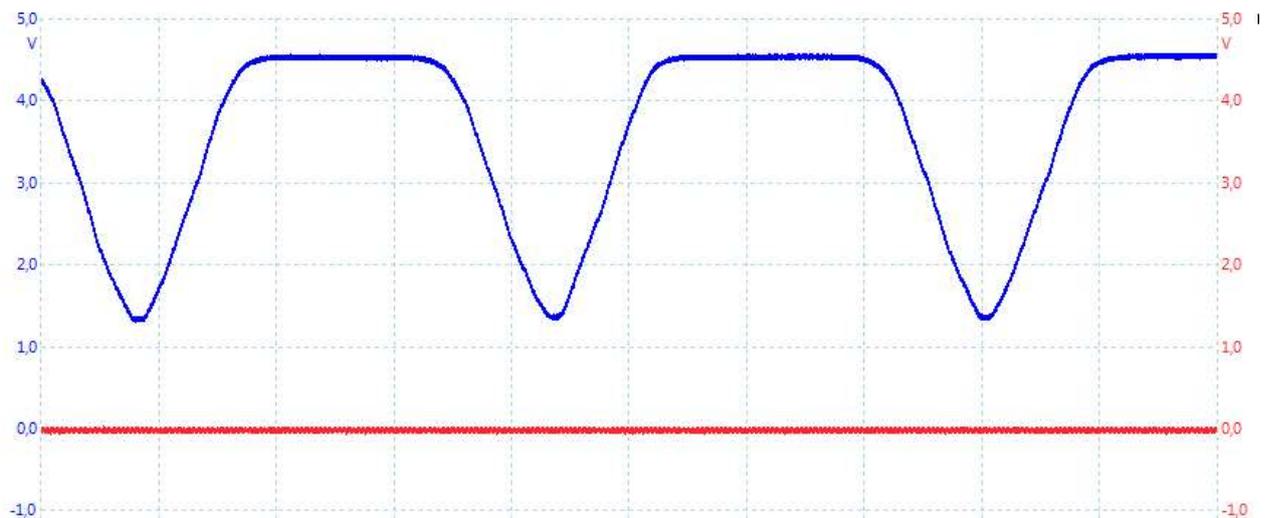


Abbildung 8: Sensor-Eingang PIC Pin 5 mit Pull-Up R5 = 6.8 kOhm

Betrachtung Dunkel @ 6.8 kOhm:

Max = 4.5V . d.h Spannungsabfall an 6.8kOhm = 0.5V → Strom = 74 µA

Betrachtung Hell @ 6.8 kOhm:

Min = 1.3 V. d.h Spannungsabfall an 6.8 kOhm = 3.7V → Strom = 544 µA

Betrachtung Dunkel @ 4.7 kOhm:

Max = 4.7V . d.h Spannungsabfall an 4.7kOhm = 0.3V → Strom = 64 µA

Betrachtung Hell @ 4.7 kOhm:

Min = 2.5 V. d.h Spannungsabfall an 4.7 kOhm = 2.5V → Strom = 532 µA

Betrachtung über den Strom:

teilweise gibt es Sensoren, die liefern ca. 3V bei WEISS @ 4.7 kΩ, d.h. **425 µA**.

bei 5.6 kΩ ergibt dies ein Spannungsabfall von 2.4 V → OK → **Grenzwert minimaler Widerstand**

bei 6.8 kΩ ergibt dies ein Spannungsabfall von 2.9 V → OK

teilweise gibt es Sensoren, die liefern ca. 2V bei WEISS @ 4.7 kΩ, d.h. **638 µA**.

bei 5.6 kΩ ergibt dies ein Spannungsabfall von 3.6 V → OK

bei **6.26 kΩ** ergibt das einen Spannungsabfall von 4 V → **Grenzwert maximaler Widerstand**

bei 6.8 kΩ ergibt dies ein Spannungsabfall von 4.3 V → NOT OK!

→ unter Betrachtung der min / max Werte des vom CNY gelieferten Stromes zwischen 425 µA und 640 µA ist 6.8 kΩ zu groß, 4.7 kΩ zu klein.

3.6 Funktionsblock „Rückmeldung“

Es können 3 Rückmelde-Kontakte eingelesen werden. Die Information wird via Optokoppler übertragen, d.h. es muss eingangsseitig eine LED (die Übertrager-LED im Opto) bestromt werden. Die Opto-Eingänge besitzen hierfür AC-Eingänge (z.B: HCPL 354 = Conrad 140253), d.h. die Polung der Spannung ist unbedeutend, ebenso kann Wechselspannung oder Digital-Spannung verwendet werden.

Als Beispiel: Ein Reed-Kontakt wird einseitig an die Digital-Spannung angeschlossen und an der 2. Seite an den Opto-Eingang. Schliesst jetzt der Kontakt (Magnet), dann wird der Opto-Eingang an die Digital-Spannung gelegt, die Opto-LED ist bestromt und der Rückmelde-Kontakt wird ausgelöst.

Ein ähnliches Beispiel kann man sich für Masse-Sensoren (Kontakt durch Achsen bei Mittelleiter – Märklin möglich).

Weitere Möglichkeiten können sich durch Lichtschranken, Hall-Sensoren, Strom-Sensoren (sinnvoll???) usw. ergeben.

3.7 Funktionsblock „SUSI Schnittstelle“

Info zu SUSI:

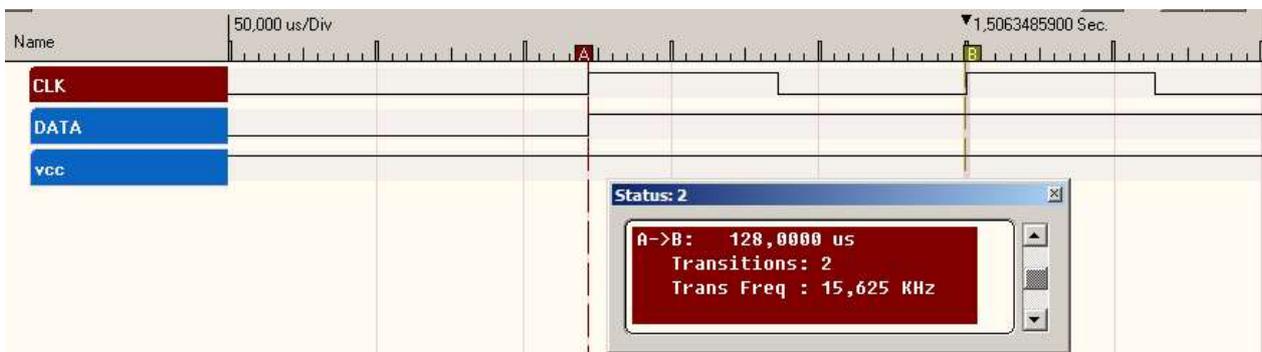
http://www.dccwiki.com/SUSI#SUSI_-_Serial_User_Standard_Interface

Es wird die Ansteuerung eines SUSI Module ermöglicht. Das entsprechende SUSI Modul mit dem Sound für die Drehscheibe ist erhältlich unter

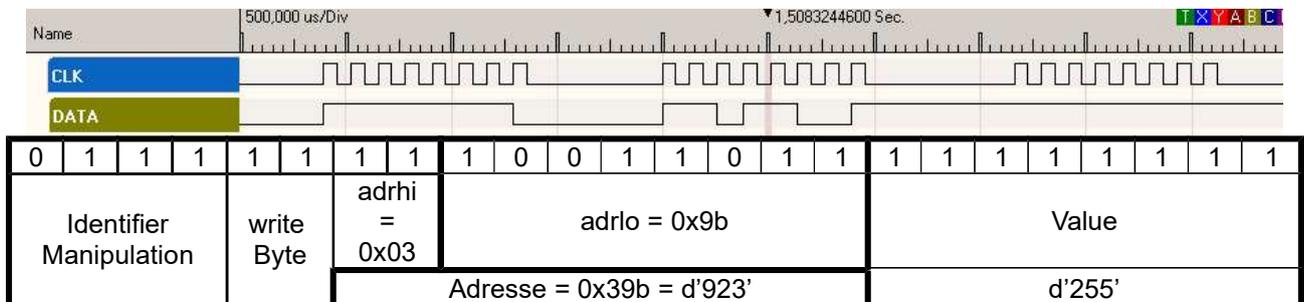
- Uhlenbrock IntelliSound Modul 32400 „Drehscheibe“
- Dietz: DGE MICRO XS „Drehscheibe“
- MÜT noch kein DSD Sound verfügbar

Die Ansteuerung wird entsprechend des Dokumentes „Serial User Standard Interface“ V1.31 (von 10.01.2004) vorgenommen, Download unter http://www.d-i-e-t-z.de/7_5.htm

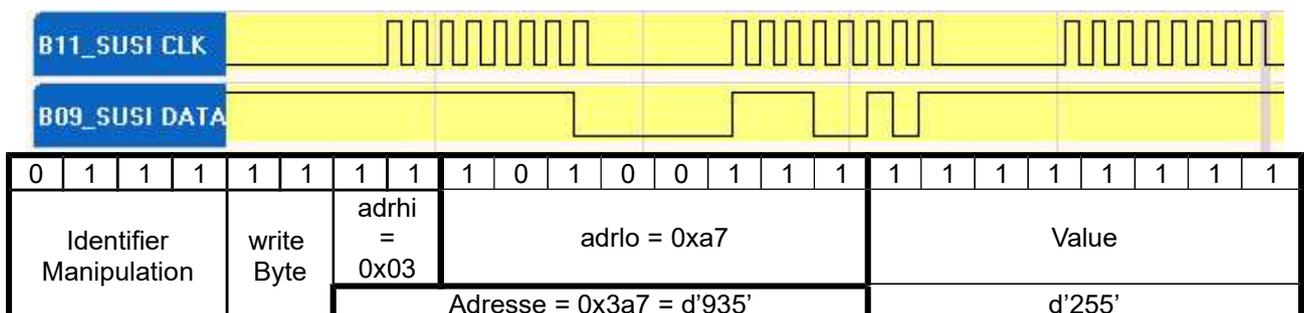
Periode: 128us (15.625 kHz):



Beim Starten wird das **CV924** einmalig mit dem Wert „FF“ beschrieben (ab Bühne V0.22). Dies verhindert, dass beim Reduzieren der Geschwindigkeit im Sound-Modul das „Bremsen“ ausgelöst wird.

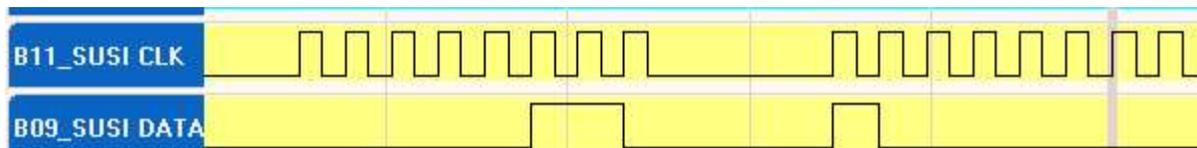


ab Bühne V0.25 wird ebenfalls die **CV936** einmalig mit dem Wert „0xff“ beschrieben. Die neuen SUSI Module 32300 haben die Bremsschwelle in dieser CV abgelegt. Beim alten 32100 ist die CV936 nicht belegt. Das Beschreiben der CV924 beim neuen Modul ist ebenfalls unkritisch, da hier das Mapping für F21 definiert wird und F21 nicht verwendet wird:

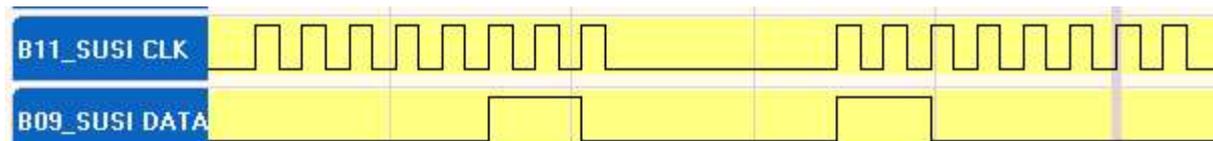


Sound-Ablauf

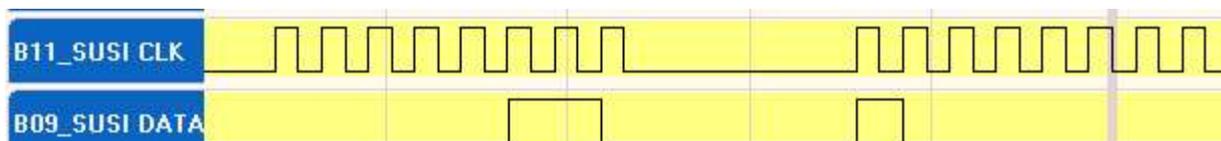
1) F1 EIN: [F0F4] [value] = 01100000 00000001 → LSB First → 00000110 10000000
 → dies führt zum Quietschen = Entriegeln



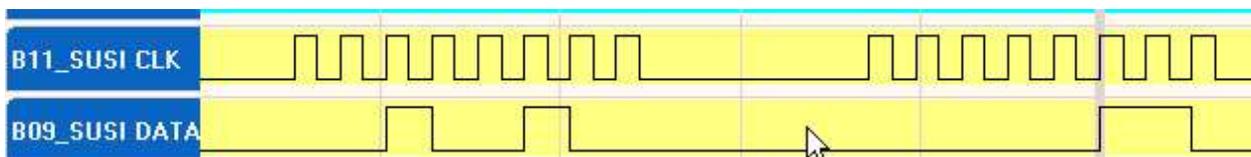
2) zusätzlich F2 EIN: [F0F4] [value] = 01100000 00000011 → LSB First → 00000110 11000000
 → zusätzlich HUPE



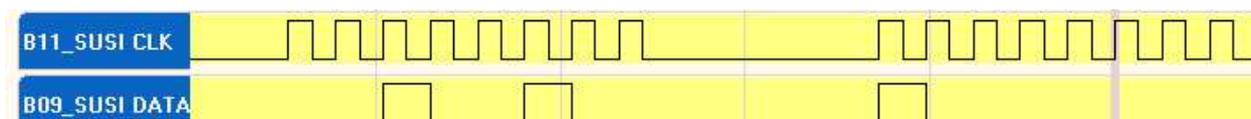
3) F2 AUS: [F0F4] [value] = 01100000 00000001 → LSB First → 00000110 10000000
 → HUPE aus



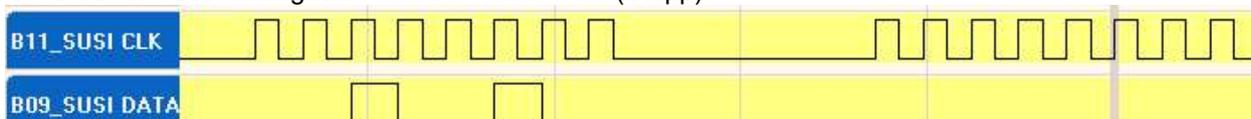
3) Geschwindigkeit: [VLOCO] [value] = 00100100 01100000 → LSB First → 00100100 00000110
 → Fahrt mit Geschwindigkeit 1100000 = 0x60 = 96 (Vorgabe-Wert für SCHNELL)



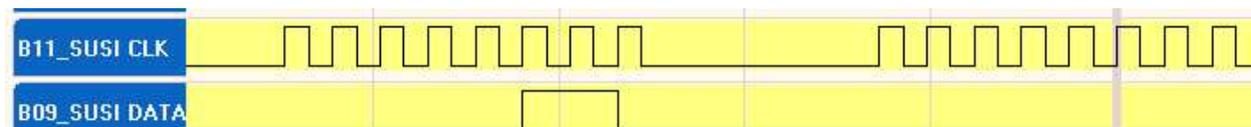
4) Geschwindigkeit: [VLOCO] [value] = 00100100 00000001 → LSB First → 00100100 10000000
 → Fahrt mit Geschwindigkeit 0000001 = 0x01 = 01 (Vorgabe-Wert für Langsam)



5) Geschwindigkeit: [VLOCO] [value] = 00100100 00000000 → LSB First → 00100100 00000000
 → Fahrt mit Geschwindigkeit 0000000 = 0x00 = 00 (Stopp)



6) F1 EIN: [F0F4] [value] = 01100000 00000000 → LSB First → 00000110 00000000
 → F1 aus = → dies führt zum Quietschen = Verriegeln



Es folgt noch HUPE Ein und HUPE Aus

3.8 Ablaufbeschreibung

3.8.1 State Machine

Im Folgenden wird die State Machine beschrieben, die den Ablauf der Bühnen-Bewegung realisiert.

Sta te #	State Name	Funktion	PWM Motor	Sound	L.	L. Sig
0	state_idle	Idle State, d.h. alle Funktionen im Grundzustand state++, wenn Bewegung erforderlich ist (F_TURN_GO gesetzt) F_TURN_ACTIVE = FALSE		aus	-	W
1	state_start_sound	Sound: einschalten state++, sobald fertig F_TURN_ACTIVE = TRUE		F1 → Verriegeln	-	W
2	state_start_weiss	Signale: Weiß ausblenden state++, sobald fertig			B	W ↓
3	state_start_dunkel	Signale: Dunkelphase state++, sobald T_DUNKEL abgelaufen			B	-
4	state_start_rot	Signale: Rot einblenden state++, sobald fertig			B	R ↑
5	state_start_horn	Horn / Hupe beim Start state++, sobald T_HORN_START abgelaufen		Horn oder Hupe	B	R
6	state_start_wait	warte bis Motor-Bewegung state++, sobald T_WAIT_START abgelaufen			B	R
7	state_start_boost	Motor Bewegung mit boost level state++, sobald T_BOOST abgelaufen F_TURNING = TRUE IO_LED_IR = ON	SLOW F_MOTO R_RUN = TRUE	FAST / SLOW @ last Step	B	R
8	state_move_acc	Warten, bis Sensor auf Betriebs-Level state++, V_Sensor < V_MESS : nach Ablauf von T_BLOCK_1 → F_ERR_KLEMM V_Sensor > V_MESS : state++	PWM ↑ @ not last Step	SLOW @ last Step	B	R
9	state_move	Bewegung, Position erfassen V_Sensor > V_Count : nach Ablauf von T_BLOCK_2 → F_ERR_KLEMM2 V_Sensor < V_Count : Positions-INC/DEC @ not last Step > 0 : state_move_ac @ last Step : state_move_stop	PWM ↑ @ not last Step SLOW @ last Step	SLOW @ last Step	B	R
10	state_move_stop	Minimale Geschwindigkeit, Stopp bei nächster Position state++, sobald stopp	PWM = 0 F_MOTO R_RUN = FALSE	FS = 0 STOP	B	R
11	state_stop_wait	warte nach Motor-Bewegung state++, sobald T_WAIT_STOP abgelaufen F_TURNING = FALSE IO_LED_IR = OFF			B	
12	state_stop_rot	Signale: Rot ausblenden state++, sobald fertig			B	R ↓
13	state_stop_dunkel	Signale: Dunkelphase state++, sobald T_DUNKEL abgelaufen			B	-
14	state_stop_weiss	Signale: Weiß einblenden state++, sobald fertig			B	W ↑
15	state_stop_horn	Horn / Hupe bei Ende state++, sobald T_HORN_STOP abgelaufen		Horn oder Hupe	-	W

16	state_stop_sound	Sound: abschalten (F1 aus → Verriegelung öffnen) state_idle (#0), sobald fertig			-	W
----	------------------	---	--	--	---	---

3.9 Kommunikation mit der Haupt-Platine

Die Bühnen-Platine kommuniziert über 1 Leitung mit der Hauptplatine, also bidirektional und seriell.

3.9.1 Bitübertragung (Physical Layer)

„1“ und „0“ sind in Anlehnung an die DCC-Norm kodiert.

Da die Bühnen-Platine den Eingang mit Interrupt einlesen kann, ist hier das Timing in der Übertragungsrichtung Grube → Bühnen-Platine schneller gewählt.

Grube an Bühne	Impuls-Länge	Typ.	Max.
1	Impuls		
	Periode	70 µs	
0	Impuls		
	Periode	500 µs	

Bühne an Grube	Impuls-Länge	Typ.	Max.
1	Impuls		
	Periode	2.5 ms 540 µs	
0	Impuls		
	Periode	44 ms 2.1 ms	

3.9.2 Gruben-Platine an Bühnen-Platine

Folgende Informationen müssen übertragen werden:

- Soll-Position
- Befehl „Stopp“
- Befehl „Go“ (tbd.)
- Flag „Rechts / Links“
- Motor-Parameter zur Anpassung
- Flag „Hausbeleuchtung ein/aus“
- 24/48 Positionen
- Sound an/aus
- Blink-Code (tbd.)

Es werden jeweils 3 Bytes übertragen (=“Info“):

1. Identifier Byte
2. Contents Byte
3. Prüfbyte: EXOR-Verknüpfung aus dem Identifier Byte und dem Contents Byte (analog zu DCC)

Info-Nummer	Identifier Byte ASCII	Identifier Byte hex	Funktion	
1		F7	Flags_01	siehe 3.9.2.1
2		A1	Soll-Position	siehe 3.9.2.2
3		A2	Flags_03	siehe 3.9.2.3
4		C1	Write EEPROM	siehe 3.9.2.4
5		C2	Request EEPROM	siehe 3.9.2.5
6		A7	Direct Drive	siehe 3.9.2.6

Zyklus-Zeit (5 Bytes)

Die Gruben-Platine ist der MASTER in der Kommunikation und initiiert damit jede Kommunikation:

1. MASTER sendet eine Info (3 Bytes)
2. MASTER geht für 200 ms auf Empfang
3. SLAVE sendet eine Info (3 Bytes)
4. MASTER beendet Empfang-Status, nachdem 3 Bytes empfangen wurden oder nach Ablauf der definierten Zeit
5. erneuter Start mit 1.

Der SLAVE sendet also nur, wenn er zuvor eine Info vom Master empfangen hat.

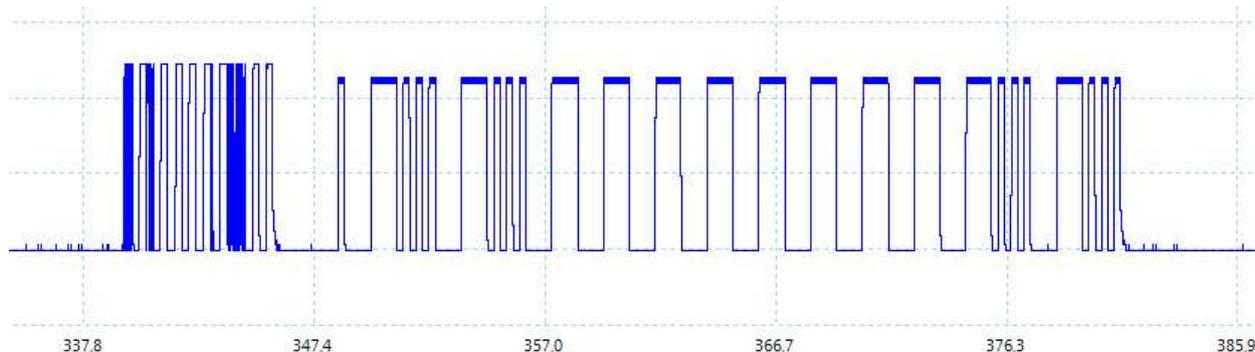


Abbildung 9: Master sendet - Slave antwortet mit Identifier 0x77

Neu-Dimensionierung 03-2016:

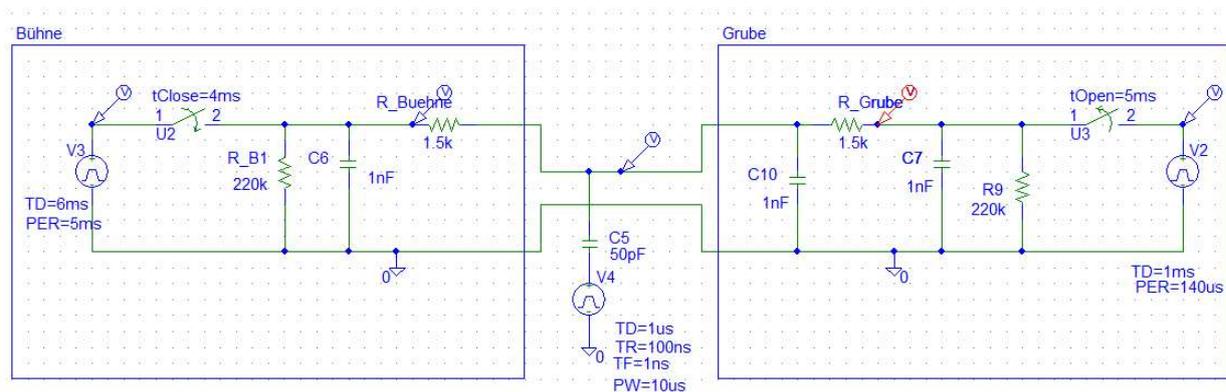


Abbildung 10: Spice Simulation Kommunikation: - Störung bei 50pF Kapazität zu Gleisspannung +- 18V Impulse

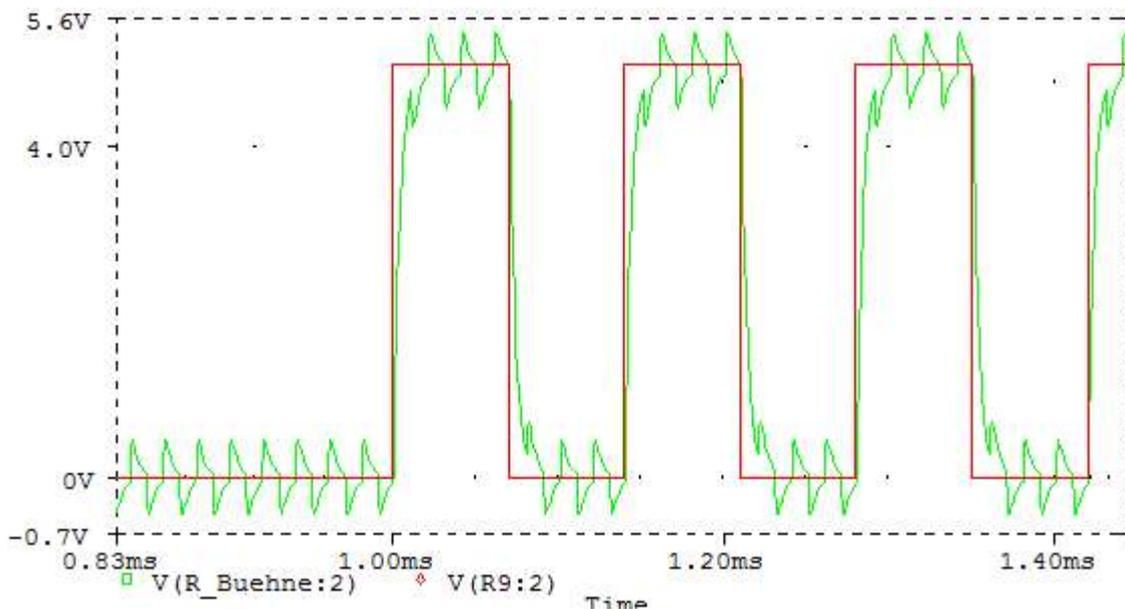


Abbildung 11: Spice Simulation Kommunikation: Grube Sendet an Bühne - Störung bei 50pF Kapazität zu Gleisspannung +- 18V Impulse

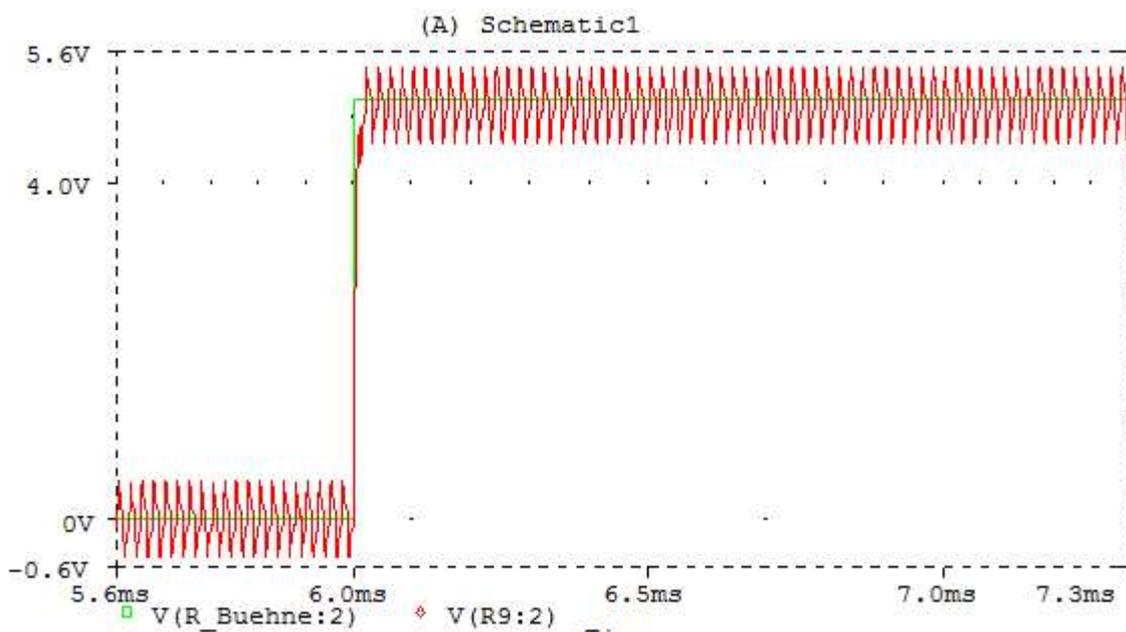


Abbildung 12: Spice Simulation Kommunikation: Bühne Sendet an Grube - Störung bei 50pF Kapazität zu Gleisspannung +- 18V Impulse

Messungen Hardware Grube V1.51 + Bühne V1.51 (1k5 / 1nF)

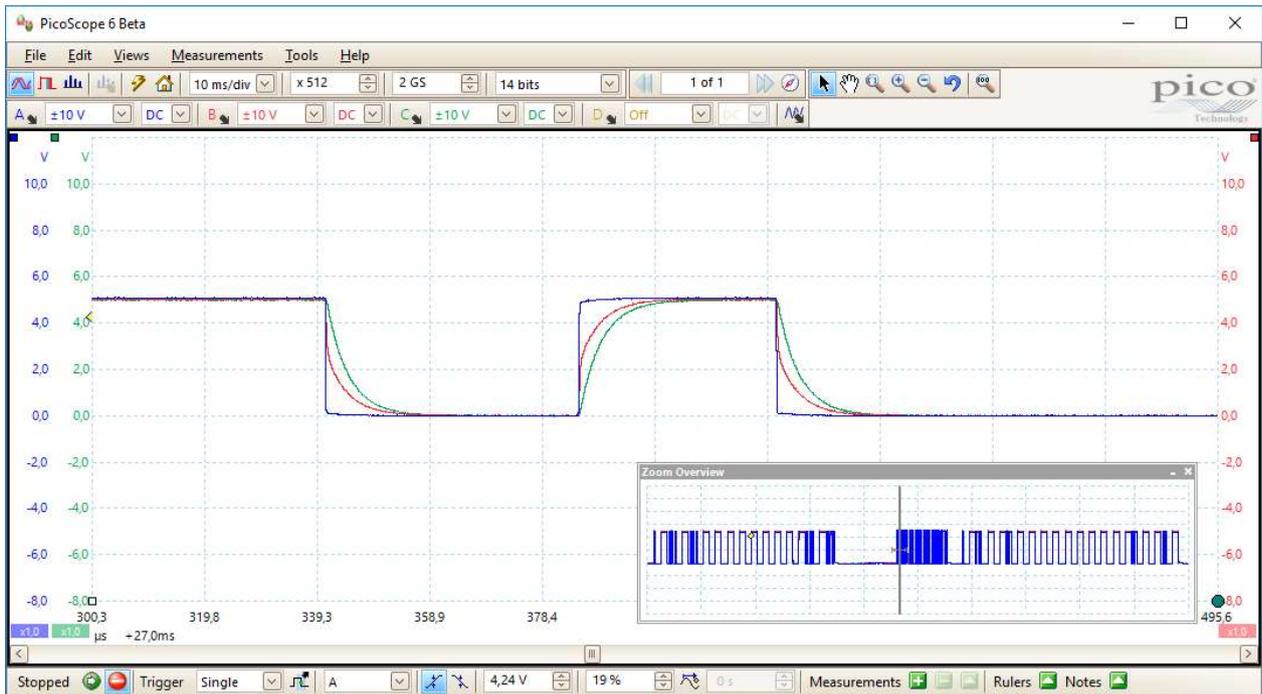


Abbildung 13: Kommunikation Grube → Bühne – Flankendetail / Blau = PIC Grube, ROT = COMM-Line, GRÜN = PIC Bühne

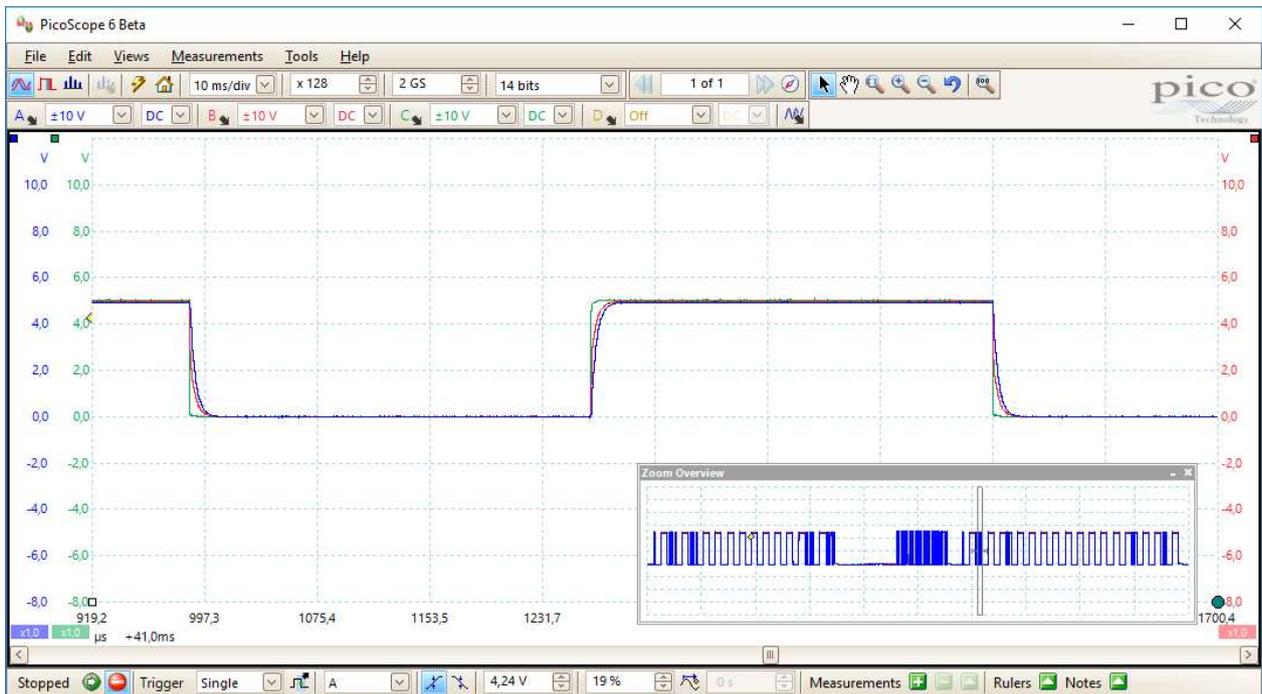


Abbildung 14: Kommunikation Bühne → Grube – Flankendetail / Blau = PIC Grube, ROT = COMM-Line, GRÜN = PIC Bühne

Kompatibilitäts-Betrachtung	Grube 22 k / 47 pF (alt)	Grube 1.5 k / 1 nF (neu)	Grube 1.5 k / 1 nF + 1 nF am Ausgang (neu - Variante 2)
Bühne 22 k / 47 pF (alt)	ja, aber zu hochohmig, kann zu Störung bei Gleisspannung führen, wenn Kabel-Kapazität zwischen Kommunikation und Gleis (+-18V) > 2 pF	unterdrückt Störungen bis ca. 17 pF, eingehende Impulse an der Grube etwas verschliffen (ca. 5 ms Rise-Time)	unterdrückt Störungen bis ca. 57 pF, eingehende Impulse an der Grube etwas verschliffen (ca. 10 ms Rise-Time)
Bühne 1.5 k / 1 nF (neu)	Kommunikations-Problem Empfang Bühne: Signale stark verschliffen, kein Empfang!	unterdrückt Störungen bis ca. 42 pF	unterdrückt Störungen bis ca. 65 pF

3.9.2.1 Info 1, Flags_01

Byte 1 (Identifizier-Byte), h'F7'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	1	0	1	1	1

Byte 2 (Contents-Byte) → R_STAT_FLAG_01 (Flags von Haupt-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.3							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.2.2 Info 2, Soll-Position

Byte 1 (Identifizier-Byte), h'A1'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	0	0	0	0	1

Byte 2 (Contents-Byte) → R_SOLLPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.1							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.2.3 Info 3, Flags_03

Byte 1 (Identifizier-Byte), h'A2'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	0	0	0	1	0

Byte 2 (Contents-Byte) → R_STAT_FLAG_03							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.5							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.2.4 Info 5, Write EEprom Bühne

Byte 1 (Identifizier-Byte), h'C1'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	0	0	0	0	1

Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEprom Adresse 0..255 (8 bit)				LSB	

Byte 3 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEprom Wert 0..255 (8 bit)				LSB	

Das 3. Byte ist hier nicht als Prüfbyte verwendet.

3.9.2.5 Info 6, Request EEprom Bühne

Über diesen Befehl kann der Inhalt einer EEprom Zelle auf der Bühnen-Platine ausgelesen werden,. Dadurch können die hier abgelegten Parameter (siehe 5.4) von der Haupt-Platine abgefragt werden, sobald ein entsprechender Request (siehe 4.2.3.7) vom PC vorliegt. Nach Empfang der Antwort (siehe 3.9.3.6) von der Bühnen-Platine wird der EEprom-Wert an den PC übermittelt (siehe 4.2.2.6)

Byte 1 (Identifizier-Byte), h'C2'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	0	0	0	1	0

Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEprom Adresse 0..255 (8 bit)				LSB	

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.2.6 Info 4, Direct Drive

Byte 1 (Identifizier-Byte), h'A7'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	0	0	1	1	1

Byte 2 (Contents-Byte) → R_STAT_FLAG_03							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 = stehen 1 = drehen	MSB		PWM-Wert 0..127			LSB	

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3 Bühnen-Platine an Haupt-Platine

Folgende Informationen müssen übertragen werden:

- Ist-Position
- Flag „Bühne dreht“
- Error-Codes
 - Klemmung
 - kein Strom (=Open Loop f. Motor-Kreis / Treiber defekt)
 - Sensor Pegel
 - Motor Parameter
- Rückmelde-Kontakte

Es werden jeweils 3 Bytes übertragen (=“Info“):

1. Identifier Byte
2. Contents Byte
3. Prüfbyte: EXOR-Verknüpfung aus dem Identifier Byte und dem Contents Byte (analog zu DCC)

Info-Nummer	Identifier Byte ASCII	Identifier Byte hex	Funktion	
1		77	Flags 02	siehe 3.9.3.1
2		66	Error Code	siehe 3.9.3.2
3		55	Ist-Position	siehe 3.9.3.3
4		44	Sensor-Spannung	siehe 3.9.3.4
5		33	Strom-Wert	siehe 3.9.3.5
6		21	Value EEPROM	siehe 3.9.3.6

Diese Infos 1-5 werden zyklisch (in dieser Reihenfolge) wiederholt, **Zyklus-Zeit** (5 Bytes) = ca. **300 ms**

Info 1 wird sofort bei Status-Änderung in R_STAT_FLAG_02 gesendet, damit insbesondere die Rückmelde-Bits schnell übertragen werden (nötig für PC-Software-Fahren).

Info 6 erfolgt nur in Antwort auf „request EEPROM Bühne“, siehe Info 6, Value EEPROM Bühne

Die Bühnen-Platine stellt in der Kommunikation den SLAVE dar und sendet nur nach Empfang einer Info (3 Bytes) von der Haupt-Platine (siehe auch 3.9.2.)

3.9.3.1 Info 1, Flags_02

Byte 1 (Identifier-Byte), h'77'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	1	0	1	1	1

Byte 2 (Contents-Byte) → R_STAT_FLAG_02 (Flags von Bühnen-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.4							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3.2 Info 2, Error-Code

Error-Code, Byte 1 (Identifizier-Byte), h'66'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	0	0	1	1	0

Byte 2 (Contents-Byte) → R_ERRORCODE_02 (von Bühnen-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.7							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3.3 Info 3, Ist-Position

Ist-Position , Byte 1 (Identifizier-Byte), h'55'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	0	1

Byte 2 (Contents-Byte) → R_ISTPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.2							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3.4 Info 4, Sensor-Spannung

Sensor-Spannung , Byte 1 (Identifizier-Byte), h'44'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	1	0	0

Byte 2 (Contents-Byte) → R_AD_SENSOR							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.8							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3.5 Info 5, Strom-Wert

Strom-Wert , Byte 1 (Identifizier-Byte), h'33'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	1	0	0	1	1

Byte 2 (Contents-Byte) → R_AD_SENSE							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.9							

3. Byte = EXOR aus Byte 1 und Byte 2

3.9.3.6 Info 6, Value EEprom Bühne

Übergabe des Inhaltes einer EEprom Zelle an die Grube (und damit auch an den PC).
Diese Info sendet die Bühne in Antwort auf 3.9.2.5

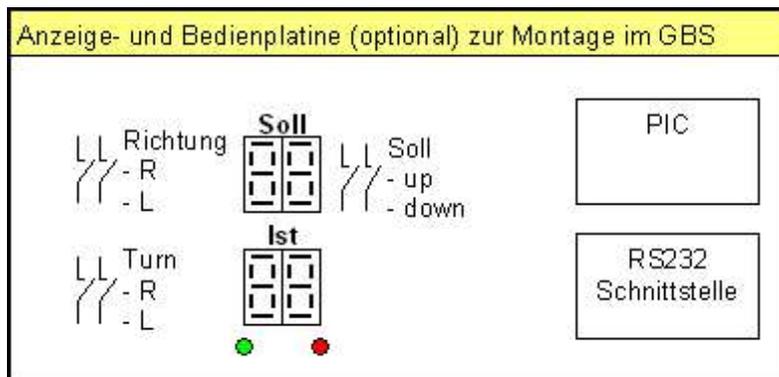
EEprom -Wert , Byte 1 (Identifizier-Byte), h'21'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	0	0	0	0	1

Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB			EEprom Adresse 0..255 (8 bit)				LSB

Byte 3 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB			EEprom Wert 0..255 (8 bit)				LSB

Das 3. Byte ist hier nicht als Prüfbyte verwendet.

4 Anzeige-Platine / PC



Über die RS232 Schnittstelle kann entweder die Anzeige-Platine oder ein PC angeschlossen werden (jedoch nicht gleichzeitig). Die hier definierte RS232 Kommunikation ist daher auch für evtl. externe PC-Anwendungen interessant.

Insbesondere zur Konfiguration (Adressen vergeben, Parameter einstellen) ist die Verbindung zum PC sinnvoll. Auch die Möglichkeiten der Informations-Anzeige und der Bedienung sind via PC deutlich erweitert. Wer jedoch die Bedienung ohne PC vornehmen möchte, der hat ist mit dem Anzeige-Modul eine ausreichende Bedien-Schnittstelle zur Verfügung.

4.1 Aufgaben

Diese Platine ist optional. Sie ermöglicht die Anzeige der Ist- und Soll-Position der Drehscheibe. Zusätzlich kann hierüber die Bühne bewegt werden, ohne dass man Digital-Befehle auslösen muss („Hand-Steuerung“)

- Kommunikation mit der Haupt-Platine (RS232)
- Anzeige der Soll-Position (2-stellig 7-Segment LED)
- Anzeige der Ist-Position (2-stellig 7-Segment LED)

Es stehen 5 Taster, 1 Drehgeber mit Tast-Funktion sowie 4 LEDs zur Verfügung.

Zusätzlich kann die Anzeige-Platine einen Fehler-Code anzeigen, sobald ein Fehler aufgetreten ist. (Codes siehe Anhang)

4.2 Kommunikation mit der Gruben-Platine

Die Gruben-Platine kommuniziert über RS232, also bidirektional (2-Draht) und seriell.

4.2.1 Bitübertragung (Physical Layer)

„1“ und „0“ sind entsprechend der RS232 Norm codiert. Es wird mit einer Baudrate von 9600 Baud übertragen (8N1).

Pausen-Zeit zwischen 2 Bytes: ca. 2 ms

Pausen-Zeit zwischen 2 Infos: ca. 5 ms

ca. 150 ms für kompletten Zyklus aus 5 Infos = 15 Bytes

4.2.2 Gruben-Platine an Anzeige-Platine / PC

Folgende Informationen werden übertragen:

- Soll-Position
- Ist-Position
- Error-Code
- Flag „Rechts / Links“
- Flag „Bühne dreht“ / „Bühne aktiv“
- Einstellung der DIP-Schalter

Es werden jeweils 3 Bytes übertragen (=“Info“), wobei das erste Byte den Identifier darstellt.

Definition der Identifier-Byte:

Info-Nummer	ASCII	hex	Funktion	
0	XYZ	58 59 5A	Synchronisations-Bitmuster *	
1	F	46	Flags 01/02	siehe 4.2.2.1
2	E	45	Error Codes	siehe 4.2.2.2
3	L	4C	Positionen (Location)	siehe 4.2.2.3
4	A	41	Analog-Werte	siehe 4.2.2.4
5	G	47	Value EEPROM Grube	siehe 4.2.2.5
6	H	48	Value EEPROM Bühne	siehe 4.2.2.6
7	O	4F	Balise	

Diese Infos 0..4 werden zyklisch (in dieser Reihenfolge) wiederholt.

Info 5 erfolgt in Antwort auf „Request EEPROM Grube“, siehe 4.2.3.5

Info 6 erfolgt in Antwort auf „Request EEPROM Bühne“, siehe 4.2.3.7

*) Das Syncro-Bitmuster ist einzigartig, da alle 3 Bytes keinem der anderen Identifier-Byte entsprechen. Daher kann keine Info ab Nr. 1 für ein Syncro-Bitmuster gehalten werden.

4.2.2.1 Info 1, Flags_01 / Flags_02

Byte 1 (Identifier-Byte), h'46', ASCII 'F'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	1	1	0

Byte 2 (Contents-Byte) → R_STAT_FLAG_01 (Flags von Haupt-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.3							

Byte 3 (letztes Byte der Info) → R_STAT_FLAG_02 (Flags von Bühnen-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.4							

4.2.2.2 Info 2, Errorcode_01 / _02

Error-Flags, Byte 1 (Identifier-Byte), h'45', ASCII 'E'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	1	0	1

Error-Byte 2 (Contents-Byte) → R_ERRORCODE_01 (von Haupt-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.6							

Error-Byte 3 (letztes Byte der Info) → R_ERRORCODE_02 (von Bühnen-Platine)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.7							

4.2.2.3 Info 3, Positionen

Position, Byte 1 (Identifier-Byte), h'4C', ASCII 'L'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	1	0	0

Position, Byte 2 (Contents-Byte) → R_SOLLPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.1							

Position, Byte 3 (letztes Byte der Info) → R_ISTPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.2							

4.2.2.4 Info 4, Analog-Werte

Analog-Werte, Byte 1 (Identifier-Byte), h'41', ASCII 'A'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	0	0	0	0	1

Analog-Wert, Byte 2 (Contents-Byte) → R_AD_SENSOR							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.8							

Analog-Werte, Byte 3 (letztes Byte der Info) → R_AD_SENSE							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.9							

4.2.2.5 Info 5, Value EEPROM Grube

Übergabe des Inhaltes einer EEPROM (aus der Haupt-Platine - Grube) Zelle an den PC. Diese Info sendet die Haupt-Platine in Antwort auf „Request EEPROM“, siehe 4.2.3.5

Parameter 1, Byte 1 (Identifier-Byte), h'47', ASCII 'G'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	1	1

Parameter 1, Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEPROM Adresse 0..255 (8 bit)				LSB	

Parameter 1, Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEPROM Wert 0..255 (8 bit)				LSB	

4.2.2.6 Info 6, Value EEPROM Bühne

Übergabe des Inhaltes einer EEPROM (aus der Bühnen-Platine - Bühne) Zelle an den PC. Diese Info sendet die Haupt-Platine in Antwort auf „Request EEPROM Bühne“, siehe 4.2.3.7

Parameter 1, Byte 1 (Identifier-Byte), h'48', ASCII 'H'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	1	0	0	0

Parameter 1, Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEPROM Adresse 0..255 (8 bit)				LSB	

Parameter 1, Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		EEPROM Wert 0..255 (8 bit)				LSB	

4.2.3 Anzeige-Platine / PC an Gruben-Platine

Folgende Informationen werden übertragen:

- Soll-Position
- Flag für Drehrichtung „Rechts / Links“
- Diverse Flags (Drehrichtung, GO, Sound on/off etc.)
- Schreiben von Daten ins EEPROM der Haupt-Platine
- Abfrage von Daten aus dem EEPROM der Haupt-Platine
- Schreiben von Daten ins EEPROM der Bühnen-Platine
- Abfrage von Daten aus dem EEPROM der Bühnen-Platine

Es werden jeweils 3 Bytes übertragen (= "Info"), wobei das erste Byte den Identifier darstellt.

Definition des Identifier-Byte (Bit#7 muss ,1' sein).

Definition des Contents-Byte (Bit#7 muss ,0' sein).

Info-Nummer	ASCII	Identifier (hex)	Funktion	
1		C1	Flags_01	siehe 4.2.3.1
2		C2	Flags_03	siehe 4.2.3.2
3		C4	Soll-Position / Ist-Position	siehe 4.2.3.3
4		Ex	nur PC: Write EEPROM Grube	siehe 4.2.3.4
5		Fx	nur PC: Request EEPROM Grube	siehe 4.2.3.5
6		8x	nur PC: Write EEPROM Bühne	siehe 4.2.3.6
7		9x	nur PC: Request EEPROM Bühne	siehe 4.2.3.7
8		Ax	nur PC: Direct Drive	siehe 4.2.3.8

ToDo: Manipulation von RAM-Zelle in Grube / Bühne (INDF Befehl dort). Damit kann z.B. eine Übertragung des aktuellen Datensatzes initiiert werden, wenn später einmal nur die geänderten Daten übertragen werden. PC kann dann z.B. beim Start den aktuellen Datensatz anfordern. Auch „Direct Drive“ kann so abgearbeitet werden sowie diverse Flags (Lichtsignale schalten etc.)

Diese Infos werden bei Bedarf (also bei Zustands-Änderung) herausgegeben, d.h. es werden vom PC zur Grube nur sporadisch Daten übertragen.

4.2.3.1 Info 1, Flags_01

Mit diesem Befehl kann

- ⇒ die Drehung gestartet werden (Flag F_TURN_GO setzen)
- ⇒ die Drehung gestoppt werden (Flag F_TURN_GO löschen)
- ⇒ die Drehrichtung bestimmt werden (Flag F_TURN_DIR)
- ⇒ die Lichtfunktion geschaltet werden (Flag F_LIGHT_ON)

Byte 1 (Identifizier-Byte), h'C1'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	0	0	0	0	1

Byte 2 (Contents-Byte) -> R_STAT_FLAGS_01							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.3							

Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Flag Filter eine "1" bedeutet, dass das Flag aus dem Contents-Byte gültig und damit zu übernehmen ist. Alle anderen Flags sind zu ignorieren. Realisierung: über eine logische UND-Verknüpfung beim Empfänger						

Beispiel 1: Licht einschalten

1. senden von **0xC1**
2. senden von b'0000 0001' = **0x01**
3. senden von b'0000 0001' = **0x01**

Beispiel 2: Licht ausschalten

1. senden von **0xC1**
2. senden von b'0000 0000' = **0x00**
3. senden von b'0000 0001' = **0x01**

4.2.3.2 Info 2, Flags_03

Mit diesem Befehl kann

- ⇒ eine Soundfunktion ausgelöst werden (F_HUPE und F_HORN)
- ⇒ das Blinklicht geschaltet werden (F_FLASH)

Byte 1 (Identifizier-Byte), h'C2'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	0	0	0	1	0

Byte 2 (Contents-Byte) → R_STAT_FLAG_03							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.5							

Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Flag Filter eine "1" bedeutet, daß das Flag aus dem Contents-Byte gültig und damit zu übernehmen ist. Alle anderen Flags sind zu ignorieren. Realisierung: über eine logische UND-Verknüpfung beim Empfänger						

4.2.3.3 Info 3, Soll-Position und Ist-Position

Die Sollposition kann jederzeit an DSD2010 übergeben werden. Eine Drehung wird erst durch das Flag „F_TURN_GO“ gestartet (in R_STAT_FLAG_01, siehe 4.2.3.1).

Soll-Position, Byte 1 (Identifizier-Byte), h'C4'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	0	0	1	0	0

Soll-Position, Byte 2 (Contents-Byte) → R_SOLLPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.1							

Soll-Position, Byte 3 (letztes Byte der Info) → R_ISTPOSITION							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
→ siehe 5.2.2							

4.2.3.4 Info 4, Write EEPROM Grube

Schreiben von Werten ins EEPROM der Haupt-Platine.
Hierdurch können insbesondere die Adressen (gespeichert im EE der Haupt-Platine) geändert werden.

Write EEPROM, Byte 1 (Identifizier-Byte), h'Ex'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	MSB von Byte 3 (DATA)	MSB von Byte 2 (ADR)

Write EEPROM, Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 EEPROM Adresse 0..127 (7 bit), MSB in Byte 1							LSB

Write EEPROM, Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 EEPROM Wert 0..127 (7 bit), MSB in Byte 1							LSB

4.2.3.5 Info 5, Request EEPROM Grube

Abfrage von Werten des EEPROM der Haupt-Platine.
Hierdurch können insbesondere die Adressen (gespeichert im EE der Haupt-Platine, siehe 5.3) vom PC ausgelesen werden.

Request EEPROM, Byte 1 (Identifizier-Byte), h'Fx'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	1	0	0	0	MSB von Byte 2

Request EEPROM, Byte 2 (Contents-Byte)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 EEPROM Adresse 0..255 (8 bit)							LSB

Request EEPROM, Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0			0			LSB	

4.2.3.6 Info 6, Write EEPROM Bühne

Schreiben von Werten ins EEPROM der Bühnen-Platine.

Hierdurch können insbesondere die Motor-Parameter / Geschwindigkeits-Parameter (gespeichert im EE der Haupt-Platine, siehe 5.4) geändert werden.

Write EEPROM, Byte 1 (Identifizier-Byte), h'8x'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	0	0	0	MSB von Byte 3 (DATA)	MSB von Byte 2 (ADR)

Write EEPROM, Byte 2 (Contents-Byte)								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0							EEprom Adresse 0..127 (7 bit), MSB in Byte 1 #1	LSB

Write EEPROM, Byte 3 (letztes Byte der Info)								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0							EEprom Wert 0..127 (7 bit), MSB in Byte 1 #0	LSB

4.2.3.7 Info 7, Request EEPROM Bühne

Abfrage von Werten des EEPROM der Haupt-Platine.

Hierdurch können insbesondere die Motor-Parameter / Geschwindigkeits-Parameter / Sound-Parameter (gespeichert im EE der Bühnen-Platine, siehe 5.4) vom PC ausgelesen werden.

Request EEPROM, Byte 1 (Identifizier-Byte), h'9x'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	1	0	0	0	MSB von Byte 2

Request EEPROM, Byte 2 (Contents-Byte)								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0							EEprom Adresse 0..127 (7 bit), MSB in Byte 1 #0	LSB

Request EEPROM, Byte 3 (letztes Byte der Info)								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0							0	LSB

4.2.3.8 Info 8, Direct Drive

Direct Drive: es wird ein PWM-Wert vorgegeben, mit dem der Motor direct bewegt wird.

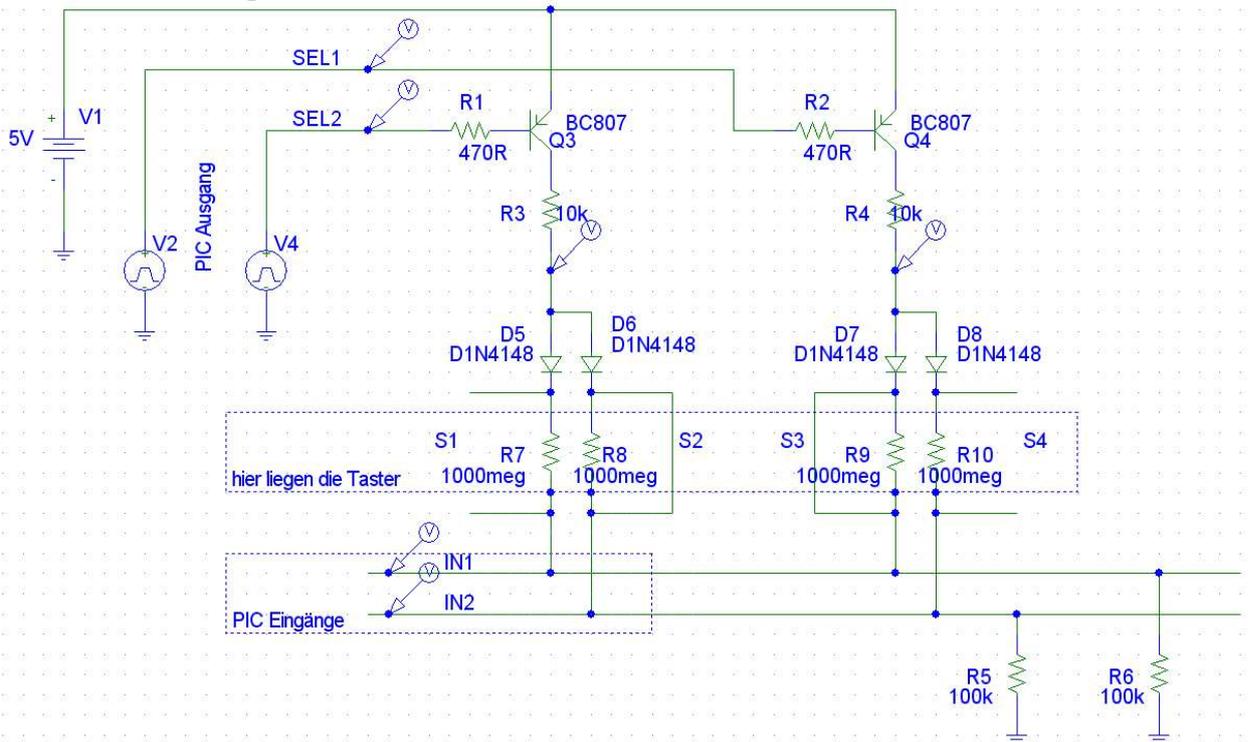
Direct Drive, Byte 1 (Identifizier-Byte), h'a0'							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	0	0	0	0	0

Direct Drive, Byte 2 (Contents-Byte)								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0							PWM-Wert 0..127	LSB

Direct Drive, Byte 3 (letztes Byte der Info)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	0	0	0=ohne LR 1=mit LR	0 = Rechts 1 = LINKS	0 = stehen 1 = drehen

4.3 Design: Multiplex Taster und Anzeigen

Simulations-Schaltung:



Ansteuerung der Ausgänge: LOW Aktiv:



Beispiel 1: S1 open / S2 open / S3 closed / S4 closed

Sel 1 = aktiv (LOW)	IN1 = inaktiv (LOW)	IN2 = inaktiv (LOW)
Sel 2 = aktiv (LOW)	IN1 = aktiv (HIGH)	IN2 = aktiv (HIGH)



Beispiel 2: S1 closed / S2 open / S3 closed / S4 closed

Sel_1 = aktiv (LOW)	IN1 = aktiv (HIGH)	IN2 = inaktiv (LOW)
Sel_2 = aktiv (LOW)	IN1 = aktiv (HIGH)	IN2 = aktiv (HIGH)



Beispiel 3: S1 closed / S2 closed / S3 closed / S4 closed

Sel_1 = aktiv (LOW)	IN1 = aktiv (HIGH)	IN2 = aktiv (HIGH)
Sel_2 = aktiv (LOW)	IN1 = aktiv (HIGH)	IN2 = aktiv (HIGH)



5 Anhang

5.1 Fehler-Codes

Die Anzeige-Platine zeigt folgende Fehler-Codes an:

Anzeige:	Bedeutung	Erläuterung
E1	Communication Error F_ERR_COM_M	Haupt-Platine kann mit der Bühnen-Platine nicht kommunizieren
E2	Motor Error: Open Loop F_ERR_MOT1	Bühnen-Platine kann keinen Motor-Strom messen („Open Loop“ oder Motor-Treiber defekt) → SENSE-Spannung zu klein
E3	Motor Error: Short F_ERR_MOT2	Bühnen-Platine erkennt Kurzschluss am Motor-Ausgang → SENSE-Spannung zu groß
E4	Motor Parameter F_ERR_MOT3	Bühnen-Platine: Motor-Parameter unstimmig
E5	Motor Klemmung F_ERR_KLEMM	Bühnen-Platine erkennt eine Einklemmung während der Start-Phase → kein Sensor-Impuls
E6	Sensor Pegel F_ERR_SENS1	Bühnen-Platine: Sensor-Pegel unterschreitet Minimal-Wert → SENSOR Min-Wert zu klein
E7	Sensor Pegel F_ERR_SENS2	Sensor-Pegel Max-Wert ist zu niedrig → SENSOR Max-Wert zu klein
E8	Motor-Klemmung F_ERR_KLEMM2	Bühnen-Platine erkennt Einklemmung während der Bewegung → kein Sensor Impuls
E9		

5.2 Beschreibung der globalen Register

Verschiedene Register werden zwischen den einzelnen Platinen kommuniziert. Diese Register sowie die entsprechende Source-Platine werden hier beschrieben.

5.2.1 R_SOLLPOSITION

R_SOLLPOSITION: Sollposition der Bühne							
Anzeige / PC	↔	Grube	→	Bühne			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Drehung 0 = normal 1 = endlos	MSB		Soll-Position 1..48 (6 bit)		LSB	

5.2.2 R_ISTPOSITION

R_ISTPOSITION: Istposition der Bühne							
Anzeige / PC	←	Grube	←	Bühne			
		↓					
		S88-N					
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	MSB		Ist-Position 1..48 (6 bit)		LSB	

5.2.3 R_STAT_FLAG_01

R_STAT_FLAG_01: Flags von Haupt-Platine				
Anzeige / PC	↔	Grube	→	Bühne

Die Bits können von Anzeige-Platine / PC manipuliert werden

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F_SEC_HALF IST-Pos. 0= 1.Hälfte 1= 2.Hälfte	F_TURN_GO Bewegung 0 = stop 1 = go	F_RELAIS Kehrschl.- Relais 0 = aus 1 = ein	F_NORM (DIP3) Zielbereich nung 0 = 24 * 1 = 48 *	F_DCC (DIP2) Protokoll 0 = DCC 1 = MM	F_24POS (DIP1) Pos.-Anz. 0 = 48 Pos 1 = xx Pos (xx aus EE)	F_TURN_DIR Drehricht. 0 = Rechts 1 = Links	F_LIGHT_ON Licht Haus 0 = aus 1 = ein

(*) Diese Werte gelten für eine 48er Drehscheibe.

1 = jede Stellung hat einen eigenen Befehl. Es wird also mit 48 Positionen (bei einer 48er Drehscheibe) gearbeitet. Es gibt z.B. Befehle für die Position 4 und 28 - jeweils hier kommt dann das Haus zum Halten. In dieser Betriebsart kann es demnach vorkommen, dass die Bühne 47 Positionen fährt (z.B. von 1 auf 2 links herum)

0 = Kompatibilitäts-Modus: berechnen der Zielposition Märklin-Kompatibel, d.h. es werden nur 24 Positionen (bei einer 48er Drehscheibe) verwendet. Die genaue Vorgabe, welche Seite der Bühne an diesem Gleis zum Halten kommt, ergibt sich aus der Drehrichtung (1. Befehl) und der Ziel-Position (2. Befehl). Die Bühne wird demnach maximal 23 Positionen fahren und hält dann, wenn eines der Bühnen-Enden am Zielgleis angekommen ist.

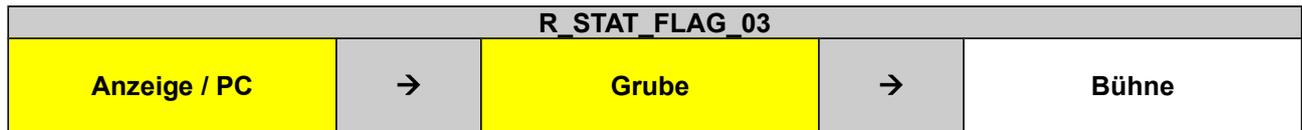
5.2.4 R_STAT_FLAG_02

R_STAT_FLAG_02: Flags von Bühnen-Platine				
Anzeige / PC	←	Grube	←	Bühne
		↓		
		S88-N (nur #6..#0)		

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
! dieses Bit wird nicht zum S88 übertragen !	F_DONE Pos. Impuls nach Ende	F_TURNING Bühne 0 = steht 1 = dreht	F_RM_01 Rückm.1 0 = idle 1 = set	F_RM_02 Rückm.2 0 = idle 1 = set	F_RM_03 Rückm.3 0 = idle 1 = set	F_HALL Hall 0 = idle 1 = aktiv	F_TURN_ACTIVE DSD2010 0 = idle 1 = aktiv

F_TURN_ACTIVE ist 1 während des gesamten Dreh-Ablaufes, während F_TURNING nur während der wirklichen Drehung 1 ist (ermöglicht Synchronisation mit PC-Steuerungen), Bei F_TURN_ACTIVE nimmt DSD keine neuen Steuerungsbefehle an mit Ausnahme von Abbruch-Befehlen.

5.2.5 R_STAT_FLAG_03

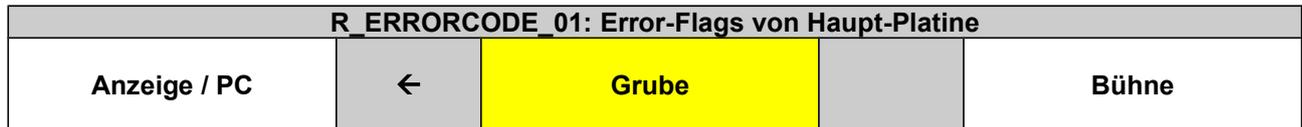


Die Bits können von Anzeige-Platine / PC manipuliert werden

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	F_DEBUG	F_SW2	F_USE_LR	F_FLASH	F_Sound	F_HUPE	F_HORN
	Debug Mode	SW2 Anzeige	Lastregelung. (*)	Blinker	Sound	Hupe	Horn
	1=DEBUG	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

(*) F_USE_LR hat nur bei Direct Drive eine Funktion. Im Betrieb wird dies über die Parameter im EEprom PB1 gesteuert!

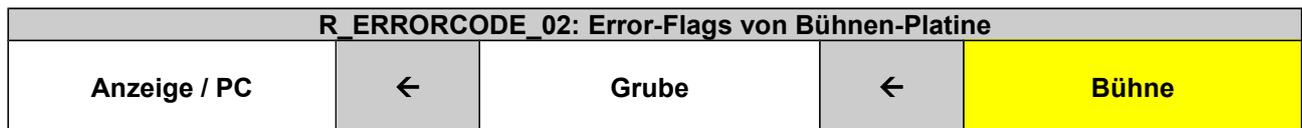
5.2.6 R_ERRORCODE_01



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			F_ERR_NOACTIV	F_ERR_ABORT	RS232_OK	RS232_FRAME	F_ERR_COM_M
			kein AKTIV von Bühne	Drehung durch Befehl abgebr.	Byte Empfang	Frame Error	Komm. Bühnen-Platine
			0 = OK 1 = ERR	1 = ABR	0 = Idle 1 = OK	0 = OK 1 = ERR	0 = OK 1 = ERR

F_ERR_ABORT: hinzu Grube V0.38, wird gesetzt, wenn die Drehung durch den Befehl STEP abgebrochen wurde (für Fehlersuche Abbruch @ TC / WDP)

5.2.7 R_ERRORCODE_02



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F_ERR_RESET	F_ERR_KLEMM2	F_ERR_SENS2	F_ERR_SENS1	F_ERR_KLEMM	F_ERR_MOT3	F_ERR_MOT2	F_ERR_MOT1
Bühne hatte RESET	Klemmung Move	Sensor	Sensor	Klemmung Acc.	Motor Para	Motor short	Motor open
	1 = ERR 0 = OK						

5.2.8 R_AD_SENSOR

R_AD_SENSOR: Analog-Werte Sensor (Reflex)				
Anzeige / PC	←	Grube	←	Bühne

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		Sensor-Spannung 0..255 (8 bit) → R_AD_SENSOR				LSB	

siehe auch 3.9.3.4

5.2.9 R_AD_SENSE

R_AD_SENSE: Analog-Werte Sense (Motor-Strom)				
Anzeige / PC	←	Grube	←	Bühne

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB		Sense-Strom 0..255 (8 bit) → R_AD_SENSE				LSB	

5.3 EEprom Belegung - Grube

Im EEprom der Grube sind die wesentlichen Parameter des Systems angelegt:

- Adressen (24+4) und deren zugeordnete Position
- Parameter

Die Belegung ist so realisiert, dass die Bitmuster für die Adressen in MM im Speicherbereich ab 0x80 liegen. Es muss demnach bei MM lediglich das #7 gesetzt werden, um auf die passende Adresse zuzugreifen.

DCC

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	SW-Version	AKT 5	AKT 6	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08	
			0xff	0xff	0x40	0x42	0x44	0x46	0x42	0x43	0x40	0x41	0x46	0x47	0x44	0x45
10	AKT 1	AKT 2	AKT 3	AKT 4	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08
	0xff	0xff	0xff	0xff	0xfe	0xf6	0xfa	0xfe	0xf0	0xf2	0xf4	0xf6	0xf8	0xfa	0xfc	0xfe
20	OFS	SF1 (SF2)	SF3	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08	
	0x00	0xff	0xff	0x00	0xf1	0xe5	0xe5	0xe5	0xe9							
30	AP	PG1	PG3	--	--	--	--	P01	P02	P03	P04	P05	P06	P07	P08	
	0x18	0x00	0x01	0x00	0xff	0xff	0xff	0xff	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
					225 g	226 g	227 g	228 g	229 r	229 g	230 r	230 g	231 r	231 g	232 r	232 g

40	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0x43	0x42	0x41	0x40	0x47	0x46	0x45	0x44	0x44	0x45	0x46	0x47	0x40	0x41	0x42	0x43
50	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0xf0	0xf2	0xf4	0xf6	0xf8	0xfa	0xfc	0xfe	0xf0	0xf2	0xf4	0xf6	0xf8	0xfa	0xfc	0xfe
60	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0xed	0xf1														
70	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
	0x08	0x09	0x0a	0x0b	0x0c	0x0d	0x0e	0x0f	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
	233 r	233 g	234 r	234 g	235 r	235 g	236 r	236 g	237 r	237 g	238 r	238 g	239 r	239 g	240 r	240 g

MM

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
80	DBG	DBG 2	--	--	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08
	0x7a	0x7b	0xff	0xff	0x00	0x00	0x00	0x00	0x03							
90	--	--	--	--	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08
	0xff	0xff	0xff	0xff	0xc8											
a0	--	--	--	--	AL	AS	AT	AD	A01	A02	A03	A04	A05	A06	A07	A08
	0xff	0xff	0xff	0xff	0xc3	0xf3	0xcf	0xff	0x03	0xc3	0x33	0xf3	0x0f	0xcf	0x3f	0xff
b0	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff
					225 g	226 g	227 g	228 g	229 r	229 g	230 r	230 g	231 r	231 g	232 r	232 g

c0	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0x02	0x00														
d0	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0xc8	0x88														
e0	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
	0x03	0xc3	0x33	0xf3	0x0f	0xcf	0x3f	0xff	0x03	0xc3	0x33	0xf3	0x0f	0xcf	0x3f	0xff
f0	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	0xff															
	233 r	233 g	234 r	234 g	235 r	235 g	236 r	236 g	237 r	237 g	238 r	238 g	239 r	239 g	240 r	240 g

Name	EE-prom Zelle	Default Wert	Beschreibung																								
AL			Adresse Licht DCC				Motorola																				
AS			Adresse STEP DCC				Motorola																				
AT			Adresse TURN DCC				Motorola																				
AD			Adresse DIR DCC				Motorola																				
A1..A24			Adressen Positionen DCC				Motorola																				
P01.. P24			<table border="1"> <thead> <tr> <th>#7</th> <th>#6</th> <th>#5</th> <th>#4</th> <th>#3</th> <th>#2</th> <th>#1</th> <th>#0</th> </tr> </thead> <tbody> <tr> <td colspan="2">Es kann hier der Adresse eine feste Fahrtrichtung zugeordnet werden</td> <td colspan="6" rowspan="2">Position es kann hier die Position 48stellig definiert werden, d.h. bei der 48er DS sind auch 48 Positionen definierbar</td> </tr> <tr> <td>Richtung definiert 1 = ja 0 = nein, d.h. ignore #6</td> <td>0 = links 1 = rechts</td> </tr> </tbody> </table>							#7	#6	#5	#4	#3	#2	#1	#0	Es kann hier der Adresse eine feste Fahrtrichtung zugeordnet werden		Position es kann hier die Position 48stellig definiert werden, d.h. bei der 48er DS sind auch 48 Positionen definierbar						Richtung definiert 1 = ja 0 = nein, d.h. ignore #6	0 = links 1 = rechts
#7	#6	#5	#4	#3	#2	#1	#0																				
Es kann hier der Adresse eine feste Fahrtrichtung zugeordnet werden		Position es kann hier die Position 48stellig definiert werden, d.h. bei der 48er DS sind auch 48 Positionen definierbar																									
Richtung definiert 1 = ja 0 = nein, d.h. ignore #6	0 = links 1 = rechts																										

Name	EE-prom Zelle	Default Wert	Beschreibung																																				
Version	0x00/ 0x01	Version des HEX GRUBE	Software Version der Gruben Platine. Ab V0.30 wird dieser Wert nicht mehr wirklich aus dem EEPROM geholt, sondern aus dem ROM der Grube. Dadurch erhält man immer die korrekten Daten, auch nach einer EEPROM Manipulation oder nach einem Update des HEX ohne Überschreiben der EE-Daten																																				
AKT1	0x10	0xFF	Hier werden die aktiven Gleise gespeichert. Diese Definition wird PC-Programm vorgenommen und kann von der Anzeige-Platine für die Auswahl der möglichen (d.h. auswählbaren) Gleise verwendet werden																																				
AKT2	0x11	0xFF																																					
AKT3	0x12	0xFF																																					
AKT4	0x13	0xFF																																					
AKT5	0x02	0xFF																																					
AKT6	0x03	0xFF																																					
OFFSET	0x20	0x00	Offset Wert für Kehrschleifenrelais (ab Grube V0.33)																																				
SF1	0x21	0xFF	Hier speichert die Grube den letzten Status von R_STAT_FLAG_01, siehe 5.2.3 Hierdurch wird insbesondere ermöglicht, den Status des Hauslichtes/KS-Relais/TURN_DIR beim Einschalten der Grube wieder herzustellen.																																				
SF2	0x22	0xFF	nicht verwendet																																				
SF3	0x23	0x00	Hier speichert die Grube den letzten Status von R_STAT_FLAG_03, siehe 5.2.5 Hierdurch wird insbesondere ermöglicht, den Status des Blinklichtes beim Einschalten der Grube wieder herzustellen.																																				
AP	0x30	0x18	Anzahl Positionen																																				
PG1	0x31	0x00	invertieren des Byte STAT_FLAG_02 (5.2.4) (d.h. XOR-Verknüpfung von STAT_FLAG_02 und PG1) ermöglicht u.a. andere Polarität für Rückmelde-Eingänge																																				
PG2	0x32	0x01	<table border="1"> <thead> <tr> <th>Bit</th> <th>Bezeichnung</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>#7</td> <td>INV_DIR_SW (neu V0.40)</td> <td>DIR normal</td> <td>DIR (alter Schalter) inv.</td> </tr> <tr> <td>#6</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#5</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#4</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#3</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#2</td> <td>F_NO_ABORT</td> <td>Abbruch durch STEP möglich</td> <td>kein Abbruch durch STEP</td> </tr> <tr> <td>#1</td> <td>TURN_ON_ZERO (nur wenn DIP3)</td> <td>kein TURN, wenn START = ZIEL</td> <td>TURN, wenn START = Ziel</td> </tr> <tr> <td>#0</td> <td>USE_RELAIS (Kehrschleifenrelais)</td> <td>nicht ansteuern</td> <td>verwenden</td> </tr> </tbody> </table>	Bit	Bezeichnung	0	1	#7	INV_DIR_SW (neu V0.40)	DIR normal	DIR (alter Schalter) inv.	#6				#5				#4				#3				#2	F_NO_ABORT	Abbruch durch STEP möglich	kein Abbruch durch STEP	#1	TURN_ON_ZERO (nur wenn DIP3)	kein TURN, wenn START = ZIEL	TURN, wenn START = Ziel	#0	USE_RELAIS (Kehrschleifenrelais)	nicht ansteuern	verwenden
Bit	Bezeichnung	0	1																																				
#7	INV_DIR_SW (neu V0.40)	DIR normal	DIR (alter Schalter) inv.																																				
#6																																							
#5																																							
#4																																							
#3																																							
#2	F_NO_ABORT	Abbruch durch STEP möglich	kein Abbruch durch STEP																																				
#1	TURN_ON_ZERO (nur wenn DIP3)	kein TURN, wenn START = ZIEL	TURN, wenn START = Ziel																																				
#0	USE_RELAIS (Kehrschleifenrelais)	nicht ansteuern	verwenden																																				
PG3	0x33	0x00	<table border="1"> <thead> <tr> <th>Bit</th> <th>Bezeichnung</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>#7</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#6</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#5</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#4</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#3</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#2</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#1</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Bezeichnung	0	1	#7				#6				#5				#4				#3				#2				#1				#0			
Bit	Bezeichnung	0	1																																				
#7																																							
#6																																							
#5																																							
#4																																							
#3																																							
#2																																							
#1																																							
#0																																							
DBG	0x80	0x7a (R_AD_SENSOR)	der Inhalt des Registers mit dieser Adresse wird anstelle der Sensor Spannung per RS232 ausgegeben → Anzeige PC in Grafik „Sensor-Spannung“																																				

Name	EE-prom Zelle	Default Wert	Beschreibung
DBG2	0x81	0x7b (R_AD_SENSE)	der Inhalt des Registers mit dieser Adresse wird anstelle der Sense Spannung per RS232 ausgegeben → Anzeige PC in Grafik „Sense-Spannung“
CNT_ACTIVE	0x90	0xaa	Counter ACTIVE Erkennung

5.4 EEprom Belegung - Bühne

Im EEPROM der Grube sind die wesentlichen Parameter des Systems angelegt:

- Position der Bühne
- Lichtstatus
- ...

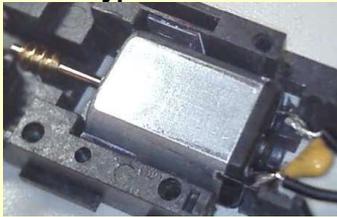
Die Belegung EE ist wie folgt:

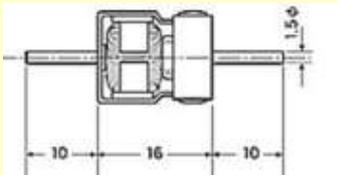
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	PP	V1L	V1R	VM	VC	VD	AP	VOL	PB1	PB2	--	--	FS	FBL	FT	FBR
	0x00	126	126	170	150	10	24	0xff	0x87	0x00	0xff	0xff	65	46	20	46
10	LS	V2L	V2R	T1	T2	T3	T4	T5	TB	TF	TG	TINC	RM_MAX	LR_DLY	SUSI_SLOW	SUSI_FAST
	0x00	50	50	2	20	21	22	55	06	25	128	90	0xff	0x0a	0x01	0x60
20	ID	IL	IR	--	--	--	--	--	--	--	--	--	--	--	--	--
	0x05	0x40	0x40	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff
30	PD	--	--	T5_DBG	DGB	--	--	--	--	--	--	--	--	--	--	--
	0x0a	0xff	0xff	0x22	0x77	0xff	0xff	0xff	0xff							
40	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff
70	--	--	--	--	--	--	--	--	--	--	--	--	--	--	SW-Version	--
	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff	0xff

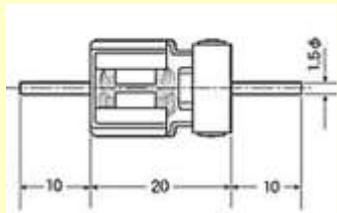
Name	EE-prom Zelle	De-fault Wert	Funktion	Beschreibung																																				
PP	0x00	00	Position	Ist-Position																																				
V1L	0x01	d'126'	SPEED_L_1	Geschwindigkeit Links schnell																																				
V1R	0x02	d'126'	SPEED_R_1	Geschwindigkeit Rechts schnell																																				
VM	0x03	d'170' h'aa'	V_MESS	x 0.02 V = voltage for start of rotation (used in <i>state_move_acc</i>). Dieser Wert definiert eine Totzeit beim Start, sodass mit der Bewertung der weißen Reflektor-Fläche solange gewartet wird, bis der weiße Balken der Start-Position aus dem Blickfeld ist.																																				
VC	0x04	d'150' h'96'	V_COUNT	x 0.02 V = voltage for AD-INPUT (SENSOR), where position is increased/decreased																																				
VD	0x05	d'10'	V_DELTA	x 70 ms = Delta für Ein- / Ausblenden der LEDs (Lichtsignal)																																				
AP	0x06	d'24'	Anzahl_Pos	Anzahl Positionen, wird verwendet, wenn DIP1 = ON																																				
VOL	0x07	d'255'	SUSI Lautstärke	Lautstärke-Wert für SUSI Modul CV902. Wert wird beim schreiben in die EE-Zelle in das SUSI-Modul geschrieben. Lesen erfolgt nur aus der EE-Zelle, nicht aus dem SUSI-Modul!																																				
PB1	0x08	0x87	PARAMETER_S Bühne 1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Bezeichnung</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>#7</td> <td>Sound @ Start</td> <td>Hupe</td> <td>Horn</td> </tr> <tr> <td>#6</td> <td>Sound @ Stop</td> <td>Hupe</td> <td>Horn</td> </tr> <tr> <td>#5</td> <td>LED_DOPPEL</td> <td>Doppel-Blitz</td> <td>Einfach-Blitz</td> </tr> <tr> <td>#3</td> <td>INV_DIR</td> <td>DIR normal</td> <td>DIR inverted</td> </tr> <tr> <td>#2</td> <td>USE_HALL</td> <td>ignore</td> <td>HALL → Pos. = 0</td> </tr> <tr> <td>#1</td> <td>USE_SOUND</td> <td>aus</td> <td>ein</td> </tr> <tr> <td>#0</td> <td>USE_REGELUNG</td> <td>aus</td> <td>ein</td> </tr> </tbody> </table>	Bit	Bezeichnung	0	1	#7	Sound @ Start	Hupe	Horn	#6	Sound @ Stop	Hupe	Horn	#5	LED_DOPPEL	Doppel-Blitz	Einfach-Blitz	#3	INV_DIR	DIR normal	DIR inverted	#2	USE_HALL	ignore	HALL → Pos. = 0	#1	USE_SOUND	aus	ein	#0	USE_REGELUNG	aus	ein				
Bit	Bezeichnung	0	1																																					
#7	Sound @ Start	Hupe	Horn																																					
#6	Sound @ Stop	Hupe	Horn																																					
#5	LED_DOPPEL	Doppel-Blitz	Einfach-Blitz																																					
#3	INV_DIR	DIR normal	DIR inverted																																					
#2	USE_HALL	ignore	HALL → Pos. = 0																																					
#1	USE_SOUND	aus	ein																																					
#0	USE_REGELUNG	aus	ein																																					
PB2	0x09	0x00	PARAMETER_S Bühne 2 (ab V0.29)	<table border="1"> <thead> <tr> <th>Bit</th> <th>Bezeichnung</th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>#7</td> <td>F_RM_USE</td> <td>ignore</td> <td>RM_0x sind als Ausgang geschaltet (ab V0.30)</td> </tr> <tr> <td>#6</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#5</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#4</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#3</td> <td></td> <td></td> <td></td> </tr> <tr> <td>#2</td> <td>F_RM_SOUND_3</td> <td>ignore</td> <td>RM3->HUPE</td> </tr> <tr> <td>#1</td> <td>F_RM_SOUND_2</td> <td>ignore</td> <td>RM2->HUPE</td> </tr> <tr> <td>#0</td> <td>F_RM_SOUND_1</td> <td>ignore</td> <td>RM1->HUPE</td> </tr> </tbody> </table>	Bit	Bezeichnung	0	1	#7	F_RM_USE	ignore	RM_0x sind als Ausgang geschaltet (ab V0.30)	#6				#5				#4				#3				#2	F_RM_SOUND_3	ignore	RM3->HUPE	#1	F_RM_SOUND_2	ignore	RM2->HUPE	#0	F_RM_SOUND_1	ignore	RM1->HUPE
Bit	Bezeichnung	0	1																																					
#7	F_RM_USE	ignore	RM_0x sind als Ausgang geschaltet (ab V0.30)																																					
#6																																								
#5																																								
#4																																								
#3																																								
#2	F_RM_SOUND_3	ignore	RM3->HUPE																																					
#1	F_RM_SOUND_2	ignore	RM2->HUPE																																					
#0	F_RM_SOUND_1	ignore	RM1->HUPE																																					
FS	0x0c	d'65'	filter_s	Filter for Minimum search: V gets Smaller (siehe Kapitel 3.5)																																				
FBL	0x0d	d'46'	filter_b links	Filter for Minimum search: V gets Bigger (siehe Kapitel 3.5)																																				
FBR	0x0f	d'46'	filter_b rechts																																					
FT	0x0e	d'20'	filter_t	Filter for threshold Dieser Wert definiert eine Totzeit beim Start, sodass mit der Bewertung der weißen Reflektor-Fläche solange gewartet wird, bis der weiße Balken der Start-Position aus dem Blickfeld ist.																																				
ST	0x10	d'00'	Status Flags speichern des letzten Status für die Wiederherstellung bei Power-ON	<table border="1"> <thead> <tr> <th>#7</th> <th>#6</th> <th>#5</th> <th>#4</th> <th>#3</th> <th>#2</th> <th>#1</th> <th>#0</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Blinklicht: 0 = aus 1 = ein</td> <td>Licht: 0 = aus 1 = ein</td> </tr> </tbody> </table>	#7	#6	#5	#4	#3	#2	#1	#0							Blinklicht: 0 = aus 1 = ein	Licht: 0 = aus 1 = ein																				
#7	#6	#5	#4	#3	#2	#1	#0																																	
						Blinklicht: 0 = aus 1 = ein	Licht: 0 = aus 1 = ein																																	
V2L	0x11	d'50'	SPEED_L_2	Geschwindigkeit Links langsam																																				
V2R	0x12	d'50'	SPEED_R_2	Geschwindigkeit Rechts langsam																																				
T1	0x13	d'02'	T_DUNKEL	x 70 ms = Zeit: Länge der Dunkelphase f. Licht-Signal																																				
T2	0x14	d'20'	T_HORN_START	x 70 ms = Zeit: Länge des Horn bei Start																																				

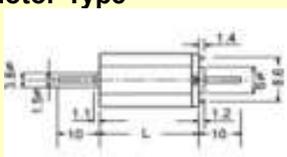
Name	EE-prom Zelle	De-fault Wert	Funktion	Beschreibung
T3	0x15	d'21'	T_HORN_STOP	x 70 ms = Zeit: Länge des Horn bei Stopp
T4	0x16	d'22'	T_WAIT_START	x 70 ms = Zeit: Länge der Wartezeit zwischen Signal Rot und Bühne Start
T5	0x17	d'55'	T_WAIT_STOP	x 70 ms = Zeit: Länge der Wartezeit zwischen Bühne Halt und Signal Rot ausblenden
TB	0x18	d'06'	T_BOOST	Zeit: Länge des Boost-Impulses.
TF	0x19	d'25'	T_BLOCK_1	Zeit bis zur Erkennung Blockierung bei Bühnen-Start → „E5“
TG	0x1a	d'128'	T_BLOCK_2	Zeit bis zur Erkennung Blockierung bei Bühnen-Fahrt → „E8“
TINC	0x1b	d'90'	R_T_INC	Anfahrverzögerung (weiches Anfahren), je kleiner, desto schneller wird beschleunigt
RM_M AX	0x1c	d'255'	R_RM_MAX	Max-Wert für die Rückmelder-Counter. Kleinerer Wert = schnellere RM Erkennung. Zulässig = 2 (schnelle erkennung) bis 255 (langsame Erkennung)
LR DLY	0x1d	d'10'	R_LR_DELAY = „Motor Trägheit“	Counter, der beim Start der Sensitiv-Phase mit diesem Wert aus dem EE geladen wird. Die Strombewertung erfolgt dann erst, wenn der Counter = ZERO. Dadurch können Motoren, die besonders träge reagieren (d.h. nach dem Umschalten auf die Sensitiv-Phase mit der Nominal-PWM sinkt der Strom langsamer auf den Nominal-Wert) verwendet werden. Ist dieser Wert zu klein, so wird der Motor bei der Lastregelung schwingen. In der Sensitiv-Phase wird dann nicht der Nominal-Strom bewertet, da dieser noch nicht erreicht wurde. In der Folge würde dann die PWM hochgedreht werden, da der falsche (zu hohe) Strom bewertet wurde.
SUSI_SLOW	0x1e	0x01	SUSI Fahrstufe SLOW	Diese Werte definieren, welche Lok Fahrstufe an das SUSI Modul gesendet wird. Zulässige Werte für Fahrstufe: 1 bis 127 (7 Bit)
SUSI_FAST	0x1f	0x60	SUSI Fahrstufe FAST	
ID	0x20	d'5'	STROM_DELTA	Lastregelung: Delta Stromanstieg für PWM+ (PWM_DELTA)
IL	0x21	d'64'	STROM_L	Lastregelung: Strom bei minimaler Geschwindigkeit Links
IR	0x22	d'64'	STROM_R	Lastregelung: Strom bei minimaler Geschwindigkeit Rechts
PD	0x30	d'10'	PWM_DELTA = „PWM Schwellwert“	Neu V0.35 Erhöhung PWM, wenn Strom STROM_DELTA überschreitet Bis V0.35 wurde dieser Wert berechnet $(127 - R_PWM (=V \times L)) / 4$ es wurde also der verbleibende PWM Wertebereich (127 = PWM_MAX, $V \times L$ = PWM in Langsamfahrt) in 4 Stufen unterteilt, d.h. bei Überschreitung des Stromes um 4x PWM_DELTA war die PWM bei MAX = 127.
T5_DBG	0x33	0x22	T_WAIT_STOP @ F_DEBUG	Wartezeit für den Halt an jeder Position im Debug-Modus (Halt an jeder Position) (neu V0.35)
DBG	0x34	0x77	ADR_DEBUG	der Inhalt des Registers mit dieser Adresse wird anstelle der Sensor Spannung per RS232 ausgegeben, wenn der DEBUG-Mode aktiv ist → Anzeige PC im Graf „Sensor-Spannung“
SW-Version	0x7e.. 0x7f	Ver-sion		Software-Version, schreiben auf diese Zellen wird ab V0.23 abgeblockt

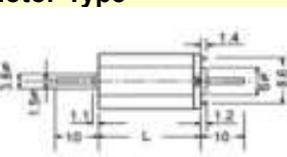
5.5 Parameter Lastregelung

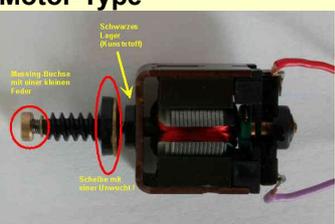
Motor-Type 	Tillig ab 2017 (geschlossen), ohne 100nF, Leerlauf 100nF Kondensator muss entfernt werden, sonst ist keine Stromerhöhung bei Blockierung messbar 4.4 mH 21 Ohm
Geschwindigkeit Langsam	5 / 22
Strom	17-18 / 29-31
Strom Schwellwert	2
PWM Schwellwert	10-20
Motor Trägheit	15

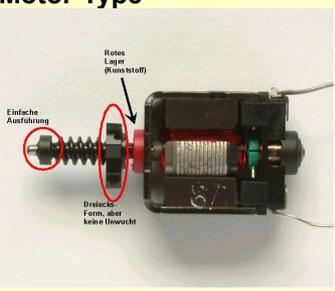
Motor-Type 	Mashima M16k (9.3 x 13.1 x 16 mm offen), Leerlauf schwer einzustellen, Motorstrom bei Blockade sinkt wieder deutlich ab xx Ohm xx mH
Geschwindigkeit Langsam	17
Strom	18-19
Strom Schwellwert	
PWM Schwellwert	
Motor Trägheit	

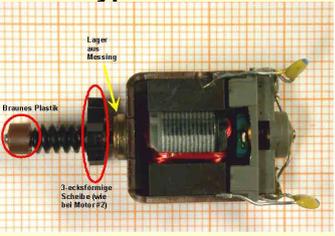
Motor-Type 	Mashima, (9.3 x 13.1 x 20 mm offen), Leerlauf xx Ohm xx mH
Geschwindigkeit Langsam	17
Strom	13
Strom Schwellwert	2
PWM Schwellwert	3
Motor Trägheit	10

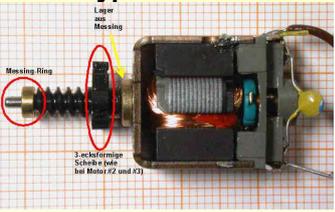
Motor-Type 	Mashima MHK1020, (10 x 12 x 20 mm geschlossen), Leerlauf xx Ohm xx mH
Geschwindigkeit Langsam	17
Strom	18-19
Strom Schwellwert	5
PWM Schwellwert	4
Motor Trägheit	10

Motor-Type 	Mashima MHK1015, (10 x 12 x 15 mm geschlossen), Leerlauf xx Ohm xx mH
Geschwindigkeit Langsam	27
Strom	29
Strom Schwellwert	8
PWM Schwellwert	5
Motor Trägheit	10

Motor-Type 	Fleischmann #1 112 Ohm 15 mH Motor mit hoher Kraft und hohem Strom
Geschwindigkeit Langsam	
Strom	
Strom Schwellwert	
PWM Schwellwert	
Motor Trägheit	

Motor-Type 	Fleischmann #2 90 Ohm (165mA @ 15V & Blockierung) 15 mH 35mA @ 15V & Leerlauf
Geschwindigkeit Langsam	51 (Leerlauf) / 52 (in Bühne)
Strom	57-61 (Leerlauf) / 59-62 (in Bühne)
Strom Schwellwert	10
PWM Schwellwert	20
Motor Trägheit	20

Motor-Type 	Fleischmann #3 xx Ohm xx mH
Geschwindigkeit Langsam	
Strom	
Strom Schwellwert	
PWM Schwellwert	
Motor Trägheit	

Motor-Type 	Fleischmann #4 220 Ohm 40-50 mH
Geschwindigkeit Langsam	32-67 (Leerlauf) / 65 (in Bühne) /
Strom	15-43 (Leerlauf) / 45 (in Bühne)
Strom Schwellwert	3
PWM Schwellwert	18
Motor Trägheit	20

Motor-Type	SB-Motor x Ohm x mH
Geschwindigkeit Langsam	21 (in Bühne) /
Strom	29 – 31 (in Bühne)
Strom Schwellwert	2
PWM Schwellwert	2
Motor Trägheit	1