

**Heinzmann GmbH & Co. KG**  
**Motor- & Turbinenmanagement**

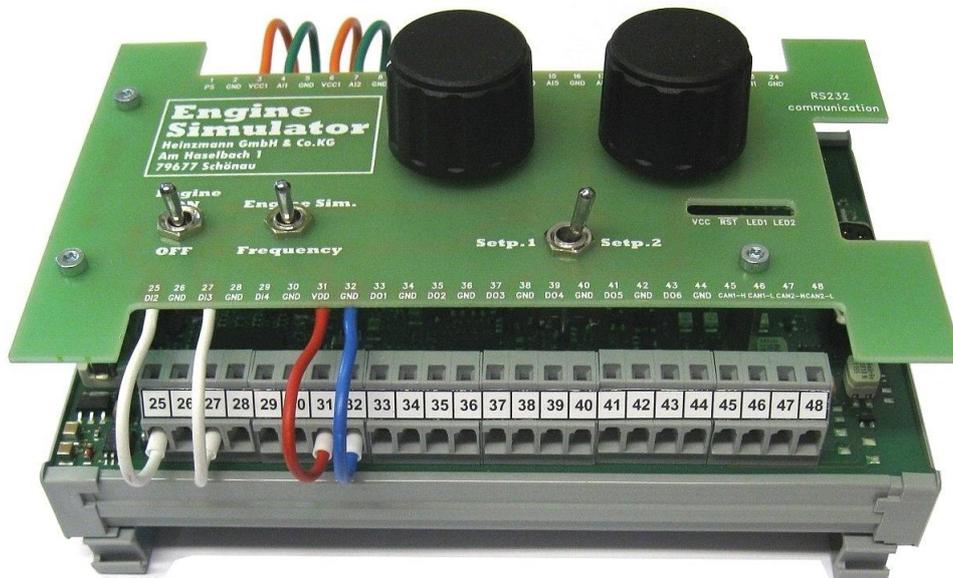
Am Haselbach 1  
D-79677 Schönau

Telefon +49 (0)7673 8208-0  
Telefax +49 (0)7673 8208-188  
E-Mail [info@heinzmann.com](mailto:info@heinzmann.com)  
[www.heinzmann.com](http://www.heinzmann.com)

USt-IdNr.: DE145551926

**HEINZMANN®**  
**Motor- & Turbinenmanagement**

**Motorsimulator**



Copyright 2012 Heinzmann GmbH & Co. KG. Alle Rechte vorbehalten.  
Diese Druckschrift darf nicht vervielfältigt oder an Dritte weitergegeben werden.



   	<p><b>Vor Installation, Inbetriebnahme und Wartung müssen die entsprechenden Handbücher im Ganzen durchgelesen werden.</b></p> <p>Alle Anweisungen, welche die Anlage und die Sicherheit betreffen, müssen unbedingt befolgt werden. Nichtbefolgen der Anweisung kann zu Personen- und/oder Sachschäden führen.</p> <p><b>HEINZMANN</b> übernimmt keine Haftung für Schäden, die durch Nichtbefolgung von Anweisungen entstehen.</p> <p>Unabhängige Tests und Überprüfungen sind von besonderer Bedeutung bei allen Anwendungen, bei denen ein fehlerhaftes Funktionieren zu Personen- oder Sachschäden führen kann.</p> <p>Alle Beispiele und Daten, sowie alle übrigen Informationen in diesem Handbuch dienen ausschließlich dem Zweck der Unterweisung und dürfen nicht für spezielle Anwendung eingesetzt werden, ohne dass der Anwender unabhängige Tests und Überprüfungen durchgeführt hat.</p> <p><b>HEINZMANN</b> übernimmt keine Garantie, weder ausdrücklich noch stillschweigend, dass die Beispiele, Daten oder sonstigen Informationen in diesem Handbuch fehlerfrei sind, Industriestandards entsprechen oder den Bedürfnissen irgendeiner besonderen Anwendung genügen.</p>
 	<p><b>Um Personenschäden und Schäden an Anlagen zu vermeiden, müssen folgende Überwachungs- und Schutzsysteme vorhanden sein:</b></p> <ul style="list-style-type: none"> <li>– vom Drehzahlregler unabhängiger Überdrehzahlschutz</li> </ul> <p><b>HEINZMANN</b> übernimmt keine Haftung für Schäden, die durch fehlenden oder unzureichenden Überdrehzahlschutz entstehen.</p> <ul style="list-style-type: none"> <li>– Übertemperaturschutz</li> </ul> <p><b>Bei Generatoranlagen zusätzlich:</b></p> <ul style="list-style-type: none"> <li>– Überstromschutz</li> <li>– Schutz vor Fehlsynchronisation bei zu großer Frequenz-, Spannungs- oder Phasendifferenz</li> <li>– Rückleistungsschutz</li> </ul> <p>Ursachen für Überdrehzahl können sein:</p> <ul style="list-style-type: none"> <li>– Ausfall des Stellgerätes, des Kontrollgerätes oder dessen Zusatzgeräten</li> <li>– schwergängiges oder klemmendes Gestänge</li> </ul>
 	<p><b>Vor einer Installation muss folgendes unbedingt beachtet werden:</b></p> <ul style="list-style-type: none"> <li>– Vor Eingriffen in die Anlage diese immer spannungsfrei schalten!</li> <li>– Nur Kabelabschirmung und Stromversorgungsanschlüsse verwenden, die der Europäischen Richtlinie bezüglich EMV entsprechen.</li> </ul> <p>Überprüfung der Funktion aller vorhandenen Schutz- und Überwachungssysteme.</p>

 <p>HINWEIS</p>	<p><b>Bei elektronisch geregelter Einspritzung (MVC) ist folgendes zusätzlich zu beachten:</b></p> <ul style="list-style-type: none"> <li>– Bei Common Rail Systemen muss für jede Injektorleitung ein separater mechanischer Durchflussbegrenzer vorhanden sein.</li> <li>– Bei <b>Pumpe-Leitung-Düse- (PLD)-</b> und <b>Pumpe-Düse- (PDE)-</b> Systemen darf die Treibstofffreigabe erst durch die Steuerkolbenbewegung des Magnetventils ermöglicht werden. Dadurch wird bei Verharren des Steuerkolbens die Treibstoffzuführung zur Einspritzdüse verhindert.</li> </ul>
 <p>⚠️ WARNUNG</p>	<p>Sobald das Stellgerät mit einer Spannung versorgt wird, kann es jederzeit selbstständig die Reglerausgangswelle ansteuern. Der Bereich der Reglerwelle bzw. des Regelgestänges ist deshalb vor unberechtigtem Zutritt zu verschließen.</p>
	<p><b>HEINZMANN</b> lehnt ausdrücklich die stillschweigende Garantie für die Marktfähigkeit oder die Eignung für einen speziellen Zweck ab, auch für den Fall, dass <b>HEINZMANN</b> auf einen speziellen Zweck aufmerksam gemacht wurde oder dass im Handbuch auf einen speziellen Zweck hingewiesen wird.</p>
	<p><b>HEINZMANN</b> lehnt jede Haftung für mittelbare und unmittelbare Schäden sowie für Begleit- und Folgeschäden ab, die sich aus irgendeiner Verwendung der in diesem Handbuch enthaltenen Beispiele, Daten oder sonstigen Informationen ergeben.</p>
	<p><b>HEINZMANN</b> übernimmt keine Gewähr für die Konzeption und Planung der technischen Gesamtanlage. Dies ist Sache des Betreibers bzw. deren Planer und Fachingenieure. Es liegt auch in deren Verantwortungsbereich zu überprüfen, ob die Leistungen unserer Geräte dem angestrebten Zweck genügen. Der Betreiber ist auch für eine ordnungsgemäße Inbetriebnahme der Gesamtanlage verantwortlich.</p>

# Inhaltsverzeichnis

Seite

<b>1 Allgemeines .....</b>	<b>1</b>
<b>2 Standardsimulator .....</b>	<b>3</b>
2.1 Allgemeines .....	3
2.2 Signalerzeugung .....	3
2.2.1 Konfiguration des Pickup1-Signals .....	3
2.2.2 Konfiguration des Pickup2-Signals .....	4
2.2.3 Konfiguration des CamIndex-Signals .....	4
2.2.4 Erzeugung von Störsignalen auf der Kurbelwelle .....	5
2.2.5 Erzeugung von Störsignalen auf der Nockenwelle .....	6
2.3 Frequenzsimulation .....	6
2.4 Motorsimulation .....	7
2.5 Drehzahlverlauf .....	8
2.5.1 Abspielen von DcDesk-Recorddateien .....	8
2.6 Mehrfachlückenrad .....	9
2.7 Misfire-Signalerzeugung .....	10
2.8 Signalmustererzeugung .....	11
2.9 Ausgangssignalerzeugung .....	12
<b>3 Pinbelegung .....</b>	<b>13</b>



## 1 Allgemeines

Die Motorsimulator-Hardware besitzt folgenden Ein- und Ausgänge:

- 6 x Frequenz-/PWM-/Digital-Ausgänge
- 4 x Frequenz-/PWM-/Digital-Eingänge
- 6 x Analogeingänge
  - 4 x 0..5 V
  - 2 x 0..25 mA
- 2 x Analogausgänge
  - 1 x 0..5 V
  - 1 x 0..25 mA
- 2 x CAN-BUS
- 1 x RS232
- 48-poliger Erweiterungsstecker

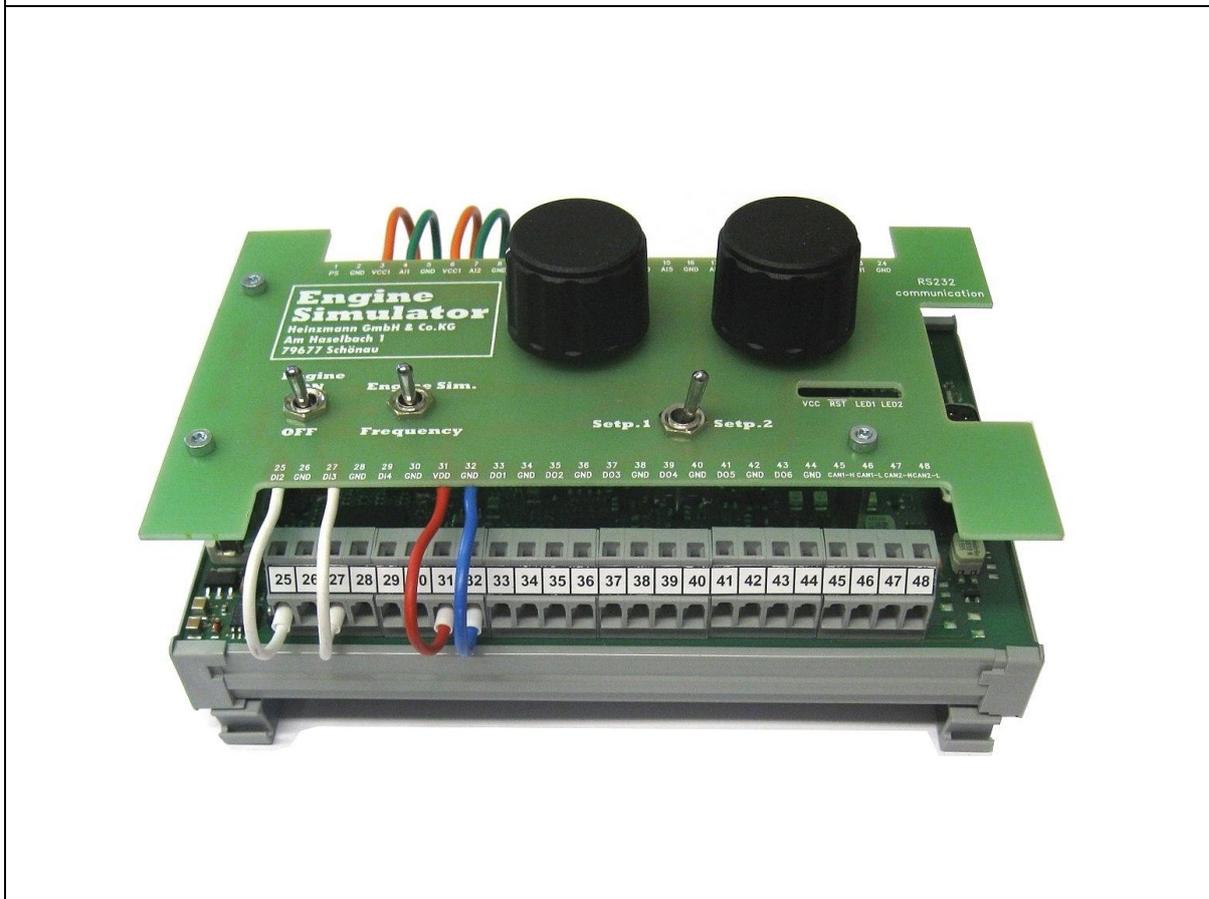
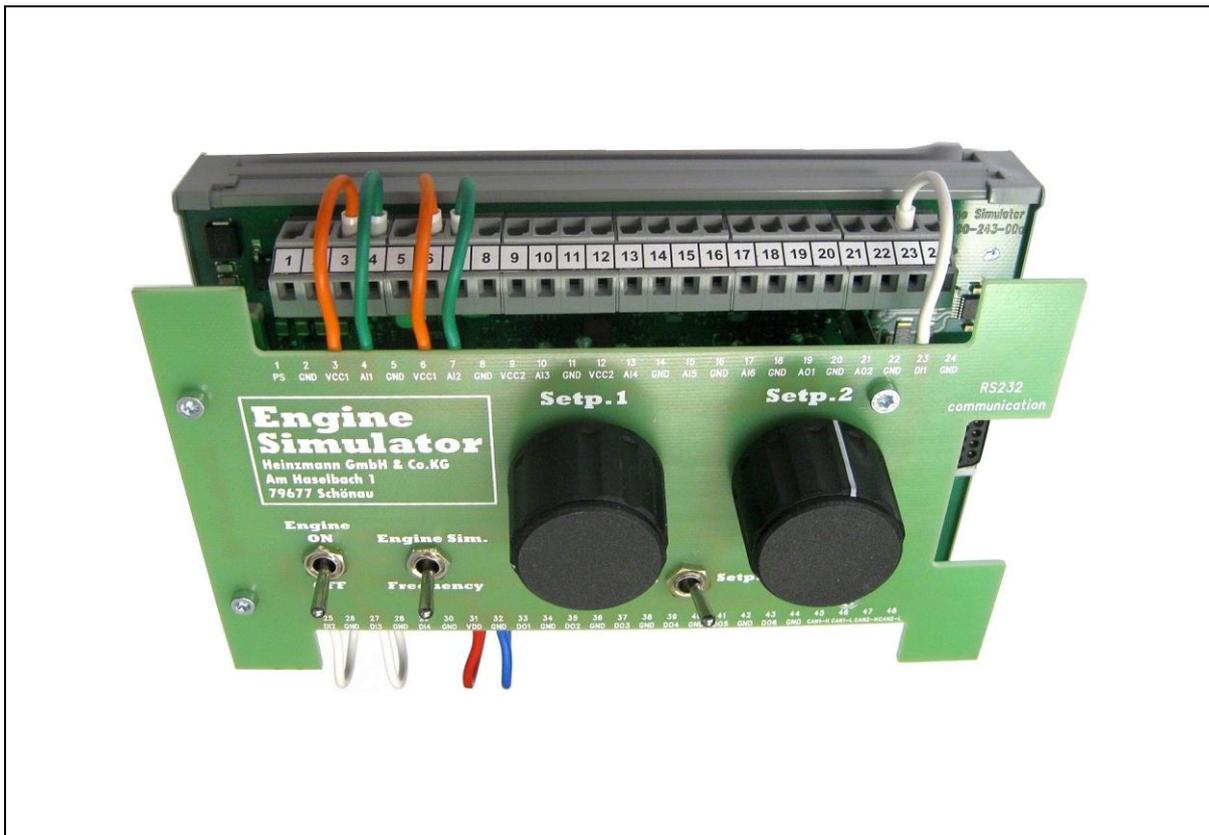


Abb. 1 Ansichten Motorsimulator

## 2 Standardsimulator

### 2.1 Allgemeines

Der Motorsimulator kann als Frequenz- oder Motorsimulator verwendet werden, d.h. auch im drehzahlgeregelten Betrieb. Er erzeugt über drei Frequenzgänge die Signale zweier Kurbelwellen- sowie eines Nockenwellensensors. Die Konfiguration dieser Signale ist dabei frei wählbar.

Zusätzlich kann auch die Startbedingung festgelegt werden, damit ein Motorstart immer an einer identischen Stelle erfolgt. Dies ist dann notwendig, wenn Untersuchungen zur Synchronisierung gemacht werden.

Die Mindestfrequenz des Simulators ist 5 Hz. Die Höchsthfrequenz ist auf 10000 Hz eingestellt. Im Prinzip ist es möglich, auch noch höhere Frequenzen zu erzeugen. Auf Wunsch kann dies implementiert werden.

### 2.2 Signalerzeugung

Die Signale von PickUp1 und PickUp2 werden unabhängig voneinander erzeugt, d.h. es gibt keinen Zusammenhang zwischen diesen Signalen. Das bedeutet, dass unter Umständen sich die Kurbelwellenlückenposition von PickUp2 relativ zu der von PickUp1 verschieben kann. Sofern für PickUp1 und PickUp2 dieselbe Zähnezahle eingetragen ist, ist diese Verschiebung kaum oder gar nicht zu sehen.

Das Signal der Nockenwelle wird dagegen direkt von PickUp1 erzeugt, d.h. dieses Signal ist exakt synchron zum PickUp1-Signal. Die Signaltbreite des Nockenwellensignals entspricht einem PickUp1-Zahn.

Für die Signalerzeugung wird ein interner Zahnzähler verwendet. Dieser Zahnzähler läuft von 1 bis zur doppelten Zähnezahle (bei 60 Zähnen also von 1..120). Die Lückenposition von PickUp1 wird immer bei Zahn Nr. 1 und der parametrisierten Zähnezahle erzeugt. Die Lückenposition von PickUp2 und die Nockenwellenpositionen werden relativ zur Zahnnummer 1 von PickUp1 eingetragen.

Die Frequenzerzeugung bezieht sich immer auf PickUp1. Die Signale von PickUp2 und Cam-Index werden ausgehend von dieser Frequenz erzeugt.

#### 2.2.1 Konfiguration des PickUp1-Signals

- |                      |                                   |
|----------------------|-----------------------------------|
| 1 TeethPickUp1       | Zähnezahle von PickUp1            |
| 2 NumberOfGapPickUp1 | Anzahl der Lücken                 |
| 3 ToothRatioPickUp1  | Tastverhältnis für PickUp1-Signal |

4	StartToothNo	Zahnnummer, mit welcher die Signalerzeugung bei Motorstart beginnt (siehe auch 4110)
5	FrequencyRampPickUp1	Frequenz-Änderungsgeschwindigkeit
4001	PickUp1GapHighOrLow	Polarität des Signals in der Lücke (wird nur bei Drehzahl null übernommen)
4005	PU1FrequencyRampOn	Aktivierung der Frequenzrampe
4110	ResetToothNoAtStop	Zahnzähler bei Motorstop rücksetzen. Damit beginnt jeder Motorstart an derselben Stelle

### 2.2.2 Konfiguration des PickUp2-Signals

11	TeethPickUp2	Zähnezahl von PickUp2
12	NumberOfGapPickUp2	Anzahl der Lücken
13	ToothRatioPickUp2	Tastverhältnis für PickUp2-Signal
14	TeethOffsetForGapPU2	Zahn-Offset der Lücke von PickUp2 relativ zum Zahnzähler von PickUp1 (wird nur bei Drehzahl null übernommen)
15	FrequencyRampPickUp2	Frequenz-Änderungsgeschwindigkeit
4011	PickUp2GapHighOrLow	Polarität des Signals in der Lücke (wird nur bei Drehzahl null übernommen)
4015	PU1FrequencyRampOn	Aktivierung der Frequenzrampe

### 2.2.3 Konfiguration des CamIndex-Signals

21	CamSimulationMode	Auswahl des Nockenwellensignals 0 = deaktiviert 1 = Einzelimpuls 2 = frei konfigurierbar 3 = Triggerscheibe mit 4+1 Zähnen 4 = Triggerscheibe mit 6+1 Zähnen 5 = Triggerscheibe mit 8+1 Zähnen
22	NoOfCamPositions	Anzahl der Nockenwellensignale bei freier Konfiguration des Signals
24	CamOffset	Zahn-Offset des ersten Nockenwellensignals relativ zum Zahnzähler von PickUp1
2250	CamPositions()	Anzeige der Zahnnummern, bei welchen ein Nockenwellenimpuls erzeugt wird (bezogen auf den Zahnzähler von PickUp1)

4021 CamIndexHighOrLow	Polarität des Signals (wird nur bei Drehzahl null übernommen)
6000 CamPositions()	Zahnnummer der Nockenwellensignale bei freier Konfiguration (bezogen auf den Zahnzähler von PickUp1)

### 2.2.3.1 Berechnung des CamOffsets

In Parameter 24 CamOffset ist der Zahn-Offset des ersten Nockenwellensignals relativ zum Zahnzähler von PickUp1 einzutragen. Dieser Abstand ist derjenige, der vom MVC-Steuergerät als Synchronisierkriterium verwendet wird. Der CamOffset berechnet sich mit den MVC-Steuergeräteparametern 3 SensorToGapPickUp1 und 5 SensorToCamIndex wie folgt:

für  $3 \text{ SensorToGapPickUp1} < 5 \text{ SensorToCamIndex}$  gilt:

$$24 \text{ CamOffset} = \frac{(720^\circ + 3 \text{ SensorToGapPickUp1} - 5 \text{ SensorToCamIndex})}{360^\circ * 1 \text{ TeethPickUp1}}$$

für  $3 \text{ SensorToGapPickUp1} > 5 \text{ SensorToCamIndex}$  gilt:

$$24 \text{ CamOffset} = \frac{(3 \text{ SensorToGapPickUp1} - 5 \text{ SensorToCamIndex})}{360^\circ * 1 \text{ TeethPickUp1}}$$

### 2.2.4 Erzeugung von Störsignalen auf der Kurbelwelle

Ein Signalausfall von PickUp1 oder PickUp2 kann simuliert werden, indem das Zahnverhältnis 3 ToothRatioPickUp1 bzw. 13 ToothRatioPickUp2 auf 0% oder 100% gesetzt wird.

Zur Erzeugung eines Synchronisierfehlers kann manuell kurz die Anzahl der Lücken 2 NumberOfGapPickUp1 bzw. 12 NumberOfGapPickUp2 auf einen falschen Wert geändert und sofort wieder der korrekte Wert eingestellt werden.

Dies erzeugt unter Umständen jedoch ein über mehrere Umdrehungen anstehenden Synchronisierfehler. Deshalb gibt es auch die Möglichkeit, einen einmaligen Synchronisierfehler zu erzeugen. Hierbei gibt es die beiden Varianten: *Lücke erkannt, jedoch Zahn erwartet* (gap detected, but tooth expected) und *Zahn erkannt, jedoch Lücke erwartet* (tooth detected, but gap expected).

Bei Aktivierung von 4002 PU1SyncToothInGap oder 4012 PU2SyncToothInGap wird die Lücke einmalig als Zahn ausgegeben. Dieser Synchronisierfehler kann durch erneutes Aktivieren des Parameters wiederholt werden.

Anhand der Parameter 4003 PU1SyncErrGapInTooth oder 4013 PU2SyncErrGapInTooth kann im Gegensatz dazu einmalig ein Zahn als Lücke ausgegeben werden. Dieser Synchronisierfehler kann durch erneutes Aktivieren des Parameters wiederholt werden.

### 2.2.5 Erzeugung von Störsignalen auf der Nockenwelle

Ein Signalausfall kann erzeugt werden, in dem Parameter 21 CamSimulationMode auf 0 (disabled) gesetzt wird.

Zur Erzeugung eines Störsignals kann in Parameter 21 CamSimulationMode eine andere Konfiguration eingetragen werden, als das Steuergerät erwartet.

Sofern der frei wählbare Modus aktiviert ist (21 CamSimulationMode = 2), kann dies auch durch eine Variation der Anzahl der zu erzeugenden Impulse in Parameter 22 NoOfCamPositions erfolgen.

Dies erzeugt unter Umständen jedoch ein über mehrere Umdrehungen anstehenden Synchronisierfehler. Deshalb gibt es auch die Möglichkeit, über Parameter 4022 CamIndexSyncErrorOn einen einmaligen Synchronisierfehler zu erzeugen. Bei Aktivierung wird einmalig ein Impuls unterdrückt. Der zu unterdrückende Impuls kann in Parameter 25 CamSyncErrorIndex ausgewählt werden. Hier ist der Index des Feldes 2250 CamPositions() einzutragen, welches die Zahnnummern anzeigt, bei denen ein Nockenwellenimpuls erzeugt wird.

Ein weiterer Synchronisierfehler kann durch erneutes Aktivieren des Parameters 4022 erzeugt werden.

## 2.3 Frequenzsimulation

Für eine Frequenzsimulation muss der Schalter 2812 SwSimEngineOrFreq auf ‚Freq‘ stehen. Die Frequenzsimulation kann über den Schalter 2810 SwEngineOn aktiviert bzw. deaktiviert werden.

Die Frequenzvorgabe kann entweder direkt über DcDesk 2000 oder über zwei Sensorwerte (normalerweise Potentiometer) vorgegeben werden. Die Auswahl erfolgt über 4100 FrequencyPotOrPC. Falls der Parameter auf ‚PC‘ steht, erfolgt die Frequenzvorgabe über Parameter 100 FrequencySetpPC.

Bei der Vorgabe über Sensorwerte werden die beiden Sensoren 2900 Setpoint1 bzw. 2901 Setpoint2 verwendet. Die Auswahl erfolgt über den Schalter 2811 SwSetpoint2Or1.

Die aus diesen Sensoren erzeugte Frequenz wird in Parameter 2101 FrequencySetp1 bzw. 2102 FrequencySetp2 angezeigt.

Mit Parameter 5 FrequencyRampPickUp1 bzw. 15 FrequencyRampPickUp2 wird die Änderungsgeschwindigkeit der Frequenzvorgabe bestimmt.

## 2.4 Motorsimulation

Für eine Motorsimulation muss der Schalter 2812 SwSimEngineOrFreq auf ‚Engine‘ stehen. Die Motorsimulation kann über den Schalter 2810 SwEngineOn aktiviert bzw. deaktiviert werden.

Für die Motorsimulation ist eine Information über das aktuelle Drehmoment notwendig. Dieses wird in 2902 TorqueInput angeschlossen.

Die Soll Drehmomentvorgabe kann entweder direkt über DcDesk 2000 oder über zwei Sensorwerte (normalerweise Potentiometer) vorgegeben werden. Die Auswahl erfolgt über 4150 TorquePotOrPC. Falls der Parameter auf ‚PC‘ steht, erfolgt die Frequenzvorgabe über Parameter 150 TorqueSetpPC.

Bei der Vorgabe über Sensorwerte werden die beiden Sensoren 2900 Setpoint1 bzw. 2901 Setpoint2 verwendet. Die Auswahl erfolgt über den Schalter 2811 SwSetpoint2Or1.

Das aus diesen Sensoren erzeugte Drehmoment wird in Parameter 2201 TorqueSetp1 bzw. 2202 TorqueSetp2 angezeigt.

Der Drehmomentsollwert, mit welchem der Simulator arbeitet, wird in 2051 EngineTorqueSetp angezeigt.

Mit Parameter 5 FrequencyRampPickUp1 bzw. 15 FrequencyRampPickUp2 wird die Änderungsgeschwindigkeit der Frequenzvorgabe bestimmt.

In Parameter 200 MomentOfInertia wird das Trägheitsmoment des Motors eingegeben. Kleinere Werte ergeben einen schnelleren Motor.

Über die Parameter 6100..6250 werden Kennlinien für das minimale und maximale Drehmoment des Motors eingegeben. Oberhalb des maximalen Drehmomentes wird kein Drehmoment mehr erzeugt, d.h. wenn mehr Drehmoment verlangt wird, ist der Motor überlastet und die Drehzahl bricht ein.

Das prozentuale, aktuelle Drehmoment des Motors bestimmt sich aus 2902 TorqueInput, bezogen auf diese Kennlinien. Dieses Drehmoment wird in 2050 EngineTorque angezeigt.

In Parameter 250 StarterSpeed wird die Anlasser-Drehzahl eingetragen. Wenn bei stehendem Motor der Schalter 2810 SwEngineOn auf ‚On‘ geht, wird diese Anlasserdrehzahl erzeugt. Sofern der Drehmoment-Sollwert kleiner als das für den Motorstart benötigte Drehmoment (aus der Kennlinie) ist, startet der Motor.

Die Anlasserdrehzahl wird mit einer sinusförmigen Schwingung überlagert, um das Anlassergeräusch zu simulieren.

## 2.5 Drehzahlungsverlauf

Anstatt der Simulation eines Motors kann das Drehzahlungsverlauf auch anhand einer reproduzierbaren und parametrierbaren Vorgabe erzeugt werden.

Dazu muss der Schalter 2812 SwSimEngineOrFreq auf ‚Engine‘ stehen. Außerdem muss bei der Funktion 4101 SpeedCurvOrEngineSim ‚SpeedCurve‘ ausgewählt sein.

Das Drehzahlungsverlauf kann über den Schalter 2810 SwEngineOn aktiviert bzw. deaktiviert werden.

Das Drehzahlungsverlauf muss in folgende Parameter eingetragen werden:

7000 SpeedSim:t()	Zeit für Drehzahlungsverlauf
7300 SpeedSim:n()	zugehörige Drehzahlwerte

Sobald der Schalteingang 2810 SwEngineOn aktiviert wird, beginnt die Erzeugung des Drehzahlungsverlaufs aus dem parametrisierten Verlauf. Wenn das Signal komplett abgespielt wurde, wird der letzte Drehzahlungsverlaufwert beibehalten.

Eine Deaktivierung mit anschließender Aktivierung von 2810 SwEngineOn erzeugt erneut denselben Signalverlauf.



*Die Frequenzrampe gilt auch bei der Erzeugung dieses Signals. Soll exakt der parametrisierte Verlauf erzeugt werden, empfiehlt es sich, die Frequenzrampe zu deaktivieren.*

### 2.5.1 Abspielen von DcDesk-Recorddateien

Mit dieser Funktion können somit auch DcDesk-Recorddateien abgespielt werden. Dazu speichert man zunächst die Parameter der beiden Kennlinien als hzm-Datei ab und öffnet dann sowohl diese hzm-Datei als auch die gewünschte rec-Datei in CodeWright. Dort kann man dann die zwei Spalten für die Zeitachse und den Drehzahlwert nacheinander markieren und in die hzm-Datei übertragen.

Eventuell empfiehlt sich ein Zwischenschritt über Excel, um zum einen ein Offset der Zeitachse zu kompensieren und zum anderen die Drehzahlwerte zu runden.

## 2.6 Mehrfachlückenrad

Einige Spezialabwendungen benötigen ein Mehrfachlückenrad. Dabei sind auf dem Messrad mehrere Lücken angebracht, wobei die Anzahl der Zähne zwischen den einzelnen Lücken jeweils unterschiedliche ist. Damit ist eine sehr schnelle Synchronisierung möglich, da nur die Anzahl der Zähne zwischen zwei Lücken gezählt werden muss.

Der Motorsimulator unterstützt die Erzeugung eines solchen Signals auf PickUp1 und PickUp2. Beide PickUps erzeugen das Signal desselben Zahnrades (falls benötigt, könnten auch unterschiedliche Räder implementiert werden).



*Bei Mehrfachlückenrad erzeugt die Verschiebung des PickUp2-Signals über Parameter 14 TeethOffsetForGapPU2 bei Motorstart ein falsches PU2-Signal und auch die Verschiebung ist nicht korrekt – deshalb Parameter 14 auf 0 lassen (Punkt für die ToDo-Liste).*

Dabei sind folgende Parameter zu beachten:

6	NumberOfMultiGaps	Anzahl der Lücken auf dem Zahnrad
2020	MultiGapIndexPickUp1	Anzeige des Lückenindex PickUp1
2021	MultiGapIndexPickUp2	Anzeige des Lückenindex PickUp2
4006	PU1MultiGapWheelOn	Aktivierung des Mehrfachlückensignals PickUp1
4016	PU2MultiGapWheelOn	Aktivierung des Mehrfachlückensignals PickUp2
6050	MultiGapPositions()	Parametrierung der Lückenpositionen

In das Feld 6050 MultiGapPositions() sind die Zahnnummern der Lückenpositionen einzutragen, d.h. bei jeder eingetragenen Nummer wird statt eines Zahnes eine Lücke erzeugt. Es ist darauf zu achten, dass die Nummern in aufsteigender Reihenfolge eingetragen werden. Außerdem muss die letzte Lückenposition genau der Zähnezahl aus Parameter 1 entsprechen.

Beispiel: MAN-Spezialrad mit 240 Zähnen und 8 Lücken

1	TeethPickUp1	= 240
5	NumberOfMultiGaps	= 8
4003	PU1MultiGapWheelOn	= 1
6050	MultiGapPositions(0)	= 21
6051	MultiGapPositions(1)	= 63
6052	MultiGapPositions(2)	= 91
6053	MultiGapPositions(3)	= 117
6054	MultiGapPositions(4)	= 149

6055 MultiGapPositions(5)	= 179
6056 MultiGapPositions(6)	= 215
6057 MultiGapPositions(7)	= 240

## 2.7 Misfire-Signalerzeugung

Der Motorsimulator ist in der Lage, ein Misfire-Signal zu erzeugen. Dies ist jedoch nicht in der Standard-Simulatorsoftware enthalten, sondern es wird Variante 2 benötigt.

Für die Misfire-Signalerzeugung wird im Steuergerät eine Tabelle mit Drehzahlwerten abgelegt, die auf Anforderung abgespielt werden können. Dadurch kann ein reproduzierbares Misfire-Drehzahlsignal erzeugt werden.

4200 MisfireSignalOn            Aktivierung der Misfire-Signalerzeugung

Wenn dieser Parameter auf 1 gesetzt wird, erfolgt die Signalerzeugung. Die im Steuergerät hinterlegte Tabelle wird einmalig abgespielt, anschließend erfolgt wieder die „normale“ Signalerzeugung. Die Signalerzeugung kann durch Setzen des Parameters auf 0 unterbrochen werden.

Dies kann auch durch die Schalterfunktion 2814 SwMisfireSignalOn erfolgen. Dabei wird die Schaltflanke 0 → 1 ausgewertet und die Simulation aktiviert, falls sie noch nicht läuft und ansonsten deaktiviert.

Damit die Signalanalyse des angeschlossenen Drehzahlreglers korrekt funktioniert, muss die richtige Zähnezahl und Simulationsfrequenz im Motorsimulator und Drehzahlregler parametrisiert werden.

Der Motorsimulator enthält zurzeit das Misfire-Signal eines Guascor-16-Zylinder Gasmotors. Bei diesem Motor wurden 12 unterschiedliche Zylinder für kurze Zeit kurzgeschlossen, so dass der entsprechend Zylinder nicht zündete. Der Drehzahlverlauf wurde aufgezeichnet und kann mit dem Motorsimulator wieder gegeben werden.

Der Motor hatte 165 Zähne und wurde bei 1500 rpm betrieben. Für den Motorsimulator bedeutet dies, dass eine Frequenz von 4125 Hz eingestellt werden muss.

Prinzipiell kann das aufgezeichnete Signal zwar auch auf andere Frequenzen aufmoduliert werden, das entspricht dann aber nicht mehr einem realen Signalverlauf.

Das aufgezeichnete Signal hat mit den obigen Einstellungen eine Dauer von 144 Sekunden und benötigt einen Speicherbedarf von knapp 600 kByte im Steuergerät.

## 2.8 Signalmustererzeugung

Die Ausgänge können auch für die Ausgabe eines Signalmusters verwendet werden. Beim Standard-Simulator stehen dafür drei Ausgänge zur Verfügung, ansonsten bis zu sechs.

Das Signal hat eine Auflösung von 1 ms (falls erforderlich, kann eine genauere Ausgangssignalerzeugung implementiert werden) und eine maximale Dauer von 65.535 s.

Das Signalmuster aller Ausgänge wird zur selben Zeit gestartet, d.h. das Signalmuster aller Ausgänge ist synchron. Der Start des Signalmusters kann entweder über die Funktion 4300 SignalPatternOn oder über die Schalterfunktion 2815 SwSignalPatternOn erfolgen. Das Signalmuster wird so lange erzeugt, bis beide Anforderungen rückgesetzt sind oder die gesamte Signalerzeugungszeit abgelaufen ist. Ein Neustart des Signalmusters erfordert zunächst ein Rücksetzen des Signals.

Die Parametrierung des Signalmusters erfolgt über die Parameter

16000 Pattern1:Time                      Zeitvorgabe für Signalmuster

16100 Pattern1:Level                      Ausgabelevel für Signalmuster

sowie der anschließenden Parameter für die weiteren Signale. Die Parametrierung von 0 ms in einem Element größer Index 0 von Pattern1:Time() kennzeichnet das Ende der Signalerzeugung. Der Signallevel des letzten Elementes wird beibehalten, bis die Signalerzeugungszeit abgelaufen ist.

Die erzeugten Signalmuster werden auf Parameter

2301 Pattern1                              aktueller Ausgabelevel für Signalmuster

und folgende angezeigt. Diese Ausgabelevel können über die normale Parametrierung der Digitalausgänge (Parameter 8800ff) dem gewünschten Ausgang zugewiesen werden. Dabei müssen die Ausgänge als Binärausgänge parametrierung sein (Parameter 4800ff).

## 2.9 Ausgangssignalerzeugung

Es besteht auch die Möglichkeit, ein willkürliches Signal über einen analogen oder einen PWM-Ausgang auszugeben.

Die Parameter für die Signalerzeugung müssen dazu folgendermaßen eingestellt werden:

1700 SignalGenSetpoint Sollwert des Signals

1701 SignalGenAmplitude Signalamplitude

1702 SignalGenFrequency Signalfrequenz

3700 SigGenOutput Signalerzeugung

5701 SignalGenMode Signaltyp

0 = konstantes Signal

1 = Rechtecksignal

2 = Dreiecksignal

3 = Sinussignal

Das anhand von 3700 SigGenOutput erzeugte Signal kann auf jedem beliebigen analogen oder PWM-Ausgang ausgegeben werden.



*Falls ein analoger Ausgang verwendet wird sollte beachtet werden, dass das Signal bei hohen Frequenzen durch den hardwareseitigen Filter auf der Platine gedämpft wird.*

### 3 Pinbelegung

Pin	Bezeichnung	Bedeutung	Verwendung Standard-Simulator
1	PS	Spannungsversorgung	
2	GND	Masse	
3	VCC1	Versorgung Analogeingang 1 (5V)	
4	AI1	Analogeingang 1 (0..5V)	Sollwert 1
5	GND	Masse Analogeingang 1	
6	VCC1	Versorgung Analogeingang 2 (5V)	
7	AI2	Analogeingang 2 (0..5V)	Sollwert 2
8	GND	Masse Analogeingang 2	
9	VCC2	Versorgung Analogeingang 3 (5V)	
10	AI3	Analogeingang 3 (0..5V)	Füllungssignal (analog)
11	GND	Masse Analogeingang 3	
12	VCC2	Versorgung Analogeingang 4 (5V)	
13	AI4	Analogeingang 4 (0..5V)	
14	GND	Masse Analogeingang 4	
15	AI5	Analogeingang 5 (0..25mA)	
16	GND	Masse Analogeingang 5	
17	AI6	Analogeingang 6 (0..25mA)	
18	GND	Masse Analogeingang 6	
19	AO1	Analogausgang 1 (0..25 mA)	
20	GND	Masse Analogausgang 1	
21	AO2	Analogausgang 2 (0..5V)	
22	GND	Masse Analogausgang 2	
23	DI1	Digital-/PWM-Eingang 1	Signal an/aus
24	GND	Masse Digital-/PWM-Eingang 1	
25	DI2	Digital-/PWM-Eingang 2	Sollwert 2 oder 1
26	GND	Masse Digital-/PWM-Eingang 2	
27	DI3	Digital-/PWM-Eingang 3	Auswahl Frequenz- oder Motorsimulation
28	GND	Masse Digital-/PWM-Eingang 3	
29	DI4	Digital-/PWM-Eingang 4	Füllungssignal (PWM)
30	GND	Masse Digital-/PWM-Eingang 4	
31	VDD	Versorgung 5V	
32	GND	Masse	
33	DO1	Frequenz-/PWM-Ausgang 1	Signal Pickup1
34	GND	Masse Frequenz-/PWM-Ausgang 1	
35	DO2	Frequenz-/PWM-Ausgang 2	Signal Pickup2

36	GND	Masse Frequenz-/PWM-Ausgang 2	
37	DO3	Frequenz-/PWM-Ausgang 3	Signal CamIndex
38	GND	Masse Frequenz-/PWM-Ausgang 3	
39	DO4	Frequenz-/PWM-Ausgang 4	
40	GND	Masse Frequenz-/PWM-Ausgang 4	
41	DO5	Frequenz-/PWM-Ausgang 5	
42	GND	Masse Frequenz-/PWM-Ausgang 5	
43	DO6	Frequenz-/PWM-Ausgang 6	
44	GND	Masse Frequenz-/PWM-Ausgang 6	
45	CAN1-H	CAN-Bus 1 High	
46	CAN1-L	CAN-Bus 1 Low	
47	CAN2-H	CAN-Bus 2 High	
48	CAN2-L	CAN-Bus 2 Low	