



**AMKASYN**  
**Applikationsinterface API**  
**ACC-Bus / CANopen Interface**

Version: 2023/25

Teile-Nr.: 200335

"Original Dokumentation"

**AMK***motion*

MEMBER OF THE ARBURG FAMILY

## Impressum

**Name:** PDK\_200335\_API\_de

**Version:**

Version	Änderung	Kurzzeichen
2023/25	AMKmotion Design	LeS

**Bisherige Version:** 2014/51

**Produktstand:**

Produkt (AMK Teile-Nr.)	Firmware Version (Teile-Nr.)	Hardware Version (Teile-Nr.)
KE	ab KE-E03 V3.03 2010/18	-

**Schutzvermerk:**

© AMKmotion GmbH + Co KG

Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und Mitteilung ihres Inhalts wird nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zum Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

**Vorbehalt:**

Änderungen im Inhalt der Dokumentation und Liefermöglichkeit der Produkte sind vorbehalten.

**Herausgeber:**

AMKmotion GmbH + Co KG

Gaußstraße 37-39

73230 Kirchheim unter Teck

Germany

Phone +49 7021 50 05-0

Fax +49 7021 50 05-176

E-Mail [info@amk-motion.com](mailto:info@amk-motion.com)

Registergericht: AG Stuttgart, HRA 230681, Kirchheim unter Teck,

Ust.-Id.-Nr.: DE 145 912 804

Komplementär: AMKmotion Verwaltungsgesellschaft mbH, HRB 774646

**Service:**

Phone +49 7021 50 05-190, Fax -193

Zur schnellen und zuverlässigen Behebung der Störung tragen Sie bei, wenn Sie unseren Service informieren über:

- die Typenschildangaben der Geräte
- die Softwareversion
- die Gerätekonstellation und die Applikation
- die Art der Störung, vermutete Ausfallursache
- die Diagnosemeldungen (Fehlernummern)

E-Mail [service@amk-motion.com](mailto:service@amk-motion.com)

**Internetadresse:**

[www.amk-motion.com](http://www.amk-motion.com)

## Inhalt

<b>Impressum</b>	<b>2</b>
<b>1 Abkürzungen</b>	<b>4</b>
<b>2 Was ist API? Einführung und Übersicht</b>	<b>5</b>
<b>3 Feldbusvariablen und Adressbereiche</b>	<b>7</b>
<b>4 Konventionen</b>	<b>10</b>
<b>5 API Variablen für die Wechselrichter KW und Umrichter KU</b>	<b>11</b>
5.1 Empfang von Sollwerten	11
5.2 Senden von Istwerten	11
5.3 Status- und Steuervariablen	11
5.4 Binäre Ein- und Ausgänge	13
5.5 Diagnosemeldungen	14
5.6 Analogausgänge schreiben	15
5.7 Konfigurierbare 4x32Bit Meldung	15
5.8 AFP Protokoll zur Antriebskommandierung	15
5.9 Vorsteuervariablen Sollwert- und Istwertliste	15
5.10 Feldbusvariablen	16
5.11 Platzhalter-Variablen	17
5.12 Parallelschaltung Servoregler	18
<b>6 API Variablen für die Zweiachs-Wechselrichtermodule KWZ</b>	<b>19</b>
6.1 Empfang von Sollwerten	19
6.2 Senden von Istwerten	19
6.3 Status- und Steuervariablen	19
6.4 Binäre Ein- und Ausgänge	21
6.5 Diagnosemeldungen	21
6.6 Konfigurierbare 4x32Bit Meldung	23
6.7 AFP Protokoll zur Antriebskommandierung	23
6.8 Vorsteuervariablen Sollwert- und Istwertliste	23
<b>7 API Variablen für die KWF Doppelfrequenzumrichter</b>	<b>24</b>
7.1 Empfang von Sollwerten	24
7.2 Status- und Steuervariablen	24
<b>8 API Variablen für die Kompakt-Einspeisung KE</b>	<b>25</b>
8.1 Senden von Istwerten	25
8.2 Status- und Steuervariablen	25
<b>9 API Variablen für AMKSMART IDT Motoren mit integriertem Servoregler</b>	<b>26</b>
9.1 Empfang von Sollwerten	26
9.2 Senden von Istwerten	26
9.3 Status- und Steuervariablen	26
9.4 Binäre Ein- und Ausgänge	28
9.5 Diagnosemeldungen	28
9.6 Konfigurierbare 4x32Bit Meldung	28
9.7 AFP Protokoll zur Antriebskommandierung	29
9.8 Vorsteuervariablen Sollwert- und Istwertliste	29
<b>10 API Variablen für die AMKAMAC Steuerungen AS-PL12, AS-PL14 und AS-C</b>	<b>30</b>
<b>Ihre Meinung zählt!</b>	<b>32</b>

## 1 Abkürzungen

ACC	AMKASYN CAN COMMUNICATION Hardwaresynchronisierte CANopen Schnittstelle
AMK-SPS	AMKASYN speicherprogrammierbare Steuerung
AIPEX	AMK Inbetriebnahme und Parametrier Explorer
AFP	AMK Feldbus Protokoll zur Antriebskommandierung
API	Applikations-Interface
CAN	Controller Area Network (Feldbusschnittstelle)
CAN-S	Hardwaresynchronisiertes CANopen Protokoll
ccb	Concise configuration binary file type *.ccb
ccf	Concise configuration file *.ccf
CiA	CAN in Automation, e.V.
Identnummer	(ID Nr.) Parameter nach SERCOS interface Standard zur Parametrierung des AMKASYN Systems
KE/KW	AMKASYN Gerätereihe Kompakt-Einspeisung, -Wechselrichter
KU	AMKASYN Digitale Kompaktumrichter
Mapping	Für den Datenaustausch über Feldbus stehen Datenpakete zur Verfügung, deren Nutzdateneinträge als „Mapping“ bezeichnet werden.
PDO	Prozess Daten Objekt
PGT	Peripherie Grund Takt
Rx	(Receive) Empfangsdaten
SDO	Service Daten Objekt
SWQ	Sollwertquelle
Tx	(Transmit) Sendedaten

## 2 Was ist API? Einführung und Übersicht

Das Applikationsinterface API ist eine Anwenderschnittstelle, die einen einfachen Zugriff auf die AMK Antriebsfunktionalität ermöglicht. Das API ist ein Variablenfeld der AMK Betriebssoftware (Firmware) und ermöglicht die Steuerung und Analyse des Antriebs. Es unterteilt sich in SendevARIABLEN (z.B. Istwerte, Statusvariablen, konfigurierbare zyklische Anzeigewerte, Variablen für die Auswertung binärer Eingänge...) und EmpfangsvARIABLEN (z.B. 16-/32-Bit Sollwerte, SteuervARIABLEN, Variable für die Steuerung binärer Ausgänge...). Sende- und Empfangsrichtung ist aus Sicht des Antriebs definiert.

AMK Antriebe, die über einen Feldbus miteinander vernetzt sind, können mit Hilfe der API Variablen Antriebsdaten untereinander austauschen.

AMK-SPS Steuerungen (SPS Optionskarten, Reglerkarten mit integrierter SPS, AMKAMAC) haben lesend und schreibend Zugriff die API Variablen der Busteilnehmer.

### Zentrale SPS Systeme:

Bei zentralen SPS Systemen werden Werte (z.B. Sollwerte) aus dem SPS Programm in eine API Variable des Empfangsantriebs gemappt (z.B. Sollwertvariable diMainSetpoint), so dass dort keine weitere Intelligenz zur Dateninterpretation nötig ist. Unter Mapping versteht man die Nachrichtenkonfiguration und Übertragung von Nutzdaten in einer Applikation.

### Dezentrale SPS Systeme:

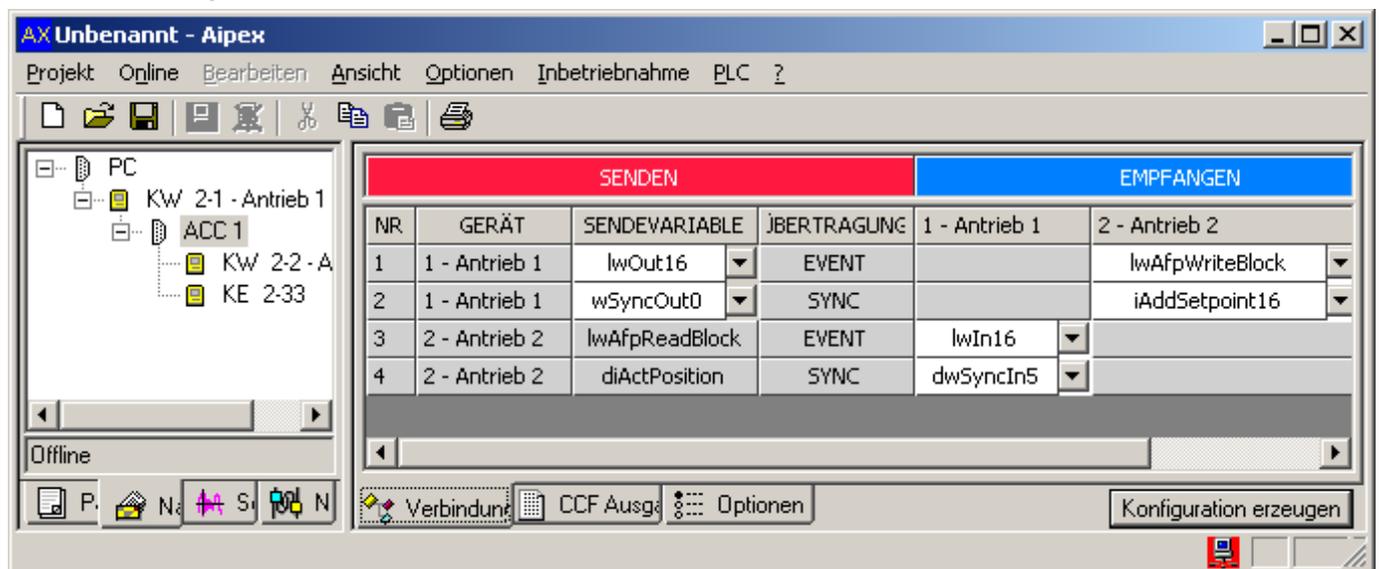
Für die Kommunikation zwischen AMK SPS Steuerungen werden Feldbusvariablen bereitgestellt mit denen Werte aus einer SPS an eine Andere übertragen werden. Die Auswertung und Interpretation der ausgetauschten Daten erfolgt in den SPS Baugruppen. Das Arbeiten mit API Variablen ermöglicht eine effiziente und zielgerichtete, synchrone Datenübertragung zwischen synchronen CANopen Netzwerkteilnehmern (ACC-Bus).

Daten können in der Zykluszeit nach ID2 SERCOS ZYKLUSZEIT ausgetauscht werden.

Der Datenaustausch zwischen den Teilnehmern wird im Nachrichtenkonfigurator der PC Software AIPEX definiert. Für jeden Teilnehmer werden SendevARIABLEN aus einer Liste ausgewählt und anschließend über die Verbindungsliste einem Empfänger zugewiesen.

Aus der Verbindungsliste erstellt AIPEX einen „Kommunikations-Beschreibungs-File“ (\*.ccf, \*.ccb) und speichert die Nachrichtenkonfiguration in einem Parameter. Anschließend muss der „Kommunikations-Beschreibungs-File“ via Parameterübertragung in den ACC-Bus Master geladen werden.

### Nachrichtenkonfiguration mit AIPEX



Die Abbildung zeigt die Verbindungsliste im AIPEX Nachrichtenkonfigurator. Mittels der Verbindungsliste werden Sende- und Empfangsdaten applikationsspezifisch zugeordnet.

Beispiel:

Senden		Empfangen	
Sendevariable	Übertragung	1-Antrieb1 SPS	2-Antrieb2
2-Antrieb2.diActPosition	SYNC	dwSyncln5	-

Die Zuordnung bewirkt, dass der Lageistwert (diActPosition) von Antrieb 2 zyklisch synchronisiert zum Lagereglertakt (ID2 Sercos Zykluszeit) an den Antrieb 1 SPS übertragen wird. Antrieb 1 empfängt den Lageistwert in der Variable dwSyncln5. Die Feldbusvariable dwSyncln5 ist eine symbolische Adressierung auf den Feldbus-Adressbereich eines Gerätes, auf den die SPS lesend und schreibend zugreifen kann.

### 3 Feldbusvariablen und Adressbereiche

Die Feldbusvariablen werden zum Senden und Empfangen von Daten durch eine AMK-SPS Baugruppe verwendet. Den Feldbusvariablen sind Adressbereiche mit unterschiedlicher Bedeutung und Verwendungszweck zugeordnet. Es wird zwischen asynchronen und synchronen Variablen unterschieden. Der synchrone Bereich unterscheidet sich von den asynchronen dadurch, dass Variablen synchronisiert zum Hardwaresynchronisationstakt (Jitter kleiner 1ms) verarbeitet werden. Diese Variablen werden für die Übertragung z.B. von Sollwerten, Istwerten... benutzt. Der asynchrone Bereich dient der Übertragung von zeitunkritischen Daten und des AFP-Protokolls.

Die Adressierung der Feldbusvariablen erfolgt dadurch, dass das „X“ durch eine Zahl ersetzt wird, die einer Adresse des Adressbereichs entspricht. Die folgende Abbildung zeigt die physikalischen Adressbereiche in den verschiedenen AMK-Hardwarekomponenten.

**Hinweis:** Die mit „reserviert“ dargestellten Adressbereiche sind durch das jeweilige "SPS-Zielsystem eingeschränkt.

Bei der Adressierung ist darauf zu achten, dass keine überlappende Adressierungen vorgenommen werden. Bytes „by0“ und „by1“ belegen die gleichen Byteadressen wie das Wort „w0“.

Die in der folgenden Abbildung dargestellten Adressbereiche existieren jeweils einmal als Eingangsbereich zum Empfangen von Variablen und einmal als Ausgangsbereich zum Senden von Variablen.

#### Adressierung Feldbusvariablen

Feldbusvariable n		Adressierungsindex																Beispiel			
Typ	Syntax																				
Long word	lwInx / lwOutx	0								1								lwIn1	lwOut1		
Double word	dwInx / dwOutx	0				1				2				3				lwIn2	lwOut2		
Word	wInx / wOutx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	wIn4	wOut4
Byte	byInx / byOutx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	byIn8	byOut8

**Hinweis:** Der asynchrone Adressbereich wird nur bytekonsistent übertragen.  
Im synchronen Adressbereich werden alle Variablen konsistent übertragen

#### Feldbusvariablen für KW-/KU-R03P

Übertragungstyp	ASYNCHRON			
Adresse in Byte	0 ... 7	8 ... 63	64...127	128 ... 255
Verwendung	Reserviert f. Hardware E/A	Binäre E/A	Res.	AFP-Kommandos
Feldbusvariablen	-	lwIn1...lwIn7 / lwOut1...lwOut7 dwIn2...dwIn15 / dwOut2...dwOut15 wIn4...wIn31 / wOut4...wOut31 byIn8...byIn63 / byOut8...byOut63	-	lwIn16...lwIn31 / lwOut16...lwOut31 dwIn32...dwIn63 / dwOut32...dwOut63 wIn64...wIn127 / wOut64...wOut127 byIn128...byIn255 / byOut128...byOut255

Übertragungstyp	SYNCHRON
Adresse in Byte	0 ... 31
Verwendung	Zyklische synchrone Daten
Feldbusvariablen	lwSynchIn0...lwSynchIn3 / lwSynchOut0...lwSynchOut3 dwSynchIn0...dwSynchIn7 / dwSynchOut0...dwSynchOut7 wSynchIn0...wSynchIn15 / wSynchOut0...wSynchOut15 bySynchIn0...bySynchIn31 / bySynchOut0...bySynchOut31

Feldbusvariablen für KW-/KU-PLC, KW-/KU-PLC1

Übertragungstyp	ASYNCHRON			
Adresse in Byte	0 ... 7	8 ... 63	64...127	128 ... 255
Verwendung	Reserviert f. Hardware E/A	Binäre E/A	Res.	AFP-Kommandos
Feldbusvariablen	-	lwIn1...lwIn7 / lwOut1...lwOut7 dwIn2...dwIn15 / dwOut2...dwOut15 wIn4...wIn31 / wOut4...wOut31 byIn8...byIn63 / byOut8...byOut63	-	lwIn16...lwIn31 / lwOut16...lwOut31 dwIn32...dwIn63 / dwOut32...dwOut63 wIn64...wIn127 / wOut64...wOut127 byIn128...byIn255 / byOut128...byOut255

Übertragungstyp	SYNCHRON			
Adresse in Byte	0 ... 63			
Verwendung	Zyklische synchrone Daten			
Feldbusvariablen	lwSynchIn0...lwSynchIn7 / lwSynchOut0...lwSynchOut7 dwSynchIn0...dwSynchIn15 / dwSynchOut0...dwSynchOut15 wSynchIn0...wSynchIn31 / wSynchOut0...wSynchOut31 bySynchIn0...bySynchIn63 / bySynchOut0...bySynchOut 63			

Feldbusvariablen für KW-/KU-PLC2

Übertragungstyp	ASYNCHRON		
Adresse in Byte	0 ... 7	8 ... 127	128 ... 255
Verwendung	Reserviert f. Hardware E/A	Binäre E/A	AFP-Kommandos
Feldbusvariablen	-	lwIn1...lwIn15 / lwOut1...lwOut15 dwIn2...dwIn31 / dwOut2...dwOut31 wIn4...wIn63 / wOut4...wOut63 byIn8...byIn127 / byOut8...byOut127	lwIn16...lwIn31 / lwOut16...lwOut31 dwIn32...dwIn63 / dwOut32...dwOut63 wIn64...wIn127 / wOut64...wOut127 byIn128...byIn255 / byOut128...byOut255

Übertragungstyp	SYNCHRON			
Adresse in Byte	0 ... 127			
Verwendung	Zyklische synchrone Daten			
Feldbusvariablen	lwSynchIn0...lwSynchIn15 / lwSynchOut0...lwSynchOut15 dwSynchIn0...dwSynchIn31 / dwSynchOut0...dwSynchOut31 wSynchIn0...wSynchIn63 / wSynchOut0...wSynchOut63 bySynchIn0...bySynchIn127 / bySynchOut0...bySynchOut 127			

Feldbusvariablen für die AMKAMAC Steuerung AS-PL12 mit eingebauter AS-FCT1 Baugruppe

Übertragungstyp	ASYNCHRON	
Adresse in Byte	0 ... 255	256 ... 511
Verwendung	Binäre E/A	AFP-Kommandos
Feldbusvariablen	lwIn0...lwIn63 / lwOut...lwOut63 dwIn0...dwIn127 / dwOut...dwOut127 wIn0...wIn255 / wOut0...wOut255 byIn0...byIn511 / byOut0...byOut511	

<b>Übertragungstyp</b>	<b>SYNCHRON</b>
<b>Adresse in Byte</b>	<b>0 ... 255</b>
<b>Verwendung</b>	Zyklische synchrone Daten
<b>Feldbusvariablen</b>	lwSyncln0...lwSyncln31 / lwSyncOut0...lwSyncOut31 dwSyncln0...dwSyncln63 / dwSyncOut0...dwSyncOut63 wSyncln0...wSyncln127 / wSyncOut0...wSyncOut127 bySyncln0...bySyncln255 / bySyncOut0...bySyncOut255

**Feldbusvariablen für die AMKAMAC Steuerung AS-PL14 mit eingebauter AS-FCT2 Baugruppe**

<b>Übertragungstyp</b>	<b>ASYNCHRON</b>	
<b>Adresse in Byte</b>	<b>0 ... 255</b>	<b>256 ... 511</b>
<b>Verwendung</b>	Binäre E/A	AFP-Kommandos
<b>Feldbusvariablen</b>	lwIn0...lwIn63 / lwOut...lwOut63 dwIn0...dwIn127 / dwOut0...dwOut127 wIn0...wIn255 / wOut0...wOut255 byIn0...byIn511 / byOut0...byOut511	

<b>Übertragungstyp</b>	<b>SYNCHRON</b>
<b>Adresse in Byte</b>	<b>0 ... 511</b>
<b>Verwendung</b>	Zyklische synchrone Daten
<b>Feldbusvariablen</b>	lwSyncln0...lwSyncln63 / lwSyncOut0...lwSyncOut63 dwSyncln0...dwSyncln127 / dwSyncOut0...dwSyncOut127 wSyncln0...wSyncln255 / wSyncOut0...wSyncOut255 bySyncln0...bySyncln511 / bySyncOut0...bySyncOut511

**Feldbusvariablen für die AMKAMAC Steuerung AS-C mit eingebauter AS-FCT1 Baugruppe**

<b>Übertragungstyp</b>	<b>ASYNCHRON</b>	
<b>Adresse in Byte</b>	<b>0 ... 255</b>	<b>256 ... 511</b>
<b>Verwendung</b>	Binäre E/A	AFP-Kommandos
<b>Feldbusvariablen</b>	lwIn0...lwIn63 / lwOut...lwOut63 dwIn0...dwIn127 / dwOut0...dwOut127 wIn0...wIn255 / wOut0...wOut255 byIn0...byIn511 / byOut0...byOut511	

<b>Übertragungstyp</b>	<b>SYNCHRON</b>
<b>Adresse in Byte</b>	<b>0 ... 127</b>
<b>Verwendung</b>	Zyklische synchrone Daten
<b>Feldbusvariablen</b>	lwSyncln0...lwSyncln15 / lwSyncOut0...lwSyncOut15 dwSyncln0...dwSyncln31 / dwSyncOut0...dwSyncOut31 wSyncln0...wSyncln63 / wSyncOut0...wSyncOut63 bySyncln0...bySyncln127 / bySyncOut0...bySyncOut127

## 4 Konventionen

Die API Variablennamen beginnen mit einem PRÄFIX, der den Datentyp kennzeichnet:

Datentyp-	Typ-Bez.-	Schlüsselwort	Präfix	Beispiel
Bitfolge 8 Bit	Byte	BYTE	<b>by</b>	byAxisState
Bitfolge 16 Bit	Wort	WORD	<b>w</b>	wRealTimeBits
Bitfolge 32 Bit	Doppelwort	DWORD	<b>dw</b>	dwDummy
Bitfolge 64 Bit	Longwort	LWORD	<b>lw</b>	lwOutX
Ganze Zahl	Integer	INT	<b>i</b>	iActValue0
Doppelte ganze Zahl	Doppelinteger	DINT	<b>di</b>	diActPosition

## 5 API Variablen für die Wechselrichter KW und Umrichter KU

### 5.1 Empfang von Sollwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diMainSetpoint	2030	01	Rx	32Bit Sollwert, relevant in den Betriebsarten ID32800...
diReserve1	2030	02	Rx	reserviert
iAddSetpoint16	2030	03	Rx	16Bit inkrementeller Lagesollwert relevant in den Betriebsarten ID32800...
diReserve2	2030	04	Rx	reserviert
diAddSetpoint32	2030	05	Rx	32Bit Sollwert für die Funktion Synchronlauf mit Winkelausrichtung (SWQW), relevant in den Betriebsarten ID32800...

### 5.2 Senden von Istwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iMessage16	2040	01	Tx	Konfigurierbare zyklische 16Bit Istwertmeldung (ID32785)
diMessage32	2040	02	Tx	Konfigurierbare zyklische 32Bit Istwertmeldung (ID32786)
diActPosition	2040	03	Tx	32Bit Lageistwert

### 5.3 Status- und Steuervariablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																																	
wDeviceState	2048	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SBM</td> <td>System bereit Meldung</td> </tr> <tr> <td>1</td> <td>QUE</td> <td>Quittierung Umrichter Ein (nur für Geräte KU)</td> </tr> <tr> <td>2 ...3</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>4</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>5</td> <td></td> <td>Quittierung Pulsweitenmessung</td> </tr> <tr> <td>6</td> <td>PR_POS</td> <td>Quittierung Messtaster positive Flanke</td> </tr> <tr> <td>7</td> <td>PR_NEG</td> <td>Quittierung Messtaster negative Flanke</td> </tr> <tr> <td>8</td> <td>QRF</td> <td>Quittierung Regler -freigabe</td> </tr> <tr> <td>9 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	SBM	System bereit Meldung	1	QUE	Quittierung Umrichter Ein (nur für Geräte KU)	2 ...3	-	Reserviert	4	-	Reserviert	5		Quittierung Pulsweitenmessung	6	PR_POS	Quittierung Messtaster positive Flanke	7	PR_NEG	Quittierung Messtaster negative Flanke	8	QRF	Quittierung Regler -freigabe	9 ...15	-	Reserviert
Statusbits																																					
BIT	SYNTAX	BEDEUTUNG																																			
0	SBM	System bereit Meldung																																			
1	QUE	Quittierung Umrichter Ein (nur für Geräte KU)																																			
2 ...3	-	Reserviert																																			
4	-	Reserviert																																			
5		Quittierung Pulsweitenmessung																																			
6	PR_POS	Quittierung Messtaster positive Flanke																																			
7	PR_NEG	Quittierung Messtaster negative Flanke																																			
8	QRF	Quittierung Regler -freigabe																																			
9 ...15	-	Reserviert																																			

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																																						
wDeviceControl	2049	00	Rx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FL</td> <td>Fehler löschen</td> </tr> <tr> <td>1</td> <td>UE</td> <td>Umrichter Ein <sup>1)</sup> (nur für Geräte KU)</td> </tr> <tr> <td>2...4</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>5</td> <td></td> <td>Start Pulsweitenmessung</td> </tr> <tr> <td>6</td> <td>CLR-I</td> <td>I-Anteil des Drehzahl- reglers wird gelöscht</td> </tr> <tr> <td>7</td> <td>PR_EN</td> <td>Freigabe Messfunktion</td> </tr> <tr> <td>8</td> <td>RF</td> <td>Reglerfreigabe <sup>1)</sup></td> </tr> <tr> <td>9</td> <td></td> <td>Restweg löschen</td> </tr> <tr> <td>10...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	FL	Fehler löschen	1	UE	Umrichter Ein <sup>1)</sup> (nur für Geräte KU)	2...4	-	Reserviert	5		Start Pulsweitenmessung	6	CLR-I	I-Anteil des Drehzahl- reglers wird gelöscht	7	PR_EN	Freigabe Messfunktion	8	RF	Reglerfreigabe <sup>1)</sup>	9		Restweg löschen	10...15	-	Reserviert					
Steuerbits																																										
BIT	SYNTAX	BEDEUTUNG																																								
0	FL	Fehler löschen																																								
1	UE	Umrichter Ein <sup>1)</sup> (nur für Geräte KU)																																								
2...4	-	Reserviert																																								
5		Start Pulsweitenmessung																																								
6	CLR-I	I-Anteil des Drehzahl- reglers wird gelöscht																																								
7	PR_EN	Freigabe Messfunktion																																								
8	RF	Reglerfreigabe <sup>1)</sup>																																								
9		Restweg löschen																																								
10...15	-	Reserviert																																								
wRealTimeBits	204A	00	Tx	<table border="1"> <thead> <tr> <th colspan="2">Echtzeitbitmeldungen</th> </tr> <tr> <th>BIT</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><math> n_{soll} - n_{ist}  &lt; n_{Fenster}</math> (<math>n_{ist} = n_{soll}</math>), Fenster nach ID157</td> </tr> <tr> <td>1</td> <td><math> n_{ist}  &lt; n_{min}</math>, Schwelle nach ID124</td> </tr> <tr> <td>2</td> <td><math> n_{ist}  &lt; n_x</math>, Schwelle nach ID125</td> </tr> <tr> <td>3</td> <td><math> M_{ist}  &gt; M_x</math>, Schwelle nach ID126</td> </tr> <tr> <td>4</td> <td><math> M_{ist}  &gt; M_{grenz}</math>, Grenzen nach ID82, ID83</td> </tr> <tr> <td>5</td> <td><math> n_{soll}  &gt;  n_{grenz} </math>, Grenzen nach ID38, ID39</td> </tr> <tr> <td>6</td> <td>In Position Fenster nach ID57</td> </tr> <tr> <td>7</td> <td><math> P_{ist}  &gt; P_x</math>, Schwelle nach ID158</td> </tr> <tr> <td>8</td> <td>Softwareendschalter negativ, nach ID50</td> </tr> <tr> <td>9</td> <td>Antrieb winkelsynchron, Fenster nach ID228</td> </tr> <tr> <td>10</td> <td>Antrieb drehzahlsynchron, Fenster nach ID32952</td> </tr> <tr> <td>11</td> <td><math>n_{ist} &gt; 0</math></td> </tr> <tr> <td>12</td> <td>Quittierung Istwert geeicht</td> </tr> <tr> <td>13</td> <td>Restweg wurde gelöscht, nach ID32922</td> </tr> <tr> <td>14</td> <td>Überstrommeldung: Auslastung &gt; 50% der Überlastgrenze</td> </tr> <tr> <td>15</td> <td>Softwareendschalter positiv Fenster ID49</td> </tr> <tr> <td colspan="2">                     Legende:                      N Drehzahl, M Drehmoment,                      P Leistung, X Lage                 </td> </tr> </tbody> </table>	Echtzeitbitmeldungen		BIT	BEDEUTUNG	0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157	1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124	2	$ n_{ist}  < n_x$ , Schwelle nach ID125	3	$ M_{ist}  > M_x$ , Schwelle nach ID126	4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83	5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39	6	In Position Fenster nach ID57	7	$ P_{ist}  > P_x$ , Schwelle nach ID158	8	Softwareendschalter negativ, nach ID50	9	Antrieb winkelsynchron, Fenster nach ID228	10	Antrieb drehzahlsynchron, Fenster nach ID32952	11	$n_{ist} > 0$	12	Quittierung Istwert geeicht	13	Restweg wurde gelöscht, nach ID32922	14	Überstrommeldung: Auslastung > 50% der Überlastgrenze	15	Softwareendschalter positiv Fenster ID49	Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage	
Echtzeitbitmeldungen																																										
BIT	BEDEUTUNG																																									
0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157																																									
1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124																																									
2	$ n_{ist}  < n_x$ , Schwelle nach ID125																																									
3	$ M_{ist}  > M_x$ , Schwelle nach ID126																																									
4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83																																									
5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39																																									
6	In Position Fenster nach ID57																																									
7	$ P_{ist}  > P_x$ , Schwelle nach ID158																																									
8	Softwareendschalter negativ, nach ID50																																									
9	Antrieb winkelsynchron, Fenster nach ID228																																									
10	Antrieb drehzahlsynchron, Fenster nach ID32952																																									
11	$n_{ist} > 0$																																									
12	Quittierung Istwert geeicht																																									
13	Restweg wurde gelöscht, nach ID32922																																									
14	Überstrommeldung: Auslastung > 50% der Überlastgrenze																																									
15	Softwareendschalter positiv Fenster ID49																																									
Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage																																										
byAxisState	204D	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>REF_ACK</td> <td>Referenzpunkt bekannt</td> </tr> <tr> <td>1...7</td> <td></td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	REF_ACK	Referenzpunkt bekannt	1...7		Reserviert																										
Steuerbits																																										
BIT	SYNTAX	BEDEUTUNG																																								
0	REF_ACK	Referenzpunkt bekannt																																								
1...7		Reserviert																																								
wStatusBitsId144	204E	00	Tx	Frei konfigurierbares Statuswort zur Anzeige von Echtzeitbits aus der Liste der Codes für Binärausgänge. Siehe auch Dokumentation Parameter unter ID144 „Statusbits“. Der Anzeigeinhalt des Statusworts ID144 wird über ID26 „Konfigurationsliste Statusbits“ festgelegt.																																						

1) Um die Signale UE und RF über die API Variable steuern zu können, ist folgende Parametrierung notwendig:

ID32795=5 Quelle UE  
ID32796=5 Quelle RF

## 5.4 Binäre Ein- und Ausgänge

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
bySystemIn	2060	00	Tx	Sendet das Prozessabbild „Binäre Eingänge“ der Eingänge BE1, BE2 und BE3 auf der Reglerkarte
bySystemOut	2070	00	Rx	Empfängt Prozessabbild „Binäre Ausgänge“ zur Ausgabe auf den Binärausgängen BA1, BA2 und BA3 der Reglerkarte <sup>1)</sup>
byInp1Byte0	6000	01	Tx	Jedes gesendete Byte stellt das Prozessabbild „Binäre Eingänge“ von 8 binären Eingängen zur Verfügung. (z.B. E/A Optionskarte KU- oder KW-EA2 im Steckplatz 1: E1...E8 werden durch die Variable byInp1Byte0, E9...E12 durch byInp1Byte1 abgebildet)
byInp1Byte1	6000	02	Tx	
byInp1Byte2	6000	03	Tx	
byInp1Byte3	6000	04	Tx	
byInp1Byte4	6000	05	Tx	
byInp1Byte5	6000	06	Tx	
byInp1Byte6	6000	07	Tx	
byInp1Byte7	6000	08	Tx	
byOutp1Byte0	6200	01	Rx	Die eingehenden Bytes als „Prozessabbild Ausgänge“ werden auf die binären Ausgänge übertragen (z.B. E/A Optionskarte KU- oder KW-EA2 im Steckplatz 1 A1...A8 werden durch die Variable byOutp1Byte0 gesteuert) <sup>2)</sup>
byOutp1Byte1	6200	02	Rx	
byOutp1Byte2	6200	03	Rx	
byOutp1Byte3	6200	04	Rx	
byOutp1Byte4	6200	05	Rx	
byOutp1Byte5	6200	06	Rx	
byOutp1Byte6	6200	07	Rx	
byOutp1Byte7	6200	08	Rx	

<sup>1)</sup> Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.

ID32864=0 Adresse Ausgangsport 3

<sup>2)</sup> Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.

ID32846=0 Adresse Ausgangsport 1

## 5.5 Diagnosemeldungen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung	
				BIT	BEDEUTUNG
wSysWarn	2104	00	Tx	<b>BIT</b>	<b>BEDEUTUNG</b>
				0	Reserviert
				1	Übertemperatur externe Komponente R0x: Übertemperatur Motor (Motorkalleiter) KE, KU: Übertemperatur Bremswiderstand
				2 ...5	Reserviert
				6	R0x: Messimpuls am Eingang TPI der KW-EN1 Optionskarte bzw. BE3 der Reglerkarte R0x wurde nichtinnerhalb des Fensters ID34155 erkannt
				7 ...15	Reserviert
wAxisError	2105	00	Tx	Gültig für R0x	
				<b>BIT</b>	<b>BEDEUTUNG</b>
				0	Nennstromüberschreitung (I <sup>2</sup> t-Überwachung)
				1	Reserviert
				2	Übertemperatur Motor (Motorkalleiter)
				3	Reserviert
				4	Logikspannungsfehler Schaltnetzteil
				5	Geberfehler
				6	Fehler Softwarekommutierung
				7	Kurzschluss / Erdschluss Motor- / Motorklemme
				8	Überspannung Zwischenkreis
				9	Unterspannung Zwischenkreis
				10	Reserviert
				11	Lageregelfehler (Exzessive Regelabweichung)
				12	Kommunikationsfehler SERCOSinterface
				13	Lagegrenzwert (Softwaregrenze) verletzt
14	Reserviert				
15	Herstellerspezifischer Fehler (siehe Diagnosebeschreibung)				
byErrorSys	204C	01	Tx	Die Auswertung der Bytes dient zur Erkennung von Fehler- bzw. Warn-meldungen. Ein Fehler im System liegt vor, wenn eines der beiden Bytes einen Wert ungleich Null liefert und das "System Bereit" Bit (SBM) nicht gesetzt ist. Eine Warnung im System liegt vor, wenn eines der beiden Bytes einen Wert ungleich Null liefert und das "System Bereit" Bit (SBM) gesetzt ist	
byErrorOpt	204C	02	Tx	Die Auswertung kann z.B. in der AMK-SPS programmiert werden. Durch das Lesen von ID390 Diagnosenummer oder ID32840 Diagnoseliste kann die Diagnosenummer (siehe Dokumentation Diagnosemeldungen) ausgelesen werden	

## 5.6 Analogausgänge schreiben

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iAnalogOutp1	6411	01	Rx	Analogausgangswert 1
iAnalogOutp2	6411	02	Rx	Analogausgangswert 2
iAnalogOutp3	6411	03	Rx	Analogausgangswert 3
iAnalogOutp4	6411	04	Rx	Analogausgangswert 4

**Hinweis:** Analogausgänge sind über die Option KW-SM1 T.-Nr.:29509 verfügbar.

## 5.7 Konfigurierbare 4x32Bit Meldung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diSetpointScr1	2050	01	Tx	Inhalt 1. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948)
diSetpointScr2	2050	02	Tx	Inhalt 2. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948)
diSetpointScr3	2050	03	Tx	Inhalt 3. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948)
diSetpointScr4	2050	04	Tx	Inhalt 4. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948)

## 5.8 AFP Protokoll zur Antriebskommandierung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
lwAFPReadBlock	2020	01-08	Tx	8Byte AFP Statusdaten vom Antrieb
lwAFPWriteBlock	2021	01-08	Rx	8Byte AFP Steuerdaten zum Antrieb

## 5.9 Vorsteuervariablen Sollwert- und Istwertliste

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iSetpoint0	2080	01	Rx	ID32838 Sollwertliste
iSetpoint1	2080	02	Rx	ID32838 Sollwertliste
iSetpoint2	2080	03	Rx	ID32838 Sollwertliste
iSetpoint3	2080	04	Rx	ID32838 Sollwertliste
diSetpoint0	2081	01	Rx	ID32838 Sollwertliste
diSetpoint1	2081	02	Rx	ID32838 Sollwertliste
iActValue0	2090	01	Tx	ID32839 Istwertliste

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iActValue1	2090	02	Tx	ID32839 Istwertliste
diActValue0	2091	01	Tx	ID32839 Istwertliste
diActValue1	2091	02	Tx	ID32839 Istwertliste

## 5.10 Feldbusvariablen

Feldbusvariablen werden von der SPS benutzt um Daten über Feldbus mit anderen SPS Baugruppen oder Antrieben austauschen zu können. Das SPS Programm schreibt Werte in die Sendvariablen und interpretiert Empfangsdaten.

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung												
lwInX lwOutX	1)		Rx Tx	<table border="1"> <tr> <td colspan="2">Longwort asynchron (64Bit)</td> </tr> <tr> <th>Hardware</th> <th>Index X [LWORD]</th> </tr> <tr> <td>KU-/KW-R03P</td> <td rowspan="4">1 ...31</td> </tr> <tr> <td>KU-/KW-PLC</td> </tr> <tr> <td>KU-/KW-PLC1</td> </tr> <tr> <td>KU-/KW-PLC2</td> </tr> </table>	Longwort asynchron (64Bit)		Hardware	Index X [LWORD]	KU-/KW-R03P	1 ...31	KU-/KW-PLC	KU-/KW-PLC1	KU-/KW-PLC2			
Longwort asynchron (64Bit)																
Hardware	Index X [LWORD]															
KU-/KW-R03P	1 ...31															
KU-/KW-PLC																
KU-/KW-PLC1																
KU-/KW-PLC2																
dwInX dwOutX	2000 200C	01+X 01+X	Rx Tx	<table border="1"> <tr> <td colspan="2">Doppelwort asynchron (32Bit)</td> </tr> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> <tr> <td>KU-/KW-R03P</td> <td rowspan="4">2 ...63</td> </tr> <tr> <td>KU-/KW-PLC</td> </tr> <tr> <td>KU-/KW-PLC1</td> </tr> <tr> <td>KU-/KW-PLC2</td> </tr> </table>	Doppelwort asynchron (32Bit)		Hardware	Index X [DWORD]	KU-/KW-R03P	2 ...63	KU-/KW-PLC	KU-/KW-PLC1	KU-/KW-PLC2			
Doppelwort asynchron (32Bit)																
Hardware	Index X [DWORD]															
KU-/KW-R03P	2 ...63															
KU-/KW-PLC																
KU-/KW-PLC1																
KU-/KW-PLC2																
wInX wOutX	2001 200D	01+X 01+X	Rx Tx	<table border="1"> <tr> <td colspan="2">Wort asynchron (16Bit)</td> </tr> <tr> <th>Hardware</th> <th>Index X [WORD]</th> </tr> <tr> <td>KU-/KW-R03P</td> <td rowspan="4">4 ...127</td> </tr> <tr> <td>KU-/KW-PLC</td> </tr> <tr> <td>KU-/KW-PLC1</td> </tr> <tr> <td>KU-/KW-PLC2</td> </tr> </table>	Wort asynchron (16Bit)		Hardware	Index X [WORD]	KU-/KW-R03P	4 ...127	KU-/KW-PLC	KU-/KW-PLC1	KU-/KW-PLC2			
Wort asynchron (16Bit)																
Hardware	Index X [WORD]															
KU-/KW-R03P	4 ...127															
KU-/KW-PLC																
KU-/KW-PLC1																
KU-/KW-PLC2																
byInX byOutX	2002 200E	01+X 01+X	Rx Tx	<table border="1"> <tr> <td colspan="2">Byte asynchron (8Bit)</td> </tr> <tr> <th>Hardware</th> <th>Index X [BYTE]</th> </tr> <tr> <td>KU-/KW-R03P</td> <td rowspan="4">8 ...255</td> </tr> <tr> <td>KU-/KW-PLC</td> </tr> <tr> <td>KU-/KW-PLC1</td> </tr> <tr> <td>KU-/KW-PLC2</td> </tr> </table>	Byte asynchron (8Bit)		Hardware	Index X [BYTE]	KU-/KW-R03P	8 ...255	KU-/KW-PLC	KU-/KW-PLC1	KU-/KW-PLC2			
Byte asynchron (8Bit)																
Hardware	Index X [BYTE]															
KU-/KW-R03P	8 ...255															
KU-/KW-PLC																
KU-/KW-PLC1																
KU-/KW-PLC2																
lwSynInX lwSynOutX	1)		Rx Tx	<table border="1"> <tr> <td colspan="2">Longwort synchron (64Bit)</td> </tr> <tr> <th>Hardware</th> <th>Index X [LWORD]</th> </tr> <tr> <td>KU-/KW-R03P</td> <td>0 ...3</td> </tr> <tr> <td>KU-/KW-PLC</td> <td>0 ...7</td> </tr> <tr> <td>KU-/KW-PLC1</td> <td>0 ...7</td> </tr> <tr> <td>KU-/KW-PLC2</td> <td>0 ...15</td> </tr> </table>	Longwort synchron (64Bit)		Hardware	Index X [LWORD]	KU-/KW-R03P	0 ...3	KU-/KW-PLC	0 ...7	KU-/KW-PLC1	0 ...7	KU-/KW-PLC2	0 ...15
Longwort synchron (64Bit)																
Hardware	Index X [LWORD]															
KU-/KW-R03P	0 ...3															
KU-/KW-PLC	0 ...7															
KU-/KW-PLC1	0 ...7															
KU-/KW-PLC2	0 ...15															

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung	
dwSyncInX dwSyncOutX	2003 200F	01+X 01+X	Rx Tx	Doppelwort synchron	
				<b>Hardware</b>	<b>Index X [DWORD]</b>
				KU-/KW-R03P	0 ...7
				KU-/KW-PLC	0 ...15
				KU-/KW-PLC1	0 ...15
KU-/KW-PLC2	0 ...31				
wSyncInX wSyncOutX	2004 2010	01+X 01+X	Rx Tx	Wort synchron	
				<b>Hardware</b>	<b>Index X [WORD]</b>
				KU-/KW-R03P	0 ...15
				KU-/KW-PLC	0 ...31
				KU-/KW-PLC1	0 ...31
KU-/KW-PLC2	0 ...63				
bySyncInX bySyncOutX	2005 2011	01+X 01+X	Rx Tx	Byte synchron	
				<b>Hardware</b>	<b>Index X [BYTE]</b>
				KU-/KW-R03P	0 ...31
				KU-/KW-PLC	0 ...63
				KU-/KW-PLC1	0 ...63
KU-/KW-PLC2	0 ...127				

<sup>1)</sup> lw (Longwort) wird AMK intern aus Bytes zusammengesetzt →Byte mapping z.B. lwOut10 beginnt mit byOut80

### 5.11 Platzhalter-Variablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
dwDummy	0004	00	Rx	Receive PDO Mapping ohne Dateninhalt, Platzhaltervariablen
wDummy	0003	00	Rx	
byDummy	0002	00	Rx	

## 5.12 Parallelschaltung Servoregler

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
uwFieldAngleMaster	2112	00	Tx	Feldwinkel Master
swlsqMaster	2113	00	Tx	ISQ Master (Stromsollwert)
swlsdMaster	2114	00	TX	ISD Master (Stromsollwert)
uwFieldAngleSlave	2115	00	Rx	Feldwinkel Slave
swlsqSlave	0003	00	Rx	ISQ Slave (Stromsollwert)
swlsdSlave	0002	00	Rx	ISD Slave (Stromsollwert)

## 6 API Variablen für die Zweiachs-Wechselrichtermodule KWZ

### 6.1 Empfang von Sollwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diMainSetpoint	2030	01	Rx	32Bit Sollwert, relevant in den Betriebsarten ID32800...
diReserve1	2030	02	Rx	reserviert
iAddSetpoint16	2030	03	Rx	16Bit inkrementeller Lagesollwert relevant in den Betriebsarten ID32800...
diReserve2	2030	04	Rx	reserviert
diAddSetpoint32	2030	05	Rx	32Bit Sollwert für die Funktion Synchronlauf mit Winkelausrichtung (SWQW), relevant in den Betriebsarten ID32800...
iSetSpeed	2030	06	Rx	Drehzahlsollwert 16 Bit in (U/min)

### 6.2 Senden von Istwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iMessage16	2040	01	Tx	Konfigurierbare zyklische 16Bit Istwertmeldung (ID32785)
diMessage32	2040	02	Tx	Konfigurierbare zyklische 32Bit Istwertmeldung (ID32786)
diActPosition	2040	03	Tx	32Bit Lageistwert

### 6.3 Status- und Steuervariablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																																	
wDeviceState	2048	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SBM</td> <td>System bereit Meldung</td> </tr> <tr> <td>1</td> <td>QUE</td> <td>Quittierung Umrichter Ein (nur für Geräte KU)</td> </tr> <tr> <td>2 ...3</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>4</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>5</td> <td></td> <td>Quittierung Pulsweitenmessung</td> </tr> <tr> <td>6</td> <td>PR_POS</td> <td>Quittierung Messtaster positive Flanke</td> </tr> <tr> <td>7</td> <td>PR_NEG</td> <td>Quittierung Messtaster negative Flanke</td> </tr> <tr> <td>8</td> <td>QRF</td> <td>Quittierung Reglerfreigabe</td> </tr> <tr> <td>9 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	SBM	System bereit Meldung	1	QUE	Quittierung Umrichter Ein (nur für Geräte KU)	2 ...3	-	Reserviert	4	-	Reserviert	5		Quittierung Pulsweitenmessung	6	PR_POS	Quittierung Messtaster positive Flanke	7	PR_NEG	Quittierung Messtaster negative Flanke	8	QRF	Quittierung Reglerfreigabe	9 ...15	-	Reserviert
Statusbits																																					
BIT	SYNTAX	BEDEUTUNG																																			
0	SBM	System bereit Meldung																																			
1	QUE	Quittierung Umrichter Ein (nur für Geräte KU)																																			
2 ...3	-	Reserviert																																			
4	-	Reserviert																																			
5		Quittierung Pulsweitenmessung																																			
6	PR_POS	Quittierung Messtaster positive Flanke																																			
7	PR_NEG	Quittierung Messtaster negative Flanke																																			
8	QRF	Quittierung Reglerfreigabe																																			
9 ...15	-	Reserviert																																			

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																																						
wDeviceControl	2049	00	Rx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FL</td> <td>Fehler löschen</td> </tr> <tr> <td>1</td> <td>UE</td> <td>Umrichter Ein <sup>1)</sup> (nur für Geräte KU)</td> </tr> <tr> <td>2...4</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>5</td> <td></td> <td>Start Pulsweitenmessung</td> </tr> <tr> <td>6</td> <td>CLR-I</td> <td>I-Anteil des Drehzahl- reglers wird gelöscht</td> </tr> <tr> <td>7</td> <td>PR_EN</td> <td>Freigabe Messfunktion</td> </tr> <tr> <td>8</td> <td>RF</td> <td>Reglerfreigabe <sup>1)</sup></td> </tr> <tr> <td>9</td> <td></td> <td>Restweg löschen</td> </tr> <tr> <td>10...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	FL	Fehler löschen	1	UE	Umrichter Ein <sup>1)</sup> (nur für Geräte KU)	2...4	-	Reserviert	5		Start Pulsweitenmessung	6	CLR-I	I-Anteil des Drehzahl- reglers wird gelöscht	7	PR_EN	Freigabe Messfunktion	8	RF	Reglerfreigabe <sup>1)</sup>	9		Restweg löschen	10...15	-	Reserviert					
Steuerbits																																										
BIT	SYNTAX	BEDEUTUNG																																								
0	FL	Fehler löschen																																								
1	UE	Umrichter Ein <sup>1)</sup> (nur für Geräte KU)																																								
2...4	-	Reserviert																																								
5		Start Pulsweitenmessung																																								
6	CLR-I	I-Anteil des Drehzahl- reglers wird gelöscht																																								
7	PR_EN	Freigabe Messfunktion																																								
8	RF	Reglerfreigabe <sup>1)</sup>																																								
9		Restweg löschen																																								
10...15	-	Reserviert																																								
wRealTimeBits	204A	00	Tx	<table border="1"> <thead> <tr> <th colspan="2">Echtzeitbitmeldungen</th> </tr> <tr> <th>BIT</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><math> n_{soll} - n_{ist}  &lt; n_{Fenster}</math> (<math>n_{ist} = n_{soll}</math>), Fenster nach ID157</td> </tr> <tr> <td>1</td> <td><math> n_{ist}  &lt; n_{min}</math>, Schwelle nach ID124</td> </tr> <tr> <td>2</td> <td><math> n_{ist}  &lt; n_x</math>, Schwelle nach ID125</td> </tr> <tr> <td>3</td> <td><math> M_{ist}  &gt; M_x</math>, Schwelle nach ID126</td> </tr> <tr> <td>4</td> <td><math> M_{ist}  &gt; M_{grenz}</math>, Grenzen nach ID82, ID83</td> </tr> <tr> <td>5</td> <td><math> n_{soll}  &gt;  n_{grenz} </math>, Grenzen nach ID38, ID39</td> </tr> <tr> <td>6</td> <td>In Position Fenster nach ID57</td> </tr> <tr> <td>7</td> <td><math> P_{ist}  &gt; P_x</math>, Schwelle nach ID158</td> </tr> <tr> <td>8</td> <td>Softwareendschalter negativ, nach ID50</td> </tr> <tr> <td>9</td> <td>Antrieb winkelsynchron, Fenster nach ID228</td> </tr> <tr> <td>10</td> <td>Antrieb drehzahlsynchron, Fenster nach ID32952</td> </tr> <tr> <td>11</td> <td><math>n_{ist}^3 \cdot 0</math></td> </tr> <tr> <td>12</td> <td>Quittierung Istwert geeicht</td> </tr> <tr> <td>13</td> <td>Restweg wurde gelöscht, nach ID32922</td> </tr> <tr> <td>14</td> <td>Überstrommeldung: Auslastung &gt; 50% der Überlastgrenze</td> </tr> <tr> <td>15</td> <td>Softwareendschalter positiv Fenster ID49</td> </tr> <tr> <td colspan="2">Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage</td> </tr> </tbody> </table>	Echtzeitbitmeldungen		BIT	BEDEUTUNG	0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157	1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124	2	$ n_{ist}  < n_x$ , Schwelle nach ID125	3	$ M_{ist}  > M_x$ , Schwelle nach ID126	4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83	5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39	6	In Position Fenster nach ID57	7	$ P_{ist}  > P_x$ , Schwelle nach ID158	8	Softwareendschalter negativ, nach ID50	9	Antrieb winkelsynchron, Fenster nach ID228	10	Antrieb drehzahlsynchron, Fenster nach ID32952	11	$n_{ist}^3 \cdot 0$	12	Quittierung Istwert geeicht	13	Restweg wurde gelöscht, nach ID32922	14	Überstrommeldung: Auslastung > 50% der Überlastgrenze	15	Softwareendschalter positiv Fenster ID49	Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage	
Echtzeitbitmeldungen																																										
BIT	BEDEUTUNG																																									
0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157																																									
1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124																																									
2	$ n_{ist}  < n_x$ , Schwelle nach ID125																																									
3	$ M_{ist}  > M_x$ , Schwelle nach ID126																																									
4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83																																									
5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39																																									
6	In Position Fenster nach ID57																																									
7	$ P_{ist}  > P_x$ , Schwelle nach ID158																																									
8	Softwareendschalter negativ, nach ID50																																									
9	Antrieb winkelsynchron, Fenster nach ID228																																									
10	Antrieb drehzahlsynchron, Fenster nach ID32952																																									
11	$n_{ist}^3 \cdot 0$																																									
12	Quittierung Istwert geeicht																																									
13	Restweg wurde gelöscht, nach ID32922																																									
14	Überstrommeldung: Auslastung > 50% der Überlastgrenze																																									
15	Softwareendschalter positiv Fenster ID49																																									
Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage																																										
byAxisState	204D	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>REF_ACK</td> <td>Referenzpunkt bekannt</td> </tr> <tr> <td>1...7</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	REF_ACK	Referenzpunkt bekannt	1...7	-	Reserviert																										
Statusbits																																										
BIT	SYNTAX	BEDEUTUNG																																								
0	REF_ACK	Referenzpunkt bekannt																																								
1...7	-	Reserviert																																								

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
wStatusBitsId144	204E	00	Tx	Frei konfigurierbares Statuswort zur Anzeige von Echtzeitbits aus der Liste der Codes für Binärausgänge. Siehe auch Dokumentation Parameter unter ID144 „Statusbits“. Der Anzeigehalt des Statusworts ID144 wird über ID26 „Konfigurationsliste Statusbits“ festgelegt.

1) Um die Signale UE und RF über die API Variable steuern zu können, ist folgende Parametrierung notwendig:

ID32795=5 Quelle UE

ID32796=5 Quelle RF

## 6.4 Binäre Ein- und Ausgänge

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
bySystemIn	2060	00	Tx	Sendet das Prozessabbild „Binäre Eingänge“ der Eingänge BE1, BE2 und BE3 auf der Reglerkarte
bySystemOut	2070	00	Rx	Empfängt Prozessabbild „Binäre Ausgänge“ zur Ausgabe auf den Binärausgängen BA1, BA2 und BA3 der Reglerkarte <sup>1)</sup>
byInp1Byte0	6000	01	Tx	Jedes gesendete Byte stellt das Prozessabbild „Binäre Eingänge“ von 8 binären Eingängen zur Verfügung.
byInp1Byte1	6000	02	Tx	
byOutp1Byte0	6200	01	Rx	Die eingehenden Bytes als „Prozessabbild Ausgänge“ werden auf die binären Ausgänge übertragen <sup>2)</sup>
byOutp1Byte1	6200	02	Rx	

1) Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.

ID32864=0 Adresse Ausgangsport 3

2) Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.

ID32846=0 Adresse Ausgangsport 1

## 6.5 Diagnosemeldungen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
byErrorSys	204C	01	Tx	Die Auswertung der Bytes dient zur Erkennung von Fehler- bzw. Warn-meldungen. Ein Fehler im System liegt vor, wenn das Byte einen Wert ungleich Null liefert und das "System Bereit" Bit (SBM) nicht gesetzt ist. Eine Warnung im System liegt vor, wenn das Byte einen Wert ungleich Null liefert und das "System Bereit" Bit (SBM) gesetzt ist Die Auswertung kann z.B. in der AMK-SPS programmiert werden. Durch das Lesen von ID390 Diagnosenummer oder ID32840 Diagnoseliste kann die Diagnosenummer (siehe Dokumentation Diagnosemeldungen) ausgelesen werden



## 6.6 Konfigurierbare 4x32Bit Meldung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diSetpointScr1	2050	01	Tx	Inhalt 1. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948)

## 6.7 AFP Protokoll zur Antriebskommandierung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
lwAFPReadBlock	2020	01-08	Tx	8Byte AFP Statusdaten vom Antrieb
lwAFPWriteBlock	2021	01-08	Rx	8Byte AFP Steuerdaten zum Antrieb

## 6.8 Vorsteuervariablen Sollwert- und Istwertliste

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iSetpoint0*	2080	01	Rx	ID32838 Sollwertliste
iSetpoint1*	2080	02	Rx	ID32838 Sollwertliste
iSetpoint2*	2080	03	Rx	ID32838 Sollwertliste
iSetpoint3*	2080	04	Rx	ID32838 Sollwertliste
diSetpoint0*	2081	01	Rx	ID32838 Sollwertliste
diSetpoint1*	2081	02	Rx	ID32838 Sollwertliste
iActValue0*	2090	01	Tx	ID32839 Istwertliste
iActValue1*	2090	02	Tx	ID32839 Istwertliste
diActValue0*	2091	01	Tx	ID32839 Istwertliste
diActValue1*	2091	02	Tx	ID32839 Istwertliste

\* in Vorbereitung

## 7 API Variablen für die KWF Doppelfrequenzumrichter

**Hinweis:** Eine Nachricht zum KWF muss immer beide Variablen (diMainSetpoint und wDeviceControl) beinhalten. Die Übertragung nur einer Variablen ist nicht möglich..

### 7.1 Empfang von Sollwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diMainSetpoint	2030	01	Rx	Drehzahlsollwert in 1/10000 U/min

### 7.2 Status- und Steuervariablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																											
wDeviceState	2048	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SBM</td> <td>System bereit Meldung</td> </tr> <tr> <td>1 ...3</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>4</td> <td>WARN</td> <td>Warnung</td> </tr> <tr> <td>5</td> <td>ERR</td> <td>Fehler</td> </tr> <tr> <td>6 ...7</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>8</td> <td>QRF</td> <td>Quittierung Reglerfreigabe</td> </tr> <tr> <td>9 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	SBM	System bereit Meldung	1 ...3	-	Reserviert	4	WARN	Warnung	5	ERR	Fehler	6 ...7	-	Reserviert	8	QRF	Quittierung Reglerfreigabe	9 ...15	-	Reserviert
Statusbits																															
BIT	SYNTAX	BEDEUTUNG																													
0	SBM	System bereit Meldung																													
1 ...3	-	Reserviert																													
4	WARN	Warnung																													
5	ERR	Fehler																													
6 ...7	-	Reserviert																													
8	QRF	Quittierung Reglerfreigabe																													
9 ...15	-	Reserviert																													
wDeviceControl	2049	00	Rx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FL</td> <td>Fehler löschen</td> </tr> <tr> <td>1 ...7</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>8</td> <td>RF</td> <td>Reglerfreigabe</td> </tr> <tr> <td>9 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	FL	Fehler löschen	1 ...7	-	Reserviert	8	RF	Reglerfreigabe	9 ...15	-	Reserviert									
Steuerbits																															
BIT	SYNTAX	BEDEUTUNG																													
0	FL	Fehler löschen																													
1 ...7	-	Reserviert																													
8	RF	Reglerfreigabe																													
9 ...15	-	Reserviert																													

<sup>1)</sup> Um das Signal RF über die API Variable steuern zu können, ist folgende Parametrierung notwendig:  
ID32796=5 Quelle RF

## 8 API Variablen für die Kompakt-Einspeisung KE

**Hinweis:** Für das KE müssen alle 3 Sendevariablen (iMessage16, diMessage32 und wDeviceState) gemappt werden. Es ist nicht zulässig nur eine der 3 Variablen zu konfigurieren. Die PDO wird ereignisgesteuert (Übertragungstyp EVENT) übertragen. Kleinstmögliche Zykluszeit dieser Nachricht ist 10ms.

### 8.1 Senden von Istwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iMessage16	2040	01	Tx	Konfigurierbare zyklische 16Bit Istwertmeldung (ID32785) Konfigurierbare Istwerte: 32836 Zwischenkreisspannung 33101 I <sup>2</sup> t Überlast Umrichter 33116 Temperatur intern 34144 Netzspannung Effektiv 34145 Netzstrom Effektiv
diMessage32	2040	02	Tx	Konfigurierbare zyklische 32Bit Istwertmeldung (ID32786)

### 8.2 Status- und Steuervariablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																											
wDeviceState	2048	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SBM</td> <td>System bereit Meldung</td> </tr> <tr> <td>1</td> <td>QUE</td> <td>Quittierung Umrichter Ein</td> </tr> <tr> <td>2</td> <td>QFL</td> <td>Quittierung Fehler löschen</td> </tr> <tr> <td>3</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>4</td> <td>WARN</td> <td>KE Warnmeldung</td> </tr> <tr> <td>5</td> <td>ERR</td> <td>KE Fehlermeldung</td> </tr> <tr> <td>6 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	SBM	System bereit Meldung	1	QUE	Quittierung Umrichter Ein	2	QFL	Quittierung Fehler löschen	3	-	Reserviert	4	WARN	KE Warnmeldung	5	ERR	KE Fehlermeldung	6 ...15	-	Reserviert
Statusbits																															
BIT	SYNTAX	BEDEUTUNG																													
0	SBM	System bereit Meldung																													
1	QUE	Quittierung Umrichter Ein																													
2	QFL	Quittierung Fehler löschen																													
3	-	Reserviert																													
4	WARN	KE Warnmeldung																													
5	ERR	KE Fehlermeldung																													
6 ...15	-	Reserviert																													
wDeviceControl	2049	00	Rx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FL</td> <td>Fehler löschen</td> </tr> <tr> <td>1</td> <td>UE</td> <td>Umrichter Ein</td> </tr> <tr> <td>2 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	FL	Fehler löschen	1	UE	Umrichter Ein	2 ...15	-	Reserviert												
Steuerbits																															
BIT	SYNTAX	BEDEUTUNG																													
0	FL	Fehler löschen																													
1	UE	Umrichter Ein																													
2 ...15	-	Reserviert																													

## 9 API Variablen für AMKSMART IDT Motoren mit integriertem Servoregler

### 9.1 Empfang von Sollwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diSetSpeed	2030	01	Rx	32Bit Drehzahlsollwert [1/10000 U/min
iSetTorque	2030	03	Rx	16Bit Drehmomentsollwert
iSetPosition	2030	04	Rx	16Bit Lagesollwert
diSetPosition	2030	05	Rx	32Bit Lagesollwert
iSetSpeed	2030	06	Rx	16Bit Drehzahlsollwert [U/min]

### 9.2 Senden von Istwerten

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iMessage16	2040	01	Tx	Konfigurierbare zyklische 16Bit Istwertmeldung (ID32785)
diMessage32	2040	02	Tx	Konfigurierbare zyklische 32Bit Istwertmeldung (ID32786)
diActPosition	2040	03	Tx	32Bit Lageistwert

### 9.3 Status- und Steuervariablen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																					
wDeviceState	2048	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SBM</td> <td>System bereit Meldung</td> </tr> <tr> <td>1</td> <td>QUE</td> <td>Quittierung 48VDC ein</td> </tr> <tr> <td>2 ...7</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>8</td> <td>QRF</td> <td>Quittierung Regler-freigabe</td> </tr> <tr> <td>9 ...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	SBM	System bereit Meldung	1	QUE	Quittierung 48VDC ein	2 ...7	-	Reserviert	8	QRF	Quittierung Regler-freigabe	9 ...15	-	Reserviert
Statusbits																									
BIT	SYNTAX	BEDEUTUNG																							
0	SBM	System bereit Meldung																							
1	QUE	Quittierung 48VDC ein																							
2 ...7	-	Reserviert																							
8	QRF	Quittierung Regler-freigabe																							
9 ...15	-	Reserviert																							
wDeviceControl	2049	00	Rx	<table border="1"> <thead> <tr> <th colspan="3">Steuerbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FL</td> <td>Fehler löschen</td> </tr> <tr> <td>1 ...7</td> <td>-</td> <td>Reserviert</td> </tr> <tr> <td>8</td> <td>RF</td> <td>Reglerfreigabe <sup>1)</sup></td> </tr> <tr> <td>9...15</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Steuerbits			BIT	SYNTAX	BEDEUTUNG	0	FL	Fehler löschen	1 ...7	-	Reserviert	8	RF	Reglerfreigabe <sup>1)</sup>	9...15	-	Reserviert			
Steuerbits																									
BIT	SYNTAX	BEDEUTUNG																							
0	FL	Fehler löschen																							
1 ...7	-	Reserviert																							
8	RF	Reglerfreigabe <sup>1)</sup>																							
9...15	-	Reserviert																							

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung																																						
wRealTimeBits	204A	00	Tx	<table border="1"> <thead> <tr> <th colspan="2">Echtzeitbitmeldungen</th> </tr> <tr> <th>BIT</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><math> n_{soll} - n_{ist}  &lt; n_{Fenster}</math> (<math>n_{ist} = n_{soll}</math>), Fenster nach ID157</td> </tr> <tr> <td>1</td> <td><math> n_{ist}  &lt; n_{min}</math>, Schwelle nach ID124</td> </tr> <tr> <td>2</td> <td><math> n_{ist}  &lt; n_x</math>, Schwelle nach ID125</td> </tr> <tr> <td>3</td> <td><math> M_{ist}  &gt; M_x</math>, Schwelle nach ID126</td> </tr> <tr> <td>4</td> <td><math> M_{ist}  &gt; M_{grenz}</math>, Grenzen nach ID82, ID83</td> </tr> <tr> <td>5</td> <td><math> n_{soll}  &gt;  n_{grenz} </math>, Grenzen nach ID38, ID39</td> </tr> <tr> <td>6</td> <td>In Position Fenster nach ID57</td> </tr> <tr> <td>7</td> <td><math> P_{ist}  &gt; P_x</math>, Schwelle nach ID158</td> </tr> <tr> <td>8</td> <td>Softwareendschalter negativ, nach ID50</td> </tr> <tr> <td>9</td> <td>Antrieb winkelsynchron, Fenster nach ID228</td> </tr> <tr> <td>10</td> <td>Antrieb drehzahlsynchron, Fenster nach ID32952</td> </tr> <tr> <td>11</td> <td><math>n_{ist}^3 0</math></td> </tr> <tr> <td>12</td> <td>Quittierung Istwert geeicht</td> </tr> <tr> <td>13</td> <td>Restweg wurde gelöscht, nach ID32922</td> </tr> <tr> <td>14</td> <td>Überstrommeldung: Auslastung &gt; 50% der Überlastgrenze</td> </tr> <tr> <td>15</td> <td>Softwareendschalter positiv Fenster ID49</td> </tr> <tr> <td colspan="2">Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage</td> </tr> </tbody> </table>	Echtzeitbitmeldungen		BIT	BEDEUTUNG	0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157	1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124	2	$ n_{ist}  < n_x$ , Schwelle nach ID125	3	$ M_{ist}  > M_x$ , Schwelle nach ID126	4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83	5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39	6	In Position Fenster nach ID57	7	$ P_{ist}  > P_x$ , Schwelle nach ID158	8	Softwareendschalter negativ, nach ID50	9	Antrieb winkelsynchron, Fenster nach ID228	10	Antrieb drehzahlsynchron, Fenster nach ID32952	11	$n_{ist}^3 0$	12	Quittierung Istwert geeicht	13	Restweg wurde gelöscht, nach ID32922	14	Überstrommeldung: Auslastung > 50% der Überlastgrenze	15	Softwareendschalter positiv Fenster ID49	Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage	
Echtzeitbitmeldungen																																										
BIT	BEDEUTUNG																																									
0	$ n_{soll} - n_{ist}  < n_{Fenster}$ ( $n_{ist} = n_{soll}$ ), Fenster nach ID157																																									
1	$ n_{ist}  < n_{min}$ , Schwelle nach ID124																																									
2	$ n_{ist}  < n_x$ , Schwelle nach ID125																																									
3	$ M_{ist}  > M_x$ , Schwelle nach ID126																																									
4	$ M_{ist}  > M_{grenz}$ , Grenzen nach ID82, ID83																																									
5	$ n_{soll}  >  n_{grenz} $ , Grenzen nach ID38, ID39																																									
6	In Position Fenster nach ID57																																									
7	$ P_{ist}  > P_x$ , Schwelle nach ID158																																									
8	Softwareendschalter negativ, nach ID50																																									
9	Antrieb winkelsynchron, Fenster nach ID228																																									
10	Antrieb drehzahlsynchron, Fenster nach ID32952																																									
11	$n_{ist}^3 0$																																									
12	Quittierung Istwert geeicht																																									
13	Restweg wurde gelöscht, nach ID32922																																									
14	Überstrommeldung: Auslastung > 50% der Überlastgrenze																																									
15	Softwareendschalter positiv Fenster ID49																																									
Legende: N Drehzahl, M Drehmoment, P Leistung, X Lage																																										
byAxisState	204D	00	Tx	<table border="1"> <thead> <tr> <th colspan="3">Statusbits</th> </tr> <tr> <th>BIT</th> <th>SYNTAX</th> <th>BEDEUTUNG</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>REF_ACK</td> <td>Referenzpunkt bekannt</td> </tr> <tr> <td>1 ...7</td> <td>-</td> <td>Reserviert</td> </tr> </tbody> </table>	Statusbits			BIT	SYNTAX	BEDEUTUNG	0	REF_ACK	Referenzpunkt bekannt	1 ...7	-	Reserviert																										
Statusbits																																										
BIT	SYNTAX	BEDEUTUNG																																								
0	REF_ACK	Referenzpunkt bekannt																																								
1 ...7	-	Reserviert																																								
wStatusBitsId144	204E	00	Tx	Frei konfigurierbares Statuswort zur Anzeige von Echtzeitbits aus der Liste der Codes für Binärausgänge. Siehe auch Dokumentation Parameter (T.-Nr. 26249) unter ID144 „Statusbits“. Der Anzeigeinhalt des Statusworts ID144 wird über ID26 „Konfigurationsliste Statusbits“ festgelegt.																																						

1) Um die Signale UE und RF über die API Variable steuern zu können, ist folgende Parametrierung notwendig:

ID32795=5 Quelle UE

ID32796=5 Quelle RF

## 9.4 Binäre Ein- und Ausgänge

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
bySystemIn	2060	00	Tx	Sendet das Prozessabbild „Binäre Eingänge“ der Eingänge des IDTs BE1, BE2 und BE3
bySystemOut	2070	00	Rx	Empfängt Prozessabbild „Binäre Ausgänge“ zur Ausgabe auf den Binärausgängen BA1, BA2 und BA3 der Reglerkarte <sup>1)</sup>
byInp1Byte0	6000	01	Tx	Das Prozessabbild (1 Byte) „Binäre Eingänge“ wird auf den Binäreingangsport 1 kopiert (ID32873-ID32881)
byOutp1Byte0	6200	01	Rx	Die eingehenden Bytes als „Prozessabbild Ausgänge“ werden auf die binären Ausgänge übertragen (z.B. E/A Optionskarte KU- oder KW-EA2 im Steckplatz 1 A1...A8 werden durch die Variable byOutp1Byte0 gesteuert <sup>2)</sup>

<sup>1)</sup> Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.  
ID32864=0 Adresse Ausgangsport 3

<sup>2)</sup> Binärausgänge können über die API Variable gesteuert werden, wenn sie nicht dem Grundsystem zugewiesen sind. Folgende Parametrierung ist notwendig.  
ID32846=0 Adresse Ausgangsport 1

## 9.5 Diagnosemeldungen

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
byErrorSys	204C	01	Tx	Die Auswertung der Variable dient zur Erkennung von Fehler- bzw. Warn-meldungen. Ein Fehler im System liegt vor, wenn das Byte einen Wert ungleich Null liefert und die "System Bereit Meldung" SBM nicht gesetzt ist. Eine Warnung liegt vor, wenn das Byte einen Wert ungleich Null liefert und die "System Bereit Meldung" SBM gesetzt ist. Die Auswertung kann z.B. in der AMK-SPS programmiert werden. Durch das Lesen von ID390 Diagnosenummer oder ID32840 Diagnosenliste kann die Diagnosenummer (siehe Dokumentation Diagnosemeldungen) ausgelesen werden

## 9.6 Konfigurierbare 4x32Bit Meldung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
diSetpointScr1	2050	01	Tx	Inhalt 1. Meldung aus der konfigurierbaren 4x32Bit Meldung (ID32948) (Es wird nur der Zählerwert (High word) übertragen, da der IDT Rechteckeingang keinen Nullimpuls hat)

## 9.7 AFP Protokoll zur Antriebskommandierung

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
lwAFPReadBlock	2020	01-08	Tx	8Byte AFP Statusdaten
lwAFPWriteBlock	2021	01-08	Rx	8Byte AFP Steuerdaten zur Antriebskommandierung über das AMK Feldbus Protokoll (siehe Dokumentation AFP T.-Nr.:27872)

## 9.8 Vorsteuervariablen Sollwert- und Istwertliste

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung
iSetpoint0	2080	01	Rx	ID32838 Sollwertliste
iSetpoint1	2080	02	Rx	ID32838 Sollwertliste
iSetpoint2	2080	03	Rx	ID32838 Sollwertliste
iSetpoint3*	2080	04	Rx	ID32838 Sollwertliste
diSetpoint0*	2081	01	Rx	ID32838 Sollwertliste
diSetpoint1*	2081	02	Rx	ID32838 Sollwertliste
iActValue0*	2090	01	Tx	ID32839 Istwertliste
iActValue1*	2090	02	Tx	ID32839 Istwertliste
diActValue0	2091	01	Tx	ID32839 Istwertliste
diActValue1	2091	02	Tx	ID32839 Istwertliste

\*) In Vorbereitung

## 10 API Variablen für die AMKAMAC Steuerungen AS-PL12, AS-PL14 und AS-C

Feldbusvariablen werden von der SPS benutzt, um Daten über Feldbus mit anderen SPS Baugruppen oder Antrieben austauschen zu können. SPS Programme schreiben Werte in die Sendevariablen und interpretieren Empfangsdaten.

API Variablenname	CAN Index [hex]	CAN Sub-Index [hex]	Kopier-richtung	Verwendung								
lwInX lwOutX	1) <sup>1)</sup>		Rx Tx	Longwort asynchron (64Bit) <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [LWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...63</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...63</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...63</td> </tr> </tbody> </table>	Hardware	Index X [LWORD]	AS-PL12 + AS-FCT1	0 ...63	AS-PL14 + ASFCT2	0 ...63	AS-C + AS-FCT1	0 ...63
Hardware	Index X [LWORD]											
AS-PL12 + AS-FCT1	0 ...63											
AS-PL14 + ASFCT2	0 ...63											
AS-C + AS-FCT1	0 ...63											
dwInX dwOutX	2000 200C	01+X 01+X	Rx Tx	Doppelwort asynchron (32Bit) <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...127</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...127</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...127</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...127	AS-PL14 + ASFCT2	0 ...127	AS-C + AS-FCT1	0 ...127
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...127											
AS-PL14 + ASFCT2	0 ...127											
AS-C + AS-FCT1	0 ...127											
wInX wOutX	2001 200D	01+X 01+X	Rx Tx	Wort asynchron (16Bit) <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...254</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...254</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...254</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...254	AS-PL14 + ASFCT2	0 ...254	AS-C + AS-FCT1	0 ...254
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...254											
AS-PL14 + ASFCT2	0 ...254											
AS-C + AS-FCT1	0 ...254											
byInX byOutX	2002 2006 200E 2012	01+X X-254 01+X X-254	Rx Rx Tx Tx	Byte asynchron (8Bit) <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...254, 255... 509</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...254, 255... 509</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...254, 255... 509</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...254, 255... 509	AS-PL14 + ASFCT2	0 ...254, 255... 509	AS-C + AS-FCT1	0 ...254, 255... 509
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...254, 255... 509											
AS-PL14 + ASFCT2	0 ...254, 255... 509											
AS-C + AS-FCT1	0 ...254, 255... 509											
lwSynclnX lwSyncOutX	1) <sup>1)</sup>		Rx Tx	Longwort synchron (64Bit) <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...31</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...63</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...15</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...31	AS-PL14 + ASFCT2	0 ...63	AS-C + AS-FCT1	0 ...15
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...31											
AS-PL14 + ASFCT2	0 ...63											
AS-C + AS-FCT1	0 ...15											
dwSynclnX dwSyncOutX	2003 200F	01+X 01+X	Rx Tx	Doppelwort synchron <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...63</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...127</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...31</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...63	AS-PL14 + ASFCT2	0 ...127	AS-C + AS-FCT1	0 ...31
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...63											
AS-PL14 + ASFCT2	0 ...127											
AS-C + AS-FCT1	0 ...31											
wSynclnX wSyncOutX	2004 2010	01+X 01+X	Rx Tx	Wort synchron <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [DWORD]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...127</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...254</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...63</td> </tr> </tbody> </table>	Hardware	Index X [DWORD]	AS-PL12 + AS-FCT1	0 ...127	AS-PL14 + ASFCT2	0 ...254	AS-C + AS-FCT1	0 ...63
Hardware	Index X [DWORD]											
AS-PL12 + AS-FCT1	0 ...127											
AS-PL14 + ASFCT2	0 ...254											
AS-C + AS-FCT1	0 ...63											
bySynclnX bySyncOutX	2005 2007 2011 2013 *	01+X X-254 01+X X-254	Rx Rx Tx Tx	Byte synchron <table border="1"> <thead> <tr> <th>Hardware</th> <th>Index X [BYTE]</th> </tr> </thead> <tbody> <tr> <td>AS-PL12 + AS-FCT1</td> <td>0 ...254, *255... 509</td> </tr> <tr> <td>AS-PL14 + ASFCT2</td> <td>0 ...254, 255... 509</td> </tr> <tr> <td>AS-C + AS-FCT1</td> <td>0 ...127</td> </tr> </tbody> </table>	Hardware	Index X [BYTE]	AS-PL12 + AS-FCT1	0 ...254, *255... 509	AS-PL14 + ASFCT2	0 ...254, 255... 509	AS-C + AS-FCT1	0 ...127
Hardware	Index X [BYTE]											
AS-PL12 + AS-FCT1	0 ...254, *255... 509											
AS-PL14 + ASFCT2	0 ...254, 255... 509											
AS-C + AS-FCT1	0 ...127											

1)

lw (Longwort) wird AMK intern aus Bytes zusammengesetztà Byte mapping z.B. lwOut10 beginnt mit byOut80

\*

Nicht unterstützt

## Ihre Meinung zählt!

Mit unseren Dokumentationen möchten wir Sie im Umgang mit den AMKmotion Produkten bestmöglich unterstützen.

Daher sind wir ständig bestrebt, unsere Dokumentationen zu optimieren.

Ihre Kommentare oder Anregungen sind für uns immer interessant.

Nehmen Sie sich kurz Zeit und beantworten Sie unsere Fragen. Bitte schicken Sie anschließend eine Kopie dieser Seite an AMKmotion zurück.



E-Mail: [Documentation@amk-motion.com](mailto:Documentation@amk-motion.com)

oder

Fax-Nr.: +49 7021/50 05-199

### Vielen Dank für Ihre Mithilfe.

#### Ihr AMKmotion Dokumentationsteam

1. Wie sind Sie mit der Optik unserer Dokumentationen zufrieden?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

2. Ist der Inhalt gut gegliedert?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

3. Ist der Inhalt verständlich dokumentiert?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

4. Haben Sie Themen in der Dokumentation vermisst?

(1) nein (2) ja, welche:

5. Fühlen Sie sich bei AMKmotion insgesamt gut betreut?

(1) sehr gut (2) gut (3) mäßig (4) kaum (5) nicht

AMKmotion GmbH + Co KG

Telefon: +49 7021/50 05-0, Telefax: +49 7021/50 05-199

E-Mail: [info@amk-motion.com](mailto:info@amk-motion.com)

Homepage: [www.amk-motion.com](http://www.amk-motion.com)