

# Modbus RTU Kommunikationsprotokoll für motrona Gerät



<b>Version:</b>	<b>Beschreibung:</b>
Modbus_RTU_oi_d / CF / Feb 2019	Erste Auflage

<b>Rechtliche Hinweise::</b>
Sämtliche Inhalte dieser Gerätebeschreibung unterliegen den Nutzungs- und Urheberrechten der motrona GmbH. Jegliche Vervielfältigung, Veränderung, Weiterverwendung und Publikation in anderen elektronischen oder gedruckten Medien, sowie deren Veröffentlichung im Internet, bedarf einer vorherigen schriftlichen Genehmigung durch die motrona GmbH.

## Inhaltsverzeichnis

<b>1. Modbus RTU Protokoll für die serielle Schnittstelle der touchMATRIX® Serie .....</b>	<b>3</b>
Allgemein .....	3
Parametereinstellung .....	3
Modbus-Kommunikation .....	5
1.1.1 Read Holding Registers und Write Multiple Registers .....	5
1.1.2 Read Coils und Write Single Coil .....	6
1.1.3 Diagnose .....	6
<b>2. Modbus RTU Protokoll für die serielle Schnittstelle des MC800 .....</b>	<b>7</b>
2.1 Allgemein .....	7
2.2 Parametereinstellung .....	7
2.3 Modbus Kommunikation .....	8
2.3.1 Read Holding Registers und Write Multiple Registers .....	8
2.3.2 Read Coils und Write Single Coil .....	9
2.3.3 Diagnose .....	9
2.4 Parametertabelle .....	10

# 1. Modbus RTU Protokoll für die serielle Schnittstelle der touchMATRIX® Serie

## Allgemein

Die Modbus-Schnittstelle der touchMATRIX Serie ist ein Standard Modbus RTU Slave und bietet folgende Modbus-Funktionen:

- Read Coils
- Write Single Coil
- Read Holding Registers
- Write Multiple Registers
- Diagnose

Für den Betrieb des Interfacemoduls und das Verständnis dieses Handbuchs wird Grundwissen in der Modbus RTU-Kommunikation vorausgesetzt.

## Parametereinstellung

Erforderliche Parametereinstellungen im "Serial Menue":

### UNIT NUMMER

Nicht gültig für die Modbus-Kommunikation

(Zur Einstellung der Modbus-Adresse siehe Parameter "MODBUS")

### SERIAL BAUD RATE

Mit diesem Parameter wird die serielle Baudrate eingestellt.

0	9600	9600 Baud
1	19200	19200 Baud
2	38400	38400 Baud

### SERIAL FORMAT

Mit diesem Parameter wird das Bit-Datenformat eingestellt.

0	7-EVEN-1	Nicht mit dem Modbus-Protokoll verwendbar.		
1	7-EVEN-2			
2	7-ODD-1			
3	7-ODD-2			
4	7-NONE-1			
5	7-NONE-2			
6	8-EVEN-1	8 Daten	Parity even	1 Stopp
7	8-ODD-1	8 Daten	Parity odd	1 Stopp
8	8-NONE-1	Nicht mit dem Modbus-Protokoll verwendbar.		
9	8-NONE-2	8 Daten	kein Parity	2 Stopps

Fortsetzung "Parametereinstellung":

**SERIAL INIT**

Nicht gültig für die Modbus-Kommunikation

**SERIAL PROTOCOL**

Nicht gültig für die Modbus-Kommunikation

**SERIAL TIMER (S)**

Nicht gültig für die Modbus-Kommunikation

**SERIAL VALUE**

Nicht gültig für die Modbus-Kommunikation

**MODBUS**

Dieser Parameter aktiviert das Modbus-Protokoll und bestimmt die Modbus-Adresse.

	0	Nicht mit Modbus-Protokoll verwenden (Modbus ist deaktiviert)
	1 ... 247	Modbus aktiviert: Die serielle Schnittstelle verwendet das Modbus RTU-Protokoll Die hier eingestellte Zahl bestimmt die Modbus-Knotenadresse.

## Modbus-Kommunikation

Folgende Modbus-Funktionen sind verfügbar:

### 1.1.1 Read Holding Registers und Write Multiple Registers

Mit den Funktionen "Read Holding Registers" und "Write Multiple Register" kann auf alle Register des Gerätes zugegriffen werden. Alle Variablen (Istwerte) und Statusregister werden Modbus-Halteregistern zugeordnet.

Da alle Geräte-Register 32-Bit-Register sind, Modbus-Holding-Register jedoch nur 16-Bit-Register, benötigt jedes Geräte-Register zwei Holding-Register. (Aus diesem Grund ist die Verwendung der Modbus-Funktion "Write Single Register" nicht möglich.)

Bei jedem Lese-oder Schreibvorgang ist es nur möglich auf ein einzelnes touchMATRIX Geräte-Register zuzugreifen, daher muss die "Menge (oder Anzahl) der Register" im Modbus-Request immer 2 sein.

### Zugriff auf Parameter

Holding Register 0x0000 / 0x0001 hex und die nachfolgenden Holding Register erlauben den Zugriff auf die Geräte Parameter.

Die Halteregisternummern für einen bestimmten Parameter können mit dem Parameter # berechnet werden, der in der Parametertabelle im jeweiligen touchMATRIX Geräte-Handbuch zu finden ist:

Holding Register niedrig = (Parameter #) x 2

Holding Register hoch = (Parameter #) x 2 + 1

Beispiel:

Zugriff auf Parameter # 51 "PRESELECTION 1" durch Holding Register 0x0066 und 0x0067 hex.

### Zugriff auf Istwerte

Holding Register 0x1000 / 0x1001 hex und die nachfolgenden Holding Register erlauben den Zugriff auf Geräte Variablen (Istwert Register):

Holding Register 0x1000 / 0x1001 hex → Istwerte mit seriellem Code ":0"  
(Anzeigewert)

Holding Register 0x1002 / 0x1003 hex → Istwerte mit seriellem Code ":1"

Holding Register 0x1004 / 0x1005 hex → Istwerte mit seriellem Code ":2"

Holding Register 0x1006 / 0x1007 hex → Istwerte mit seriellem Code ":3"

etc.

### Zugriff aus Statusregister

Holding Register 0x2000 / 0x2001 hex und die folgenden Holding Register erlauben den Zugriff auf die Geräte Statusregister:

Holding Register 0x2000 / 0x2001 hex → Ausgangstatus (Ctrl. Out status, nur lesbar)

Holding Register 0x2002 / 0x2003 hex → Serielle Befehle

Holding Register 0x2004 / 0x2005 hex → Externe Befehle (Ctrl. In Status, nur lesbar)

Holding Register 0x2006 / 0x2007 hex → Alle Befehle (nur lesbar)

### 1.1.2 Read Coils und Write Single Coil

Mit den Funktionen "Read Coils" und "Write Single Coil" können einzelne Befehle gelesen und gesetzt / zurückgesetzt werden:

Coil Number	Serieller Code des Befehls	Befehl	
0	54	Reset / Set	Rücksetzen / Setzen des Wertes
1	55	Freeze Display	Einfrieren des Display Wertes
2	56	Touch Disable	Tastensperre: Touchscreen
3	57	Clear Lock	Sperre löschen
4	58	Clear Min/Max	Rücksetzen der Min. / Max. Werte
5	59	Serial Print (do not use with Modbus)	Senden serieller Daten
6	60	Teach Preset 1	Aktueller Anzeigewert wird als PRESELECTION 1 gespeichert
7	61	Teach Preset 2	Aktueller Anzeigewert wird als PRESELECTION 2 gespeichert
8	62	Teach Preset 3	Aktueller Anzeigewert wird als PRESELECTION 3 gespeichert
9	63	Teach Preset 4	Aktueller Anzeigewert wird als PRESELECTION 4 gespeichert
10	64	Scroll Display	Display Umschaltung (Siehe Anzeigedarstellung im Betrieb)
11	65	Clear Loop Time	Alle festgelegten Schaltbedingungen werden freigegeben
12	66	Start Preselection	Vorauswahl starten
13	67	Activate Data (not required with Modbus)	Daten aktivieren (nicht erforderlich bei Modbus)
14	68	Store to EEPROM	In EEPROM speichern
15	69	Testprogram (do not use with Modbus)	Testprogramm (nicht mit Modbus verwenden)

### 1.1.3 Diagnose

Das Gerät unterstützt die Diagnose-Unterfunktion 00 "Return Query Data".

Andere Diagnosefunktionen sind nicht verfügbar.

## 2. Modbus RTU Protokoll für die serielle Schnittstelle des MC800

### 2.1 Allgemein

Das Protokoll der seriellen Schnittstelle des MC800 wird vom DIL-Switch 1 ausgewählt:  
**Um das Modbus-Protokoll verwenden zu können, muss der DIL-Schalter 1 auf ON stehen.**

Die Modbus-Schnittstelle des MC800 ist ein Standard-Modbus-RTU-Slave und bietet die folgenden Modbus-Funktionen:

- Read Coils
- Write Single Coil
- Read Holding Registers
- Write Multiple Registers
- Diagnose

**Voraussetzung für den Betrieb des Interfacemoduls und das Verständnis dieses Handbuchs sind Grundkenntnisse in der Modbus RTU-Kommunikation.**

### 2.2 Parametereinstellung

Erforderliche Einstellungen in den "Communication Settings":

<b>Ser. Unit Address</b>	Nicht gültig für die Modbus-Kommunikation. (Zur Einstellung der Modbus-Adresse siehe Parameter "Modbus Address")			
<b>Ser. Baud Rate</b>	Übertragungsrate der seriellen Schnittstelle. 0: 38400 Bit/s 1: 19200 Bit/s 2: 9600 Bit/s (Werkseinstellung) 3: 4800 Bit/s 4: 2400 Bit/s			
<b>Serial Data Format:</b>	Datenformat der seriellen Schnittstelle			
	<b>Setting:</b>	<b>Data bits</b>	<b>Stop bits</b>	<b>Parity</b>
	0	Nicht mit dem Modbus-Protokoll verwendbar.		
	1			
	2			
	3			
	4			
	5			
	6	8	1	even
	7	8	1	odd
	8	Nicht mit dem Modbus-Protokoll verwendbar.		
	9	8	2	none
	<b>Werkseinstellung: 0</b>			

## 2.3 Modbus Kommunikation

Folgenden Modbus-Funktionen sind verfügbar:

### 2.3.1 Read Holding Registers und Write Multiple Registers

Mit den Funktionen "Read Holding Registers" und "Write Multiple Register" kann auf alle Register des MC800 zugegriffen werden.

Alle Parameter, Prozessdaten (Istwerte) und Statusregister werden Modbus-Halteregistern zugeordnet.

Da alle Geräte-Register 32-Bit-Register sind, Modbus-Holding-Register jedoch nur 16-Bit-Register, benötigt jedes Geräte-Register zwei Holding-Register. (Aus diesem Grund ist die Verwendung der Modbus-Funktion "Write Single Register" nicht möglich.)

Bei jedem Lese-oder Schreibvorgang ist es nur möglich auf ein einzelnes MC800 Geräte-Register oder einen kompletten Parameterblock (32 Parameter) zuzugreifen. Daher muss die "Menge (oder Anzahl) der Register" in der Modbus-Anforderung entweder 2 (für Einzelregisterzugriff) oder 64 (für den Zugriff auf einen kompletten Parameterblock) sein.

**Die detaillierte Zuordnung der MC800-Register zu den Halteregistern finden Sie in der Parametertabelle in Kapitel 4 dieses Handbuchs.**

Für allgemeine Parameter und Blockparameter ist es auch möglich, die Holding-Registernummer anhand der folgenden Formeln zu berechnen:

Zugriff auf allgemeine Parameter X:

$$\text{Holding Register number (low)} = 2 * X$$

$$\text{Holding Register number (high)} = 2 * X + 1$$

Zugriff auf Parameter X in Block Y:

$$\text{Holding Register number (low)} = 01C0 \text{ hex} + Y * 40 \text{ hex} + 2 * X$$

$$\text{Holding Register number (high)} = 01C0 \text{ hex} + Y * 40 \text{ hex} + 2 * X + 1$$



## 2.3.2 Read Coils und Write Single Coil

Mit den Funktionen "Read Coils" und "Write Single Coil" können einzelne Befehle gelesen und gesetzt / zurückgesetzt werden. Die Befehlsnamen sind Firmware-spezifisch. Der genaue Befehlsname ist in der Befehlstabelle der entsprechenden Firmware-Beschreibung zu finden.

Input signals (Commands)				
Command #	Coil #	Bit # in "Status of Serial Commands" (Holding Register 1602 / 1603 hex)	Hardware input (X4)	Anmerkung
Command 00	0	0	FAST IN 1	
Command 01	1	1	FAST IN 2	
Command 02	2	2	FAST IN 3	
Command 03	3	3	FAST IN 4	
Command 04	4	4	CTRL IN 1	
Command 05	5	5	CTRL IN 2	
Command 06	6	6	CTRL IN 3	
Command 07	7	7	CTRL IN 4	
Command 08	8	8	—	
Command 09	9	9	—	
Command 10	10	10	—	
Command 11	11	11	—	
Command 12	12	12	—	
Command 13	13	13	—	
Command 14	14	14	—	
Command 15	15	15	—	
Command 16	16	16	—	
Command 17	17	17	—	
Command 18	18	18	—	
Command 19	19	19	—	
Command 20	20	20	—	
Command 21	21	21	—	
Command 22	22	22	—	
Command 23	23	23	—	
Command 24	24	24	—	
Command 25	25	25	—	
Command 26	26	26	—	
Command 27	27	27	—	Nicht verwenden!
Select Test Var.	28	28	—	Nicht verwenden!
Store EEPROM	29	29	—	
Adjust Program	30	30	—	Nicht verwenden!
Test Program	31	31	—	Nicht verwenden!

## 2.3.3 Diagnose

Das Gerät unterstützt die Diagnose-Unterfunktion 00 "Return Query Data"  
Andere Diagnosefunktionen stehen nicht zur Verfügung.

## 2.4 Parametertabelle

Parameternamen sind Firmware-spezifisch; Die genauen Parameternamen entnehmen Sie bitte den Parametertabellen im entsprechenden MC800-Firmware-Beschreibung.

Alle MC800-Register sind 32-Bit-Register, daher ist jedes MC800-Register zwei Modbus Holding Registers zugeordnet.

Parameter	Modbus Holding Register (hex)		Data Type	Access*	Anmerkungen
General Parameter 00	0000	0001	Integer32	RW	
...	...	...	...	...	
General Parameter 31	003E	003F	Integer32	RW	
(not used)	...	...			
Block01 Parameter 00	0200	0201	Integer32	RW	
...	...	...	...	...	
Block01 Parameter 31	023E	023F	Integer32	RW	
Block02 Parameter 00	0240	0241	Integer32	RW	
...	...	...	...	...	
Block02 Parameter 31	027E	027F	Integer32	RW	
Block03 Parameter 00	0280	0281	Integer32	RW	
...	...	...	...	...	
Block03 Parameter 31	02BE	02BF	Integer32	RW	
Block04 Parameter 00	02C0	02C1	Integer32	RW	
...	...	...	...	...	
Block04 Parameter 31	02FE	02FF	Integer32	RW	
Block05 Parameter 00	0300	0301	Integer32	RW	
...	...	...	...	...	
Block05 Parameter 31	033E	033F	Integer32	RW	
Block06 Parameter 00	0340	0341	Integer32	RW	
...	...	...	...	...	
Block06 Parameter 31	037E	037F	Integer32	RW	
Block07 Parameter 00	0380	0381	Integer32	RW	
...	...	...	...	...	
Block07 Parameter 31	03BE	03BF	Integer32	RW	
Block08 Parameter 00	03C0	03C1	Integer32	RW	
...	...	...	...	...	
Block08 Parameter 31	03FE	03FF	Integer32	RW	

\*) RW = Read and write, RO = Read only

Fortsetzung "Parameter Table":

Parameter	Modbus Holding Register (hex)		Data type	Access	Anmerkungen
Block09 Parameter 00	0400	0401	Integer32	RW	
...	...	...	...	...	
Block09 Parameter 31	043E	043F	Integer32	RW	
Block10 Parameter 00	0440	0441	Integer32	RW	
...	...	...	...	...	
Block10 Parameter 31	047E	047F	Integer32	RW	
Block11 Parameter 00	0480	0481	Integer32	RW	
...	...	...	...	...	
Block11 Parameter 31	04BE	04BF	Integer32	RW	
Block12 Parameter 00	04C0	04C1	Integer32	RW	
...	...	...	...	...	
Block12 Parameter 31	04FE	04FF	Integer32	RW	
Counter Settings Par. 00	0500	0501	Integer32	RW	
...	...	...	...	...	
Counter Settings Par. 31	053E	053F	Integer32	RW	
Encoder Settings Par. 00	0540	0541	Integer32	RW	
...	...	...	...	...	
Encoder Settings Par. 31	057E	057F	Integer32	RW	
Communication Par. 00	0580	0581	Integer32	RW	
...	...	...	...	...	
Communication Par. 31	05BE	05BF	Integer32	RW	
Set-up Settings Par. 00	05C0	05C1	Integer32	RW	
...	...	...	...	...	
Set-up Settings Par. 31	05FE	05FF	Integer32	RW	
(Not used)	...				
Actual Value 00	1000	1001	Integer32	RO	
...	...	...	...	...	
Actual Value 31	103E	103F	Integer32	RO	
(Not used)	...				
Status of Hardware Inputs	1600	1601	Unsigned32	RO	
Status of Serial Commands	1602	1603	Unsigned32	RW	Zu verwenden, um Befehle über Modbus einzustellen
Status of Bus Commands	1604	1605	Unsigned32	RO	
Status of all Commands	1606	1607	Unsigned32	RO	
Status of Outputs	1608	1609	Unsigned32	RO	
Error Status	160A	160B	Unsigned32	RO	
Status 06 (reserved)	160C	160D	Unsigned32	RO	
...	...	...	...	...	
Status 15 (reserved)	161E	161F	Unsigned32	RO	

Fortsetzung "Parameter Table":

Parameter	Modbus Holding Register (hex)				Data type	Access	Anmerkungen
Firmware Version	2000	2001	2002	2003	String8	RO	
Firmware Type	2004	2005	2006	2007	String8	RO	
...	...	...	...	...	...	...	
(reserved)	2018	2019	201A	201B	String8	RO	
(Not used)	...						
Serial Number	3000	3001	3002	3003	String8	RO	
Unit Type	3004	3005	3006	3007	String8	RO	
...	...	...	...	...	...	...	
(reserved)	303C	303D	303E	303F	String8	RO	