



MONTAGE- UND BETRIEBSANLEITUNG IF SMI MODBUS MS / TP

IF SMI MODBUS MS / TP



Dieses Handbuch ist für folgende Typen geeignet:

- IF SMI Modbus MS / TP 230VAC (01092130)
- IF SMI Modbus MS / TP 24VDC (01092131)

Dokumentnummer: 85900625 A1

Motorsteuerung SMI
Modbus für Hutschiene
zur Steuerung von 16
SMI-Motoren 230 VAC
oder 24VDC.

Lesen Sie das Handbuch, bevor Sie mit der Installation beginnen.
Die Nichtbeachtung der Anweisungen kann zu Mängeln führen,
die nicht durch die Garantie abgedeckt sind.
Druckfehler und technische Änderungen vorbehalten.



INHALT

■ Sicherheitshinweise	2	■ Modbus Implementierung	16
■ Technische Daten	3	■ Register-Überblick	30
■ Anschlusspläne	4		
■ Was ist SMI	8		
■ Hardware	8		
■ Menüführung	10		
■ Moduswahl	15		
(RS485 Komm.-Protokoll / Modbus-Protokoll)			

SICHERHEITSHINWEISE

ALLGEMEINE HINWEISE

Diese Sicherheitshinweise sind Bestandteil des Produkts und müssen vor Montage, elektrischem Anschluss, Inbetriebnahme und Betrieb vollständig gelesen und verstanden werden.

- Das Produkt ist ausschließlich für den bestimmungsgemäßen Gebrauch vorgesehen.
- Installation, Anschluss, Inbetriebnahme und Wartung dürfen ausschließlich durch eine qualifizierte Elektrofachkraft erfolgen.
- Alle geltenden gesetzlichen Vorschriften, Normen und Richtlinien sind einzuhalten, insbesondere DIN VDE 0100, DIN EN 60204-1, DIN EN 82079-1 sowie die Vorschriften der örtlichen Energieversorgungsunternehmen.
- Das Gerät ist vor der Montage auf Beschädigungen zu prüfen. Bei festgestellten Schäden darf das Gerät nicht in Betrieb genommen werden.
- Änderungen oder Modifikationen am Gerät sind unzulässig und führen zum Erlöschen aller Gewährleistungs- und Haftungsansprüche.
- Der Betreiber hat sicherzustellen, dass diese Sicherheitshinweise allen nachfolgenden Nutzern zugänglich sind.

KRITISCHE WARNHINWEISE



GEFAHREN DURCH ELEKTRISCHEN STROM | GEFAHR – Lebensgefahr durch elektrischen Schlag.

Vor Montage und Arbeiten am Gerät Netzspannung vollständig abschalten.



GEFAHREN DURCH FEHLKONFIGURATION DES MODBUS-BUSSES | VORSICHT – Sachschäden und Funktionsstörungen durch falsche Bus-Topologie.



GEFAHREN DURCH UNBEABSICHTIGTE MOTORBEWEGUNGEN | WARNUNG – Verletzungsgefahr durch unkontrollierte Bewegungen der angeschlossenen Antriebe.



HINWEIS
Die vollständigen Sicherheitsanweisungen finden Sie unter: www.vestamatic.com/safety



SCAN ME


SUPPORT/KONTAKT

Vestamatic International GmbH
Am Tannenbaum 2 | 41066 Mönchengladbach
E-Mail: info@vestamatic.com

TECHNISCHE DATEN

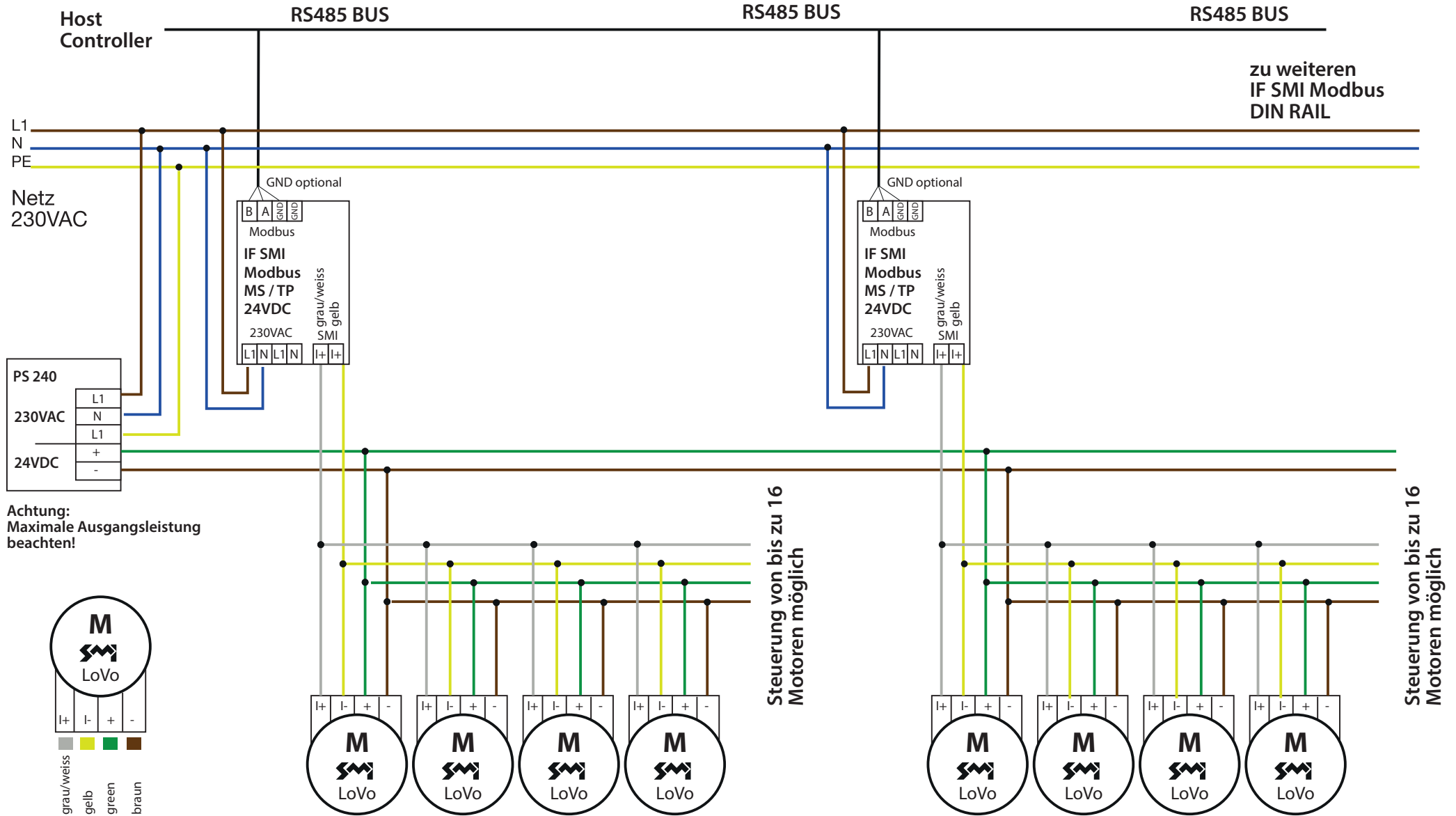
KURZBESCHREIBUNG

- Motorsteuerung zur Montage auf einer Hutschiene
- Zur Steuerung von 16 SMI-Motoren - entweder als 230VAC oder als 24VDC (ein Mix am selben Gerät ist nicht möglich)
- Umsetzung von Modbus auf SMI-Befehle
- Test-Bedienung aller angeschlossenen SMI-Motoren mit den AUF/AB-Tasten am Modul
- Display am Gerät erlaubt einfache Navigation zwischen den Motoren und visualisiert Status
- Konfiguration der Slave Adressen der Motoren über Modbus
- Vollständige Konfiguration sämtlicher Standard-SMI-Befehlssätze
- Zusätzlich herstellerspezifische Befehlssätze in Kombination mit Brel Vestamatic Group Antrieben

PARAMETER	WERT
Geräteart	Umsetzer SMI – Modbus
Betriebsspannung	230 VAC
Bemessungs-Stoßspannung	2,5 kV
Leistungsaufnahme	2 W
Schutzart	IP 20
Schutzklasse	II
Abmessungen (H × B × T)	90mm × 52mm × 60mm (3 TE)
Montageart	Hutschiene
Anwendungsbereich	Innenbereich, Schaltschrank
Betriebstemperaturbereich	0 - 40 °C / 32 - 104 °F
Verschmutzungsgrad	2
Gehäuse	(Hellgrau) UL94-V0 flammhemmendes Polycarbonat
Konformität	

ANSCHLUSSPLAN

IF SMI MODBUS 24VDC



WAS IST SMI

- SMI ist die Abkürzung für Standard Motor Interface.
- SMI wurde entwickelt, um intelligente Antriebe mit Rollläden und Sonnenschutzeinrichtungen zu verbinden.
- SMI ermöglicht die Übertragung von Meldungen vom Steuerungssystem an den Antrieb und umgekehrt.
- Durch SMI können Produkte verschiedener Hersteller miteinander kombiniert werden.
- Die SMI-Schnittstelle vereinfacht den Einsatz hochwertiger Lösungen und erhöht die Kompatibilität zwischen handelsüblichen Antrieben und Steuerungen verschiedener Hersteller.
- Anwendungen für Rollläden und Sonnenschutzeinrichtungen erfordern größte Robustheit und Wirtschaftlichkeit. SMI wurde entwickelt, um diese Anforderungen zu erfüllen.

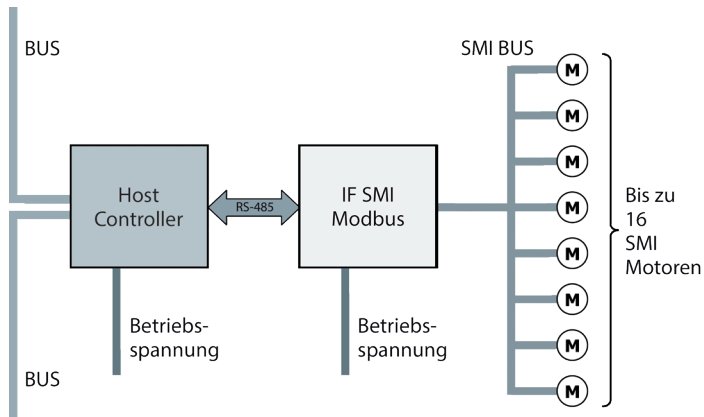
HARDWARE

Die Motorsteuerung IF SMI Modbus MS / TP kann für SMI (230VAC) und SMI LoVo (24VDC) Anwendungen genutzt werden.



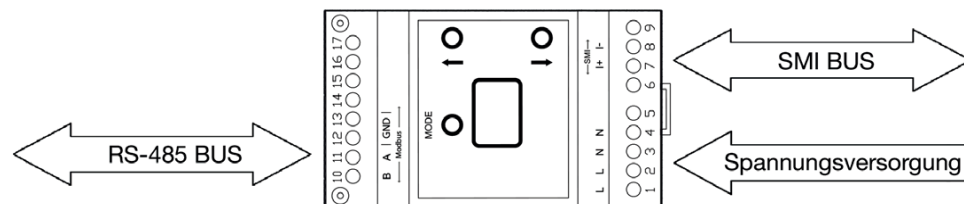
HINWEIS

Die kombinierte Nutzung von SMI (230VAC) und SMI LoVo (24VDC) auf dem selben SMI BUS ist nicht gestattet.



ÜBERSICHT IF SMI MODBUS

Das IF SMI Modbus ist ein intelligentes Modul, das Modbus- Befehle in SMI-Befehle umsetzt.



HARDWARE

ÜBERSICHT IF SMI MODBUS

MODBUS BEDIENOBERFLÄCHE

MODE x 1



Anzeige verschiedener Kommunikationsparameter für 5 Sek

MODE x 3s



Öffnet Ändern der Basis-Adresse und weiterer Parameter

AUF/AB x 1



Im Hauptmenü: Alle SMI Motoren aufwärts/abwärts fahren.
Im Adressmenü: Aktuelle Basis-Adresse erhöhen/senken
Im Fehlerprotokoll: Nächsten/vorherigen Fehler zeigen

RS-485 BUS

RS-485 ist der Kommunikationsbus zwischen dem Host Controller und dem IF SMI Modbus. Nachstehende Konfiguration wird verwendet:

- Baud-Rate: 19200
- Datenbits: 8
- Stoppbits: 1
- Parität: Gerade
- Signal: - 7V bis +10V Gleichtakt-Eingangsspannungsbereich
- Maximal zulässiger Abstand Byte-to-Byte: 5 Millisekunden

SPANNUNGSVERSORGUNG

Das IF SMI Modbus benötigt folgende Spannungsversorgungssignale:

- L, N (230VAC/50Hz)

SMI BUS

Der SMI BUS unterstützt folgende Signale:

- I+ (SMI BUS)
- I- (SMI BUS)

MASTER AUF/AB-TASTEN

Das Modul IF SMI RS-485/Modbus besitzt jeweils eine AUF- und AB-Taste zur gleichzeitigen Bedienung aller angeschlossenen SMI-Motoren.

BASIS-ADRESSE

Das IF SMI Modbus wird über seine Basis-Adresse angewählt. Jedes an einen gemeinsamen RS-485 BUS angeschlossene IF SMI Modbus muss über eine einmalige Basis-Adresse verfügen. Maximal 50 IF SMI Modbus Geräte können an einen RS-485 BUS angeschlossen werden.

HARDWARE

TERMINIERUNG RS-485 BUS (OPTIONAL)

Wenn sich Kommunikationsprobleme ergeben, lesen Sie bitte den nachfolgenden Abschnitt. Im Gerät ist eine in der Regel ausreichende Basis-Terminierung vorgesehen. In Ausnahmefällen (bspw. bei langen oder unverdrillten Leitungen) empfehlen wir an beiden Enden der Busleitung einen Abschlusswiderstand von $120\ \Omega$ einzusetzen. Dieser ist zwischen den Klemmen A und B der Signalleitung anzubringen. Ein Abschlusswiderstand ist ein am äußersten Ende bzw. an Leitungsenden angebrachter Widerstand. Der Abschlusswiderstand hat idealerweise den selben Wert wie der charakteristische Leitungswiderstand.

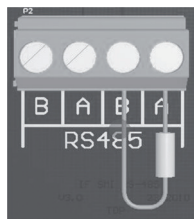


Korrekte Buserminierung $R_T = Z_0$
 $R_T = 120\ \Omega$

2-Draht-Busleitung mit einem charakteristischen Leitungswiderstand von Z_0
Example: $Z_0 = 120\ \Omega$

Korrekte Buserminierung $R_T = Z_0$
 $R_T = 120\ \Omega$

- In der Regel sind Abschlusswiderstände an beiden Leitungsenden einzusetzen.
- Obwohl eine korrekte Buserminierung an beiden Leitungsenden für die meisten Systeme unerlässlich ist, kann in Ausnahmefällen eine Buserminierung ausreichend sein.
- Diese Ausnahme betrifft ein System mit einem einzigen Sender, der am Ende der Busleitung angebracht ist. In diesem Fall ist das Einsetzen eines Abschlusswiderstands am Leitungsende – an dem sich der Sender befindet – überflüssig, weil es sich an diesem Leitungsende stets um ein ausgehendes Signal handelt.



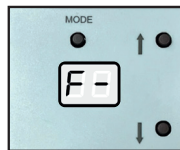
Für jedes IF SMI Modbus ist ein Abschlusswiderstand von $120\ \Omega$ vorgesehen. Wird ein Abschlusswiderstand empfohlen, ist dieser zwischen den Signalleitungen A und B des Steckers an der Leiterplatte einzusetzen (Bild nur als Referenz).

MENÜFÜHRUNG

START

Während der Start wird die Firmware-Version in 3 Schritten angezeigt.

Schritt 1



Das Display zeigt 2 Sekunden lang „F-“ als Hinweis, dass die Firmware-Version folgt.

Schritt 2

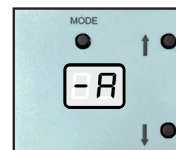


Das Display zeigt die Firmware-Version (Hauptversion) für 3 Sekunden an.

MENÜFÜHRUNG

START

Schritt 3



Das Display zeigt die Firmware-Version für 2 Sekunden an.

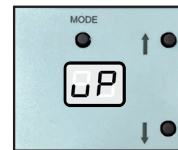
HAUPTMENÜ

Das Hauptmenü ist das Standardmenü, das angezeigt wird, wenn keine anderen Benutzeraktionen stattfinden. Das Hauptmenü zeigt:

- Ein Zähler - zählt die Anzahl der korrekt empfangenen Modbus-Befehle von 0 bis 99 und beginnt dann wieder bei 0.
- Ein blinkender Punkt in der rechten unteren Ecke mit einer Blinkfrequenz von 1 Hz, um eine laufende Anwendung anzuzeigen.

MENÜFÜHRUNG

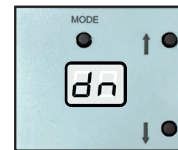
AUF x 1



Hauptmenü

Oder

AUF x 1



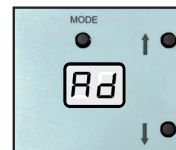
alle angeschlossenen SMI-Motoren fahren nach oben oder unten

Diese Funktion ist bei der Inbetriebnahme zur Überprüfung hilfreich:

- ob die SMI-Kommunikation funktioniert
- ob alle angeschlossenen Motoren in die richtige Richtung fahren
- ob die Endlagen richtig eingestellt sind

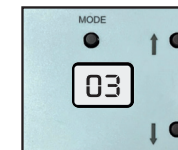
ADRESS-MENÜ

Mode halten



„Ad“ wird auf dem Display angezeigt.

Mode loslassen

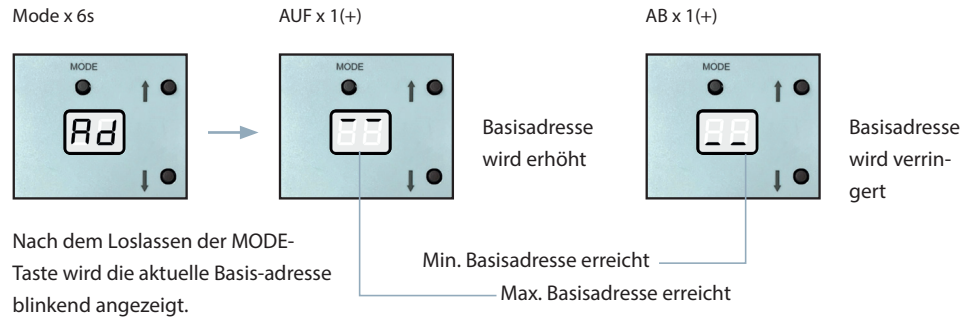


aktuelle Modbus-Basisadresse wird für 5 Sekunden angezeigt.

Zurück zum Hauptmenü: MODE-Taste im letzten Menüpunkt erneut drücken. (gilt für alle Menüeinträge)

MENÜFÜHRUNG

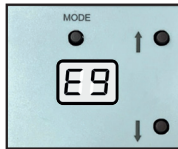
BASIS-ADRESSE ÄNDERN



i HINWEIS

Die Modbus-Basisadresse kann von 1 bis 247 geändert werden. Bei Erreichen der max-/min. Basisadresse wird das im Display über die oberen/untern Striche angezeigt.

FEHLER ANZEIGEN



Wenn ein Fehler auftritt, wird dieser direkt angezeigt.

i HINWEIS

Die Fehlermeldung wird für 5 Sekunden angezeigt.

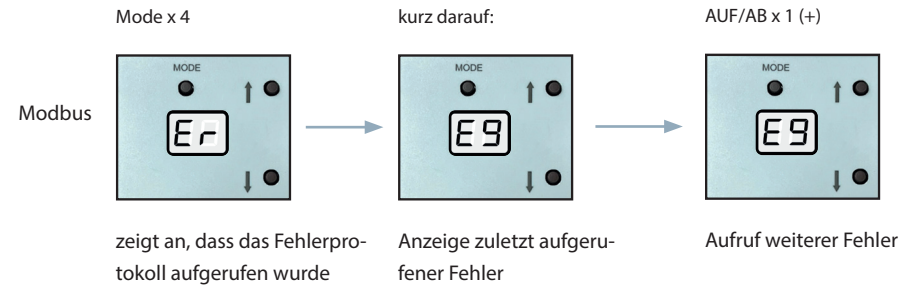
BESCHREIBUNG FEHLERCODES

E0 Kein Error	E5 RS-485 Befehlsfehler (nicht unterst./ungült. Länge)	EA SMI Datenüberlauf
E1 RS-485 Framing Error	E6 RS-485 beschäftigt (kann keinen neuen Befehl verarbeiten)	EB SMI Echo Error
E2 RS-485 Zeitüberschreitung	E7 SMI Format Error	EC SMI Warteschlange voll
E3 RS-485 Datenüberlauf	E8 SMI Checksummen Error	ED SMI Blockanzahl-Fehler (nur Modbus)
E4 RS-485 CRC Error	E9 SMI Zeitüberschreitung	EE Modbus Paritätsfehler (nur Modbus)

MENÜFÜHRUNG

MENÜ FEHLERPROTOKOLL

Das Fehlerprotokoll verfolgt die 5 zuletzt aufgetretenen Fehler und wird folgendermaßen aufgerufen:



i HINWEIS

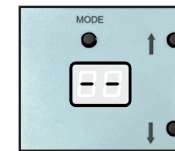
Zeigt bei gedrückter AB-Taste an, dass der erste Fehler dargestellt wird.

Zeigt bei gedrückter AUF-Taste an, dass der letzte Fehler dargestellt wird.

NEUSTART

Das IF SMI Modbus MS / TP kann ohne Unterbrechung der Stromversorgung neu gestartet werden.

Mode x 10s



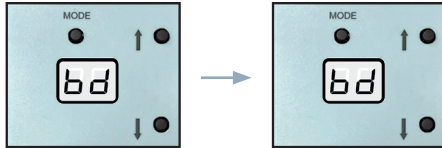
Neustart wird ausgeführt

MENÜFÜHRUNG

BAUDRATE (NUR BEI MODBUS VARIANTEN)

Mode x 5s plus
Mode x 1

AUF/AB x 1(+)



Unterstützte Baudraten

- 19200 Baud: Display zeigt,19'
- 9600 Baud: Display zeigt,9.6'
- 4800 Baud: Display zeigt,4.8'

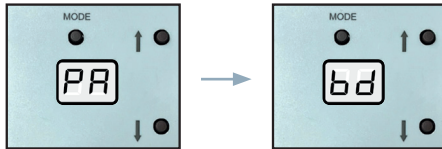
bd blinkt für 0,5 Sek.
Danach Anzeige aktuelle
Baudrate

Baudrate wird erhöht
oder verringert

PARITÄT (NUR BEI MODBUS VARIANTEN)

Mode x 5s plus
Mode x 2

AUF/AB x 1(+)



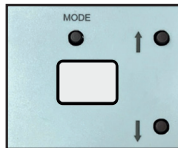
- Keine Parität: Display zeigt, n'
- Ungerade Parität: Display zeigt, o'
- Gerade Parität: Display zeigt, E'

PA blinkt für 0,5 Sek.
Danach Anzeige aktuelle
Parität

Änderung Paritätsmodus

AUTO-ADRESSIERUNG | NEUADRESSIERUNG SMI-MOTOREN

AUF/AB gleich-
zeitig x 10s



Darstellung Animation
und Anzeige Anzahl der
gefundenen Motoren

MODUSAUSWAHL

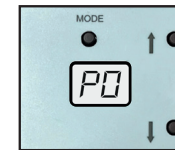
Das IF SMI Modbus MS / TP unterstützt zwei Betriebsmodi auf der RS485-Schnittstelle:

- Type 1: Für **Modbus RTU** Protokoll **P1** einstellen
- Type 2: Für **RS485** Kommunikationsprotokoll **P0** einstellen

Beide Protokollvarianten sind integriert. **Die Auswahl des gewünschten Modus sowie die Anzeige des aktuellen Modus sind nur während des Startvorgangs möglich.** Nach der Änderung des Modus startet der IF SMI Modbus MS / TP automatisch neu, um die neue Einstellung zu übernehmen.

AKTUELLEN MODUS ANZEIGEN

Beim einschalten: Mode x 6s

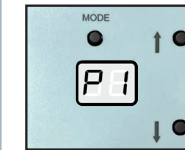


- P0 = RS485 Kommunikationsprotokoll
- P1 = Modbus-Protokoll

Display zeigt den
aktuellen Modus an

WECHSEL ZU MODUS TYPE 1 P1

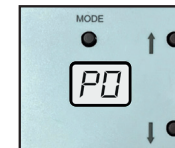
Beim einschalten: Mode + AB x 6s



Display zeigt = P1
Modbus-Protokoll

WECHSEL ZU MODUS TYPE 2 P0

Beim einschalten: Mode + AUF x 6s



Display zeigt = P0 RS485
Kommunikationsprotokoll

MODBUS IMPLEMENTIERUNG

Dieses Kapitel behandelt die Implementierung des Modbus Kommunikationsprotokoll.
Es bezieht sich somit auf die Varianten IF SMI Modbus MS / TP 230VAC / 24 VDC

1. INFORMATION AND STATUS

1.1 SOFTWARE VERSION

Read software version of the application

1.1.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

1.1.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30001-30002	0x0000-0x0001	VERSION	Software version of the application	Bit 31-24: alphanumeric char to identify hardware platform Bit 23-16: major software version Bit 15-8: minor software version Bit 7-0: alphanumeric char to identify internal revision

1.1.3 LAYOUT

Bit 31–24	Bit 23–16	Bit 15–8	Bit 7–0
Register 30001 / Bit 15–8	Register 30001 / Bit 7–0	Register 30002 / Bit 15–8	Register 30002 / Bit 7–0
Hardware platform	Major software version	Minor software version	Internal revision

1.2 GET GENERAL STATE

Read general state of SMI drives.

1.2.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

1.2.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30003	0x0002	PRESENT	SMI drive 0..15 present state	Bit n set: SMI drive n is present Bit n clear: not in use
30004	0x0003	READY	SMI drive 0..15 ready state	Bit n set: SMI drive n ready (idle) Bit n clear: SMI drive n busy (running)

1.3 GET ERROR LOG

Retrieve error log.

MODBUS IMPLEMENTIERUNG

1.3 GET ERROR LOG

Retrieve error log.

1.3.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

1.3.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30005	0x0004	ERR_0	Most recent error in log	Bit 7–0: Error Bit 15–8: not used
30006	0x0005	ERR_1	Error log entry	Bit 7–0: Error
30007	0x0006	ERR_2	Error log entry	Bit 7–0: Error
30008	0x0007	ERR_3	Error log entry	Bit 7–0: Error
30009	0x0008	ERR_4	Oldest error in log	Bit 7–0: Error

1.3.3 ERROR CODE DESCRIPTION

Please refer to section „Fehler anzeigen“.

1.4 GET POSITION

Read position of SMI drive(s).

1.4.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

1.4.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30017	0x0010	POS_0	SMI drive 0 position	SMI drive position 0x0000 = upper limit 0xFFFF = lower limit
30018	0x0011	POS_1	SMI drive 1 position	SMI drive position
:	:	:	:	:
30032	0x001F	POS_15	SMI drive 15 position	SMI drive position

1.4.3 REMARKS

Actual SMI drive position will be refreshed when drive is moving. Interval between position updates depends on total number of moving drives and can take maximum up to 3.5 seconds.

MODBUS IMPLEMENTIERUNG

1.5 GET STATUS AND TILT

Read status and tilt angle of SMI drive(s).

1.5.1 COMMAND

Accessible with Read Inputs Registers (0x04) command

1.5.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30033	0x0020	STAT_TILT_0	SMI drive 0 status and tilt	Bit 7–0: Status Bit 15–8: Tilt position
30034	0x0021	STAT_TILT_1	SMI drive 1 status and tilt	Bit 7–0: Status Bit 15–8: Tilt position
:	:	:	:	:
30048	0x002F	STAT_TILT_15	SMI drive 15 status and tilt	Bit 7–0: Status Bit 15–8: Tilt position

1.5.3 REMARKS

Status information

Status Bitbelegung								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bedeutung
X	X	X	X	0	X	X	X	Motor error occurred
X	X	X	X	1	0	0	0	UP + DOWN + STOP
X	X	X	X	1	0	0	1	DOWN + STOP
X	X	X	X	1	0	1	0	UP + STOP
X	X	X	X	1	0	1	1	ALL STOP
X	X	X	X	1	1	0	0	UP + DOWN
X	X	X	X	1	1	0	1	ALL DOWN
X	X	X	X	1	1	1	1	ALL UP
0	0	0	1	X	X	X	X	Not valid
1	1	1	1	X	X	X	X	Invalid SMI response

- Tilt information only supported for Domain 6 drives (Vestamatic) with tilt capability. For none D6 drives, this field is always zero (0).
- Actual tilt position will be refreshed when drive is moving. Interval between tilt position updates depends on total number of moving drives and can take maximum up to 3.5 seconds.

MODBUS IMPLEMENTIERUNG

1.6 GET MANUFACTURER ID

Read manufacturer ID and type of SMI drive(s).

1.6.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

1.6.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30049	0x0030	MAN_ID_0	SMI drive 0 manufacturer ID and type	Bit 7–0: Man-ID Bit 15–8: Drive type
30050	0x0031	MAN_ID_1	SMI drive 1 manufacturer ID and type	Bit 7–0: Man-ID Bit 15–8: Drive type
:	:	:	:	:
30064	0x003F	MAN_ID_15	SMI drive 15 manufacturer ID and type	Bit 7–0: Man-ID Bit 15–8: Drive type

1.7 GET KEY-ID

Read the 4 byte Key-ID of SMI drive(s).

1.7.1 COMMAND

Accessible with Read Inputs Registers (0x04) command

1.7.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30065–30066	0x0040–0x0041	KEY_ID_0	SMI drive 0 Key-ID	Bit 31–0: Key-ID
30067–30068	0x0042–0x0043	KEY_ID_1	SMI drive 1 Key-ID	Bit 31–0: Key-ID
:	:	:	:	:
30095–30096	0x005E–0x005F	KEY_ID_15	SMI drive 15 Key-ID	Bit 31–0: Key-ID

1.7.3 LAYOUT

Bit 31–16	Bit 15–0
Register 30065 / Bit 15–0	Register 30066 / Bit 15–0
SMI drive 0 Key-ID / Bit 31–16	SMI drive 0 Key-ID / Bit 15–0

1.8 FORCE INFORMATION UPDATE

Set mask of drives of which information needs to be updated:

POSITION AND TILT Normally, when drives move the position and tilt information is updated automatically.

Setting this register, the position and tilt information update can be forced.

POS1

Update (read) POS1 information from selected drives. Normally this information is retrieved from drive during start-up of the SMI gateway.

POS2

Update (read) POS2 information from selected drives.

MODBUS IMPLEMENTIERUNG

1.8.1 COMMAND

Accessible with Write Single Register (0x06), Write Multiple Registers (0x10) or Read Holding Registers (0x03) command.

1.8.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40012	0x000B	POS_TILT_UPD	Update position and tilt info mask	Bit 15–0: Mask
40013	0x000C	POS1_UPD	Update POS1 info mask	Bit 15–0: Mask
40014	0x000D	POS2_UPD	Update POS2 info mask	Bit 15–0: Mask

1.8.3 REMARKS

- Update position and tilt information will be done directly.
- Update POS1/POS2 information – when update requested - will be done once every minute.

2 CONTROL AND STEER

2.1 UP MOVEMENT

Start an UP movement for selected drive(s).

2.1.1 COMMAND

Accessible with Write Multiple Coils (0x0F) or Write Single Coil (0x05) command.

2.1.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
1	0x0000	UP_0	SMI drive 0 UP	1 = UP / 0 = no action
2	0x0001	UP_1	SMI drive 1 UP	1 = UP / 0 = no action
:	:	:	:	:
16	0x000F	UP_15	SMI drive 15 UP	1 = UP / 0 = no action

2.2 DOWN MOVEMENT

Start an DOWN movement for selected drive(s).

2.2.1 COMMAND

Accessible with Write Multiple Coils (0x0F) or Write Single Coil (0x05) command.

MODBUS IMPLEMENTIERUNG

2.2.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
17	0x0010	DOWN_0	SMI drive 0 DOWN	1 = DOWN / 0 = no action
18	0x0011	DOWN_1	SMI drive 1 DOWN	1 = DOWN / 0 = no action
:	:	:	:	:
32	0x001F	DOWN_15	SMI drive 15 DOWN	1 = DOWN / 0 = no action

2.3 STOP MOVEMENT

STOP movement for selected drive(s).

2.3.1 COMMAND

Accessible with Write Multiple Coils (0x0F) or Write Single Coil (0x05) command.

2.3.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
33	0x0020	STOP_0	SMI drive 0 STOP	1 = STOP / 0 = no action
34	0x0021	STOP_1	SMI drive 1 STOP	1 = STOP / 0 = no action
:	:	:	:	:
48	0x002F	STOP_15	SMI drive 15 STOP	1 = STOP / 0 = no action

2.4 GOTO POS1 MOVEMENT

Go to intermediate position 1 for selected drive(s).

2.4.1 COMMAND

Accessible with Write Multiple Coils (0x0F) or Write Single Coil (0x05) command.

2.3.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
49	0x0030	GOTO_POS1_0	SMI drive 0 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
50	0x0031	GOTO_POS1_1	SMI drive 1 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
:	:	:	:	:
64	0x003F	GOTO_POS1_15	SMI drive 15 GOTO_POS1	1 = GOTO_POS1 / 0 = no action

MODBUS IMPLEMENTIERUNG

2.5 GOTO POS2 MOVEMENT

Go to intermediate position 2 for selected drive(s).

2.5.1 COMMAND

Accessible with Write Multiple Coils (0x0F) or Write Single Coil (0x05) command.

2.5.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
65	0x0040	GOTO_POS2_0	SMI drive 0 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
66	0x0041	GOTO_POS2_1	SMI drive 1 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
:	:	:	:	:
80	0x004F	GOTO_POS2_15	SMI drive 15 GOTO_POS2	1 = GOTO_POS2 / 0 = no action

2.6 STEP MOVEMENT

Start a relative STEP movement in UP and/or DOWN direction for selected drive(s). A single STEP is defined as 2° rotation of the drive output shaft in given direction.

2.6.1 COMMAND

Accessible with Write Single Register (0x06) or Write Multiple Registers (0x10) command.

2.6.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40001-40002	0x0000-0x0001	STEP	SMI drive STEP UP/DOWN	Bit 15-0: Mask Bit 23-16: UP steps Bit 31-24: DOWN steps

2.6.3 LAYOUT

Bit 31–24	Bit 23–16	Bit 15–0
Register 40001 / Bit 15–8	Register 40001 / Bit 7–0	Register 40002 / Bit 15–0
DOWN_STEPS	UP_STEPS	MASK

- MASK** 16-bit mask to select the drive(s) that should start STEP movement.
- UP_STEPS** Number of steps (0..255) in UP direction. Zero (0) means no STEPS in UP direction.
- DOWN_STEPS** Number of steps (0..255) in DOWN direction. Zero (0) means no STEPS in DOWN direction.

2.7 POS AND STEP MOVEMENT

Start an absolute positioning movement for selected drive(s). Optionally a relative movement in UP or DOWN direction can be defined.

MODBUS IMPLEMENTIERUNG

2.7.1 COMMAND

Accessible with Write Single Register (0x06) or Write Multiple Registers (0x10) command.

2.7.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40003-40005	0x0002-0x0004	POS_STEP	SMI drive POS STEP UP/DOWN	Bit 15-0: Mask Bit 31-16: Position Bit 39-32: UP steps Bit 47-40: DOWN steps

2.7.3 LAYOUT

Bit 47–40	Bit 39–32	Bit 31–16	Bit 15–0
Register 40003 / Bit 15–8	Register 40003 / Bit 7–0	Register 40004 / Bit 15–0	Register 40005 / Bit 15–0
DOWN_STEPS	UP_STEPS	POS	MASK

- MASK** 16-bit mask to select the drive(s) that should start STEP movement.
- POS** Absolute position to move to. 0x0000 = upper position, 0xFFFF = lower position.
- UP_STEPS** Number of steps (0..255) in UP direction. Zero (0) means no STEPS in UP direction.
- DOWN_STEPS** Number of steps (0..255) in DOWN direction. Zero (0) means no STEPS in DOWN direction.

2.7.4 REMARKS

It is not allowed to define both STEPS in UP and DOWN direction. When both UP_STEPS and DOWN_STEPS are set, only UP_STEPS will be used.

2.8 TILT MOVEMENT

Tilt movement is only for D6 drives (Vestamatic). Adjust the slat angle for the selected drive(s) to an absolute tilt orientation.

2.8.1 COMMAND

Accessible with Write Single Register (0x06) or Write Multiple Registers (0x10) command.

2.8.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40006-40007	0x0005-0x0006	TILT	SMI drive TILT position	Bit 15-0: Mask Bit 23-16: Tilt

2.8.3 LAYOUT

Bit 31–24	Bit 23–16	Bit 15–0
Register 40006 / Bit 15-8	Register 40006 / Bit 7-0	Register 40007 / Bit 15-0
-	TILT	MASK

MODBUS IMPLEMENTIERUNG

- MASK** 16-bit mask to select the drive(s) that should start TILT movement.
TILT Absolute tilt orientation (0..255).

The absolute tilt setting is defined as follows:

- 127 (0x7F) slats are completely closed in the DOWN direction
- 0 (0x00) slats are open horizontally
- -128 (0x80) slats are completely closed in the UP direction

2.9 POS AND TILT MOVEMENT

Start an absolute positioning movement for selected drive(s), followed by absolute tilt orientation. Only for D6 drives (Vestamatic).

2.9.1 COMMAND

Accessible with Write Single Register (0x06) or Write Multiple Registers (0x10) command.

2.9.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40008-40010	0x0007-0x0009	POS_TILT	SMI drive POS TILT	Bit 15-0: Mask Bit 31-16: Position Bit 39-32: Tilt

2.9.3 LAYOUT

Bit 47-40	Bit 39-32	Bit 31-16	Bit 15-0
Register 40008 / Bit 15-8	Register 40008 / Bit 7-0	Register 40009 / Bit 15-0	Register 40010 / Bit 15-0
-	TILT	POS	MASK

- MASK** 16-bit mask to select the drive(s) that should start TILT movement.
POS Absolute position to move to. 0x0000 = upper position, 0xFFFF = lower position.
TILT Absolute tilt orientation (0..255). (For conversion, refer to 9.2.8.3)

2.10 READ OR WRITE POS1

Read or write intermediate position 1 (POS1).

2.10.1 COMMAND

Accessible with Write Single Register (0x06), Write Multiple Registers (0x10) or Read Holding Registers (0x03) command.

2.9.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40017	0x0010	POS1_0	SMI drive 0 POS1	Bit 15-0: POS1
40018	0x0011	POS1_1	SMI drive 1 POS1	Bit 15-0: POS1
:	:	:	:	:
40032	0x001F	POS1_15	SMI drive 15 POS1	Bit 15-0: POS1

MODBUS IMPLEMENTIERUNG

2.11 READ OR WRITE POS2

Read or write intermediate position 2 (POS2).

2.11.1 COMMAND

Accessible with Write Single Register (0x06), Write Multiple Registers (0x10) or Read Holding Registers (0x03) command.

2.9.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40033	0x0020	POS2_0	SMI drive 0 POS2	Bit 15-0: POS2
40034	0x0021	POS2_1	SMI drive 1 POS2	Bit 15-0: POS2
:	:	:	:	:
40048	0x002F	POS2_15	SMI drive 15 POS2	Bit 15-0: POS2

3 MAINTENANCE

3.1 GATEWAY OPTIONS

Read or write gateway options.

3.1.1 COMMAND

Accessible with Write Single Register (0x06), Write Multiple Registers (0x10) or Read Holding Registers (0x03) command.

3.1.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40011	0x000A	GW_OPTIONS	Gateway options	Bit 0: error feedback enable flag Bit 1-3: - Bit 4: error feedback flag state after power-up Bit 5: cycle logging flag state after power-up

3.1.3 REMARKS

Error feedback is not possible in Modbus mode.

3.2 SMI TUNNEL (TRANSMIT)

Send SMI message direct to the SMI bus.

3.2.1 COMMAND

Accessible with Write Multiple Registers (0x10) command.

MODBUS IMPLEMENTIERUNG

3.2.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40049	0x0030	SML_TX_LEN	SML_TX message length	Bit 7-0: SMI transmit message length
40050	0x0031	SML_TX_MSG_0_1	SML TX message data byte 0 and 1	SMI transmit message data Bit 7-0: data byte 0 Bit 15-8: data byte 1
40051	0x0032	SML_TX_MSG_2_3	SML TX message data byte 2 and 3	SMI transmit message data Bit 7-0: data byte 2 Bit 15-8: data byte 3
:	:	:	:	:
40058	0x0039	SML_TX_MSG_16	SML TX message data byte 16	SMI transmit message data Bit 7-0: data byte 16 Bit 15-8:

3.3 SMI TUNNEL (RECEIVE)

Receive response of sent SMI message.

3.3.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

3.3.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30129	0x0080	SML_RX_RTYP	SML RX response type and SMI command	Bit 7-0: Response type Bit 15-8: SMI command
30130	0x0081	SML_RX_LEN	SML RX response data length	Bit 7-0: data length Bit 15-8: -
30131	0x0082	SML_RX_MSG_0_1	SML RX message data byte 0 and 1	SMI receive message data Bit 7-0: data byte 0 Bit 15-8: data byte 1
30132	0x0083	SML_RX_MSG_2_3	SML RX message data byte 2 and 3	SMI receive message data Bit 7-0: data byte 2 Bit 15-8: data byte 3

3.3.3 REMARKS

Response type:

- 0x00 SMI message send/receive still in progress
- 0xFF SMI message send successfully and accepted by addressed drive(s)
- 0xE0 Error occurred. Received message data byte 0 contains error code:
 - 0x01: Checksum error
 - 0x02: Format error
 - 0x03: Echo error

Weiter auf Folgeseite

MODBUS IMPLEMENTIERUNG

3.3.3 REMARKS

Response type:

- 0x04: Data overflow
- 0x05: Timeout error

0xEF SMI message send successfully and response is in received message data bytes 0..3
SMI command to which response belongs.

3.4 PARAMETER (REQUEST)

Parameter requested from addressed drive.

3.4.1 COMMAND

Accessible with Write Multiple Registers (0x10) command.

3.4.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40059	0x003A	PAR_REQ_SLAVE_SIZE	Parameter size and addressed slave	Bit 7-0: Slave address Bit 15-8: Parameter size
40060	0x003B	PAR_REQ_INDEX	Parameter index	Bit 11-0: Parameter index Bit 15-12: always 0

3.5 PARAMETER (RESPONSE)

Parameter response from addressed drive.

3.5.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

3.5.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30134	0x0085	PAR_RSP_SLAVE_SIZE	Parameter size and addressed slave	Bit 7-0: Slave address Bit 15-8: Parameter size
30135	0x0086	PAR_RSP_INDEX	Parameter index	Bit 11-0: Parameter index Bit 15-12: always 0
30136	0x0087	PAR_RSP_DATA_0_1	Parameter data byte 0 and 1	Bit 7-0: Data byte 0 Bit 15-8: Data byte 1
30137	0x0088	PAR_RSP_DATA_2_3	Parameter data byte 2 and 3	Bit 7-0: Data byte 2 Bit 15-8: Data byte 3

3.5.3 REMARKS

Parameter size zero (0) means that no response has been received from slave.

MODBUS IMPLEMENTIERUNG

3.6 AUTO ADDRESSING

Writing a ON (0xFF00) value will start the auto address procedure. Drive address conflicts will be solved and every drive gets an unique slave address (0..15). This procedure can take up to 60 seconds. As long as the auto address procedure is in progress, reading the "coil" value will return a ON value.

3.6.1 COMMAND

Accessible with Write Single Coil (0x05) or Read Coils (0x01) command.

3.6.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
81	0x0050	AUTO_ADDRESS	Start auto address procedure	WRITE ON will start auto-address procedure. READ ON means auto-address still in progress.

3.6.3 SLAVE ADDRESS (REQUEST)

Modify slave address of drive with given manufacturer ID and key ID.

3.7.1 COMMAND

Accessible with Write Multiple Registers (0x10) command.

3.7.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40061-40062	0x003C-0x003D	SLAVE_REQ_KEY_ID	Set slave address key ID request	Bit 31-0: Key-ID
40063	0x003E	SLAVE_REQ_MAN_ADDR	Set slave address manufacturer ID and new slave address request	Bit 7-0: Manufacturer ID Bit 15-8: New slave address

3.8 SLAVE ADDRESS (RESPONSE)

Check if modify slave address was successful.

3.8.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

MODBUS IMPLEMENTIERUNG

3.7.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30138-30139	0x0089-0x008A	SLAVE_RSP_KEY_ID	Set slave address key ID response	Bit 31-0: Key-ID
30140	0x008B	SLAVE_RSP_MAN_ADDR	Set slave address manufacturer ID and new slave address response	Bit 7-0: Manufacturer ID Bit 15-8: Set slave address

3.8.3 REMARKS

When high-nibble of SLAVE address is 0xF0, an error occurred during SLAVE address set.

3.9 SLAVE SWAP (REQUEST)

Swap slave addresses of 2 drives.

3.9.1 COMMAND

Accessible with Write Single Register (0x06) or Write Multiple Registers (0x10)

3.9.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
40015	0x000E	SWAP_REQ_SLA	Swap slave addresses request	Bit 15-12: - Bit 11-8: Slave address 1 Bit 7-4: - Bit 3-0: Slave address 2

3.9.3 REMARKS

- Only low-nibble is used of slave address. High-nibble is ignored.
- No address-conflict allowed on specified slave addresses.

3.10 SLAVE SWAP (RESPONSE)

Check if swap slave addresses was successful.

3.10.1 COMMAND

Accessible with Read Inputs Registers (0x04) command.

3.10.2 REGISTER

Register address	Modbus protocol start address	Field name	Description	Result
30010	0x0009	SWAP_RSP_SLA	Swap slave addresses response	Bit 15-8: Swap result 1 Bit 7-0: Swap result 2

3.10.3 REMARKS

Swap result 0xFF = swap slave addresses in progress 0x01..0x05 = error occurred during swap
0x00 = swap result successfully

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
1	0x0000	UP_0	SMI drive 0 UP	1 = UP / 0 = no action
2	0x0001	UP_1	SMI drive 1 UP	1 = UP / 0 = no action
3	0x0002	UP_2	SMI drive 2 UP	1 = UP / 0 = no action
4	0x0003	UP_3	SMI drive 3 UP	1 = UP / 0 = no action
5	0x0004	UP_4	SMI drive 4 UP	1 = UP / 0 = no action
6	0x0005	UP_5	SMI drive 5 UP	1 = UP / 0 = no action
7	0x0006	UP_6	SMI drive 6 UP	1 = UP / 0 = no action
8	0x0007	UP_7	SMI drive 7 UP	1 = UP / 0 = no action
9	0x0008	UP_8	SMI drive 8 UP	1 = UP / 0 = no action
10	0x0009	UP_9	SMI drive 9 UP	1 = UP / 0 = no action
11	0x000A	UP_10	SMI drive 10 UP	1 = UP / 0 = no action
12	0x000B	UP_11	SMI drive 11 UP	1 = UP / 0 = no action
13	0x000C	UP_12	SMI drive 12 UP	1 = UP / 0 = no action
14	0x000D	UP_13	SMI drive 13 UP	1 = UP / 0 = no action
15	0x000E	UP_14	SMI drive 14 UP	1 = UP / 0 = no action
16	0x000F	UP_15	SMI drive 15 UP	1 = UP / 0 = no action
17	0x0010	DOWN_0	SMI drive 0 DOWN	1 = DOWN / 0 = no action
18	0x0011	DOWN_1	SMI drive 1 DOWN	1 = DOWN / 0 = no action
19	0x0012	DOWN_2	SMI drive 2 DOWN	1 = DOWN / 0 = no action
20	0x0013	DOWN_3	SMI drive 3 DOWN	1 = DOWN / 0 = no action
21	0x0014	DOWN_4	SMI drive 4 DOWN	1 = DOWN / 0 = no action
22	0x0015	DOWN_5	SMI drive 5 DOWN	1 = DOWN / 0 = no action
23	0x0016	DOWN_6	SMI drive 6 DOWN	1 = DOWN / 0 = no action
24	0x0017	DOWN_7	SMI drive 7 DOWN	1 = DOWN / 0 = no action
25	0x0018	DOWN_8	SMI drive 8 DOWN	1 = DOWN / 0 = no action
26	0x0019	DOWN_9	SMI drive 9 DOWN	1 = DOWN / 0 = no action
27	0x001A	DOWN_10	SMI drive 10 DOWN	1 = DOWN / 0 = no action
28	0x001B	DOWN_11	SMI drive 11 DOWN	1 = DOWN / 0 = no action
29	0x001C	DOWN_12	SMI drive 12 DOWN	1 = DOWN / 0 = no action
30	0x001D	DOWN_13	SMI drive 13 DOWN	1 = DOWN / 0 = no action
31	0x001E	DOWN_14	SMI drive 14 DOWN	1 = DOWN / 0 = no action
32	0x001F	DOWN_15	SMI drive 15 DOWN	1 = DOWN / 0 = no action
33	0x0020	STOP_0	SMI drive 0 STOP	1 = STOP / 0 = no action
34	0x0021	STOP_1	SMI drive 1 STOP	1 = STOP / 0 = no action
35	0x0022	STOP_2	SMI drive 2 STOP	1 = STOP / 0 = no action
36	0x0023	STOP_3	SMI drive 3 STOP	1 = STOP / 0 = no action
37	0x0024	STOP_4	SMI drive 4 STOP	1 = STOP / 0 = no action
38	0x0025	STOP_5	SMI drive 5 STOP	1 = STOP / 0 = no action
39	0x0026	STOP_6	SMI drive 6 STOP	1 = STOP / 0 = no action
40	0x0027	STOP_7	SMI drive 7 STOP	1 = STOP / 0 = no action
41	0x0028	STOP_8	SMI drive 8 STOP	1 = STOP / 0 = no action

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
42	0x0029	STOP_9	SMI drive 9 STOP	1 = STOP / 0 = no action
43	0x002A	STOP_10	SMI drive 10 STOP	1 = STOP / 0 = no action
44	0x002B	STOP_11	SMI drive 11 STOP	1 = STOP / 0 = no action
45	0x002C	STOP_12	SMI drive 12 STOP	1 = STOP / 0 = no action
46	0x002D	STOP_13	SMI drive 13 STOP	1 = STOP / 0 = no action
47	0x002E	STOP_14	SMI drive 14 STOP	1 = STOP / 0 = no action
48	0x002F	STOP_15	SMI drive 15 STOP	1 = STOP / 0 = no action
49	0x0030	GOTO_POS1_0	SMI drive 0 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
50	0x0031	GOTO_POS1_1	SMI drive 1 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
51	0x0032	GOTO_POS1_2	SMI drive 2 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
52	0x0033	GOTO_POS1_3	SMI drive 3 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
53	0x0034	GOTO_POS1_4	SMI drive 4 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
54	0x0035	GOTO_POS1_5	SMI drive 5 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
55	0x0036	GOTO_POS1_6	SMI drive 6 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
56	0x0037	GOTO_POS1_7	SMI drive 7 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
57	0x0038	GOTO_POS1_8	SMI drive 8 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
58	0x0039	GOTO_POS1_9	SMI drive 9 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
59	0x003A	GOTO_POS1_10	SMI drive 10 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
60	0x003B	GOTO_POS1_11	SMI drive 11 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
61	0x003C	GOTO_POS1_12	SMI drive 12 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
62	0x003D	GOTO_POS1_13	SMI drive 13 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
63	0x003E	GOTO_POS1_14	SMI drive 14 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
64	0x003F	GOTO_POS1_15	SMI drive 15 GOTO_POS1	1 = GOTO_POS1 / 0 = no action
65	0x0040	GOTO_POS2_0	SMI drive 0 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
66	0x0041	GOTO_POS2_1	SMI drive 1 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
67	0x0042	GOTO_POS2_2	SMI drive 2 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
68	0x0043	GOTO_POS2_3	SMI drive 3 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
69	0x0044	GOTO_POS2_4	SMI drive 4 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
70	0x0045	GOTO_POS2_5	SMI drive 5 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
71	0x0046	GOTO_POS2_6	SMI drive 6 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
72	0x0047	GOTO_POS2_7	SMI drive 7 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
73	0x0048	GOTO_POS2_8	SMI drive 8 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
74	0x0049	GOTO_POS2_9	SMI drive 9 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
75	0x004A	GOTO_POS2_10	SMI drive 10 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
76	0x004B	GOTO_POS2_11	SMI drive 11 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
77	0x004C	GOTO_POS2_12	SMI drive 12 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
78	0x004D	GOTO_POS2_13	SMI drive 13 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
79	0x004E	GOTO_POS2_14	SMI drive 14 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
80	0x004F	GOTO_POS2_15	SMI drive 15 GOTO_POS2	1 = GOTO_POS2 / 0 = no action
81	0x0050	AUTO_ADDRESS	Start auto address procedure	WRITE ON will start auto-address procedure. READ ON means auto-address still in progress.

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
30001-30002	0x0000-0x0001	VERSION	Software version of application	Bit 31-24: HW Bit 23-16: SW major Bit 15-8: SW minor Bit 7-0: SW revision
30003	0x0002	PRESENT	SMI drive 0..15 present state	Bit n set: Present Bit n clear: Not in use
30004	0x0003	READY	SMI drive 0..15 ready state	Bit n set: Ready (idle) Bit n clear: Busy (running)
30005	0x0004	ERR_0	Error log entry 0 (most recent)	Error code
30006	0x0005	ERR_1	Error log entry 1	Error code
30007	0x0006	ERR_2	Error log entry 2	Error code
30008	0x0007	ERR_3	Error log entry 3	Error code
30009	0x0008	ERR_4	Error log entry 4	Error code
30010	0x0009	SWAP_RSP_SLA	Swap slave addresses response	Bit 15-8: Swap result 1 Bit 7-0: Swap result 2
30011-30016	0x000A-0x000F	-	-	-
30017	0x0010	POS_0	SMI drive 0 position	SMI drive position
30018	0x0011	POS_1	SMI drive 1 position	SMI drive position
30019	0x0012	POS_2	SMI drive 2 position	SMI drive position
30020	0x0013	POS_3	SMI drive 3 position	SMI drive position
30021	0x0014	POS_4	SMI drive 4 position	SMI drive position
30022	0x0015	POS_5	SMI drive 5 position	SMI drive position
30023	0x0016	POS_6	SMI drive 6 position	SMI drive position
30024	0x0017	POS_7	SMI drive 7 position	SMI drive position
30025	0x0018	POS_8	SMI drive 8 position	SMI drive position
30026	0x0019	POS_9	SMI drive 9 position	SMI drive position
30027	0x001A	POS_10	SMI drive 10 position	SMI drive position
30028	0x001B	POS_11	SMI drive 11 position	SMI drive position
30029	0x001C	POS_12	SMI drive 12 position	SMI drive position
30030	0x001D	POS_13	SMI drive 13 position	SMI drive position
30031	0x001E	POS_14	SMI drive 14 position	SMI drive position
30032	0x001F	POS_15	SMI drive 15 position	SMI drive position
30033	0x0020	STAT_TILT_0	SMI drive 0 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30034	0x0021	STAT_TILT_1	SMI drive 1 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30035	0x0022	STAT_TILT_2	SMI drive 2 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30036	0x0023	STAT_TILT_3	SMI drive 3 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30037	0x0024	STAT_TILT_4	SMI drive 4 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30038	0x0025	STAT_TILT_5	SMI drive 5 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30039	0x0026	STAT_TILT_6	SMI drive 6 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30040	0x0027	STAT_TILT_7	SMI drive 7 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30041	0x0028	STAT_TILT_8	SMI drive 8 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30042	0x0029	STAT_TILT_9	SMI drive 9 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30043	0x002A	STAT_TILT_10	SMI drive 10 status and tilt	Bit 7-0: Status Bit 15-8: Tilt

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
30044	0x002B	STAT_TILT_11	SMI drive 11 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30045	0x002C	STAT_TILT_12	SMI drive 12 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30046	0x002D	STAT_TILT_13	SMI drive 13 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30047	0x002E	STAT_TILT_14	SMI drive 14 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30048	0x002F	STAT_TILT_15	SMI drive 15 status and tilt	Bit 7-0: Status Bit 15-8: Tilt
30049	0x0030	MAN_ID_0	SMI drive 0 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30050	0x0031	MAN_ID_1	SMI drive 1 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30051	0x0032	MAN_ID_2	SMI drive 2 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30052	0x0033	MAN_ID_3	SMI drive 3 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30053	0x0034	MAN_ID_4	SMI drive 4 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30054	0x0035	MAN_ID_5	SMI drive 5 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30055	0x0036	MAN_ID_6	SMI drive 6 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30056	0x0037	MAN_ID_7	SMI drive 7 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30057	0x0038	MAN_ID_8	SMI drive 8 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30058	0x0039	MAN_ID_9	SMI drive 9 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30059	0x003A	MAN_ID_10	SMI drive 10 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30060	0x003B	MAN_ID_11	SMI drive 11 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30061	0x003C	MAN_ID_12	SMI drive 12 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30062	0x003D	MAN_ID_13	SMI drive 13 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30063	0x003E	MAN_ID_14	SMI drive 14 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30064	0x003F	MAN_ID_15	SMI drive 15 manufacturer ID and type	Bit 7-0: MID Bit 15-8: Type
30065-30066	0x0040-0x0041	KEY_ID_0	SMI drive 0 Key-ID	Bit 31-0: Key-ID
30067-30068	0x0042-0x0043	KEY_ID_1	SMI drive 1 Key-ID	Bit 31-0: Key-ID
30069-30070	0x0044-0x0045	KEY_ID_2	SMI drive 2 Key-ID	Bit 31-0: Key-ID
30071-30072	0x0046-0x0047	KEY_ID_3	SMI drive 3 Key-ID	Bit 31-0: Key-ID
30073-30074	0x0048-0x0049	KEY_ID_4	SMI drive 4 Key-ID	Bit 31-0: Key-ID
30075-30076	0x004A-0x004B	KEY_ID_5	SMI drive 5 Key-ID	Bit 31-0: Key-ID
30077-30078	0x004C-0x004D	KEY_ID_6	SMI drive 6 Key-ID	Bit 31-0: Key-ID
30079-30080	0x004E-0x004F	KEY_ID_7	SMI drive 7 Key-ID	Bit 31-0: Key-ID
30081-30082	0x0050-0x0051	KEY_ID_8	SMI drive 8 Key-ID	Bit 31-0: Key-ID
30083-30084	0x0052-0x0053	KEY_ID_9	SMI drive 9 Key-ID	Bit 31-0: Key-ID
30085-30086	0x0054-0x0055	KEY_ID_10	SMI drive 10 Key-ID	Bit 31-0: Key-ID
30087-30088	0x0056-0x0057	KEY_ID_11	SMI drive 11 Key-ID	Bit 31-0: Key-ID
30089-30090	0x0058-0x0059	KEY_ID_12	SMI drive 12 Key-ID	Bit 31-0: Key-ID
30091-30092	0x005A-0x005B	KEY_ID_13	SMI drive 13 Key-ID	Bit 31-0: Key-ID
30093-30094	0x005C-0x005D	KEY_ID_14	SMI drive 14 Key-ID	Bit 31-0: Key-ID
30095-30096	0x005E-0x005F	KEY_ID_15	SMI drive 15 Key-ID	Bit 31-0: Key-ID
30129	0x0080	SMI_RX_RTYP	SMI RX response type and SMI command	Bit 7-0: Response type Bit 15-8: SMI command
30130	0x0081	SMI_RX_LEN	SMI RX response data length	Bit 7-0: data length Bit 15-8: -
30131	0x0082	SMI_RX_MSG_0_1	SMI RX message data byte 0 and 1	SMI receive message data Bit 7-0: data byte 0 Bit 15-8: data byte 1

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
30132	0x0083	SMI_RX_MSG_2_3	SMI RX message data byte 2 and 3	SMI receive message data Bit 7-0: data byte 2 Bit 15-8: data byte 3
30133	0x0084	-	-	-
30134	0x0085	PAR_RSP_SLAVE_SIZE	Parameter size and addressed slave	Bit 7-0: Slave address Bit 15-8: Parameter size
30135	0x0086	PAR_RSP_INDEX	Parameter index	Bit 11-0: Parameter index Bit 15-12: always 0
30136	0x0087	PAR_RSP_DATA_0_1	Parameter data byte 0 and 1	Bit 7-0: Data byte 0 Bit 15-8: Data byte 1
30137	0x0088	PAR_RSP_DATA_2_3	Parameter data byte 2 and 3	Bit 7-0: Data byte 2 Bit 15-8: Data byte 3
30138-30139	0x0089-0x008A	SLAVE_RSP_KEY_ID	Set slave address key ID response	Bit 31-0: Key-ID
30140	0x008B	SLAVE_RSP_MAN_ADDR	Set slave address manufacturer ID and new slave address response	Bit 7-0: Manufacturer ID Bit 15-8: Set slave address
40001-40002	0x0000-0x0001	STEP	SMI drive STEP UP/DOWN	Bit 15-0: Mask Bit 23-16: UP steps Bit 31-24: DOWN steps
40003-40005	0x0002-0x0004	POS_STEP	SMI drive POS STEP UP/DOWN	Bit 15-0: Mask Bit 31-16: Position Bit 39-32: UP steps Bit 47-40: DOWN steps
40006-40007	0x0005-0x0006	TILT	SMI drive TILT position	Bit 15-0: Mask Bit 23-16: Tilt
40008-40010	0x0007-0x0009	POS_TILT	SMI drive POSTILT	Bit 15-0: Mask Bit 31-16: Position Bit 39-32: Tilt
40011	0x000A	GW_OPTIONS	Gateway options	Bit 0: error flag Bit 4: error flag after power-up Bit 5: cycle logging flag after power-up
40012	0x000B	POS_TILT_UPD	Update position and tilt info mask	Bit 15-0: Mask
40013	0x000C	POS1_UPD	Update POS1 info mask	Bit 15-0: Mask
40014	0x000D	POS2_UPD	Update POS2 info mask	Bit 15-0: Mask
40015	0x000E	SWAP_REQ_SLA	Swap slave addresses request	Bit 15-12: - Bit 11-8: Slave address 1 Bit 7-4: - Bit 3-0: Slave address 2
40016	0x000F	-	-	-
40017	0x0010	POS1_0	SMI drive 0 POS1	Bit 31-0: POS1
40018	0x0011	POS1_1	SMI drive 1 POS1	Bit 31-0: POS1
40019	0x0012	POS1_2	SMI drive 2 POS1	Bit 31-0: POS1
40020	0x0013	POS1_3	SMI drive 3 POS1	Bit 31-0: POS1
40021	0x0014	POS1_4	SMI drive 4 POS1	Bit 31-0: POS1
40022	0x0015	POS1_5	SMI drive 5 POS1	Bit 31-0: POS1
40023	0x0016	POS1_6	SMI drive 6 POS1	Bit 31-0: POS1
40024	0x0017	POS1_7	SMI drive 7 POS1	Bit 31-0: POS1
40025	0x0018	POS1_8	SMI drive 8 POS1	Bit 31-0: POS1

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
40026	0x0019	POS1_9	SMI drive 9 POS1	Bit 31-0: POS1
40022	0x0015	POS1_5	SMI drive 5 POS1	Bit 31-0: POS1
40023	0x0016	POS1_6	SMI drive 6 POS1	Bit 31-0: POS1
40024	0x0017	POS1_7	SMI drive 7 POS1	Bit 31-0: POS1
40025	0x0018	POS1_8	SMI drive 8 POS1	Bit 31-0: POS1
40026	0x0019	POS1_9	SMI drive 9 POS1	Bit 31-0: POS1
40027	0x001A	POS1_10	SMI drive 10 POS1	Bit 31-0: POS1
40028	0x001B	POS1_11	SMI drive 11 POS1	Bit 31-0: POS1
40029	0x001C	POS1_12	SMI drive 12 POS1	Bit 31-0: POS1
40030	0x001D	POS1_13	SMI drive 13 POS1	Bit 31-0: POS1
40031	0x001E	POS1_14	SMI drive 14 POS1	Bit 31-0: POS1
40032	0x001F	POS1_15	SMI drive 15 POS1	Bit 31-0: POS1
40033	0x0020	POS2_0	SMI drive 0 POS2	Bit 31-0: POS2
40034	0x0021	POS2_1	SMI drive 1 POS2	Bit 31-0: POS2
40035	0x0022	POS2_2	SMI drive 2 POS2	Bit 31-0: POS2
40036	0x0023	POS2_3	SMI drive 3 POS2	Bit 31-0: POS2
40037	0x0024	POS2_4	SMI drive 4 POS2	Bit 31-0: POS2
40038	0x0025	POS2_5	SMI drive 5 POS2	Bit 31-0: POS2
40039	0x0026	POS2_6	SMI drive 6 POS2	Bit 31-0: POS2
40040	0x0027	POS2_7	SMI drive 7 POS2	Bit 31-0: POS2
40041	0x0028	POS2_8	SMI drive 8 POS2	Bit 31-0: POS2
40042	0x0029	POS2_9	SMI drive 9 POS2	Bit 31-0: POS2
40043	0x002A	POS2_10	SMI drive 10 POS2	Bit 31-0: POS2
40044	0x002B	POS2_11	SMI drive 11 POS2	Bit 31-0: POS2
40045	0x002C	POS2_12	SMI drive 12 POS2	Bit 31-0: POS2
40046	0x002D	POS2_13	SMI drive 13 POS2	Bit 31-0: POS2
40047	0x002E	POS2_14	SMI drive 14 POS2	Bit 31-0: POS2
40048	0x002F	POS2_15	SMI drive 15 POS2	Bit 31-0: POS2
40049	0x0030	SMI_TX_LEN	SMI_TX message length	Bit 7-0: SMI transmit message length
40050	0x0031	SMI_TX_MSG_0_1	SMI TX message data byte 0 and 1	SMI transmit message data Bit 7-0: data byte 0 Bit 15-8: data byte 1
40051	0x0032	SMI_TX_MSG_2_3	SMI TX message data byte 2 and 3	Bit 7-0: data byte 2 Bit 15-8: data byte 3
40052	0x0033	SMI_TX_MSG_4_5	SMI TX message data byte 4 and 5	Bit 7-0: data byte 4 Bit 15-8: data byte 5
40053	0x0034	SMI_TX_MSG_6_7	SMI TX message data byte 6 and 7	Bit 7-0: data byte 6 Bit 15-8: data byte 7
40054	0x0035	SMI_TX_MSG_8_9	SMI TX message data byte 8 and 9	Bit 7-0: data byte 8 Bit 15-8: data byte 9
40055	0x0036	SMI_TX_MSG_10_11	SMI TX message data byte 10 and 11	Bit 7-0: data byte 10 Bit 15-8: data byte 11
40056	0x0037	SMI_TX_MSG_12_13	SMI TX message data byte 12 and 13	Bit 7-0: data byte 12 Bit 15-8: data byte 13

REGISTER-ÜBERBLICK

Register address	Modbus protocol start address	Field name	Description	Result
40057	0x0038	SMI_TX_MSG_14_15	SMI TX message data byte 14 and 15	Bit 7-0: data byte 14 Bit 15-8: data byte 15
40058	0x0039	SMI_TX_MSG_16	SMI TX message data byte 16	Bit 7-0: data byte 16 Bit 15-8: -
40059	0x003A	PAR_REQ_SLAVE_SIZE	Parameter size and addressed slave	Bit 7-0: Slave address Bit 15-8: Parameter size
40060	0x003B	PAR_REQ_INDEX	Parameter index	Bit 15-0: Parameter index
40061-40062	0x003C-0x003D	SLAVE_REQ_KEY_ID	Set slave address key ID request	Bit 31-0: Key-ID
40063	0x003E	SLAVE_REQ_MAN_ADDR	Set slave address manufacturer ID and new slave address request	Bit 7-0: Manufacturer ID Bit 15-8: New slave address