

# Software S (Standard)

## DEUTSCH

### 1. Datenübertragung

Die Datenübertragung vom Funkmodul zu einer Steuerung (PC, Terminal) erfolgt über ein 1:1-Verbindungskabel und der RS232-Schnittstelle. Die Schnittstelle ist wie folgt zu parametrieren:

- 19200 Baud
- 8 Datenbit
- 1 Stopbit
- no Parity
- no Handshake

Die Parametrierung des Funkmoduls, d.h. die Auswahl einer Funk-Kanalnummer (0...49), erfolgt über die RS232-Schnittstelle und wird im Funkmodul nichtflüchtig gespeichert. Das Funkmodul befindet sich in ständiger Empfangsbereitschaft.

Das Funkmodul RTX500 kann mit 2 verschiedenen Kommunikationsprotokollen, dem **Service-Standard-Protokoll** oder dem **SIKONETZ3-Protokoll** betrieben werden. Bevor die Spannungsversorgung eingeschaltet wird, muss mit der Lötbrücke BR4 (siehe Benutzerinformation RTX500, 5.1 Anschluss) bestimmt werden, welches Schnittstellenprotokoll verwendet werden soll. Bei geschlossener Brücke (Auslieferungszustand) wird das Service-Standard-Protokoll, bei offener Brücke das SIKONETZ3-Protokoll verwendet.

Sobald ein gültiges Funktelegramm empfangen wurde, können die gewünschten Informationen über die RS232-Schnittstelle ausgelesen werden. Die Steuerung (PC, Terminal) gibt dabei einen Befehl, ggf. mit zusätzlichen Zahlenparametern aus. Das Funkmodul antwortet darauf entsprechend dem eingestellten Protokoll nach folgenden Tabellen.

#### 1.1 Kanalauswahl



Damit das Modul korrekt initialisiert werden kann, ist darauf zu achten, dass die RS232 seitens der Steuerung erst dann aktiviert wird, nachdem die 24V Versorgung angelegt wurde!

Über die Kanalnummer wird die Empfangsfrequenz des Systems eingestellt. Sie muss mit der beim Empfänger eingestellten Kanalnummer übereinstimmen. Gleichzeitig wird mit diesem Parameter auch die Sendeleistung des Systems vorgegeben.

### Frequenztabelle 868MHz-ISM-Band:

Sendeleistung dbm (mW)	Kanalnummer	Sende- bzw. Empfangsfrequenz [MHz]
+15 (32)	00...09	Beginn: 869.475 Ende: 869.575 Kanalabstand: 9kHz CE-Norm: 869.400 ... 869.650 (500mW <10% DutyCycle) <b>Achtung!</b> Bei dieser Einstellung können vereinzelt Oberwellen, der Nutzfrequenz, die zertifizierten Leistungsbereiche übersteigen.
+10 (10)	10...19	Beginn: 868.075 Ende: 868.525 Kanalabstand: 50kHz CE-Norm: 868.000 ... 868.600 (25mW <1% DutyCycle)
	20...29	Beginn: 869.350 Ende: 869.395 Kanalabstand: 5kHz CE-Norm: 869.300 ... 869.400 (10mW offener DutyCycle)
+5 (3.2)	30...39	Beginn: 869.750 Ende: 869.975 Kanalabstand: 25kHz CE-Norm: 869.700 ... 870.000 (5mW 10 ... 100% DutyCycle)
+0 (1)	40...49	

### Frequenztabelle 915MHz-ISM-Band:

Sendeleistung dbm (mW)	Kanalnummer	Sende- bzw. Empfangsfrequenz [MHz]
+10 (10)	00...09	Beginn: 903.000 Ende: 907.500 Kanalabstand: 500kHz
	10...19	Beginn: 908.000 Ende: 912.500 Kanalabstand: 500kHz
	20...29	Beginn: 913.000 Ende: 917.500 Kanalabstand: 500kHz
	30...39	Beginn: 918.000 Ende: 922.500 Kanalabstand: 500kHz
	40...49	Beginn: 923.000 Ende: 927.500 Kanalabstand: 500kHz

FCC-Norm: 902.000 ... 928.000MHz.

## 2. Datenübertragung zur Steuerung

### 2.1 Service-Standard-Protokoll

Alle Antworttelegramme werden mit einem CR (= Ox13) vervollständig. Ungültige Eingaben werden mit einem "?" beantwortet.

Befehl	Länge	Antwort	Bemerkungen
A0	2/12	xxxxxxxxxx>	Hardware Kennung ausgeben
A1	2/9	xxxxxxx>	Firmware Version ausgeben
A2	2/9	xxxxxxx>	Sendefrequenz ausgeben
A3	2/12	xxxxxxxxxx>	Applikation ausgeben
C	1/15	+yyyyyyyy_ zzz_0xss>	y = Positionswert _ = Leerzeichen z = Senderadresse s = Statusbyte des letzten empfangenen Telegrammes in hex-Darstellung
0y	2/5	xxx>	Parameter ausgeben y = 5: Funkkanalnummer x = dezimaler Wert

Befehl	Länge	Antwort	Bemerkungen
Pyyx	5/2	>	Parameter eingeben y = 5: Funkkanalnummer x = dezimaler Wert (000 .. 049)
Syyyyy	6/2	>	y = 11100: Werkseinstellung wiederherstellen
U	1/6	0xyy>	Statusregister des letzten empfangenen Telegramms ausgeben. Entfällt ab Softwarestand V0.05.
Z	1/11	+xxxxxxxx	Positionswert des letzten empfangenen Telegramms ausgeben.

## 2.2 Protokollbeschreibung SIKONETZ3

Das Protokoll ist als Master-Slave-System aufgebaut. Der Sensor hat nur Slave-Funktion. Es existieren 2 Telegrammlängen:

3 Byte:

Adress-Byte	Befehl	Prüf-Byte

6 Byte:

Adress-Byte	Befehl	Daten-Byte Low	Daten-Byte Middle	Daten-Byte High	Prüf-Byte

Das Adressbyte setzt sich wie folgt zusammen:

1	0	A0	A1	A2	A3	A4	0	RR	L	1
	Start									Stopp

Das Prüfbyte wird als EXOR-Verknüpfung der restlichen 2 bzw. 5 Bytes des Telegramms erzeugt.

A0 ... A4: Binärkodierte Adresse 1 ... 31;

Adresse 0 definiert für Master

RR: Rundruf-Bit = 1; Befehl gilt für alle angeschlossenen Slaves, Slaves antworten nicht

L: Längen-Bit: 1 = Kurztelegramm (3 Byte)  
0 = Langtelegramm (6 Byte)

Spalte	Erläuterung
Hex:	Hexadezimalwert des Befehls.
TX:	Telegrammlänge vom Master an Sensor.
RX:	Telegrammlänge vom Sensor an Master.
S:	Übergebener Parameter wird nichtflüchtig im Sensor gespeichert.
P:	Für diesen Befehl ist es notwendig, den Programmiermode einzuschalten (Bef 0x32; 0x33).
R:	Dieser Befehl ist rundruffähig.

Hex	TX	RX	S	P	R	Funktion
0x16	3	6	-	-	-	Positionswert auslesen
0x18	3	6	-	-	-	Kalibrierwert auslesen
0x1b	3	6	-	-	-	Geräteerkennung auslesen D-Byte 1: Kennung 23 D-Byte 2: Softwareversion D-Byte 3: Hardwareversion

Hex	TX	RX	S	P	R	Funktion
0x1d	3	6	-	-	-	Zählrichtung auslesen Wert = 0: "auf" (+) Wert = 1: "ab" (-)
0x28	6	6	S	P	-	Kalibrierwert programmieren Wert auf den der Positionswert gesetzt wird, wenn der Sensor geleert wird (Bef 0x48).
0x2d	6	6	S	P	-	Zählrichtung programmieren Wert = 0: "auf" (+) Wert = 1: "ab" (-)
0x32	3	3	-	-	-	Programmiermode "Ein" Programmiermode muss "Ein" sein, um Parameter (0x28 und 0x2d) zu programmieren.
0x33	3	3	-	-	-	Programmiermode "Aus" Default
0x3a	3	6	-	-	-	Systemstatus ausgeben
0x3b	3	3	-	-	-	Systemstatus löschen Systemstatus Bytes 2 und 3 werden gelöscht.
0x48	3	3	S	P	-	Sensor nullen Positionswert wird auf Kalibrierwert gesetzt.
0x4f	3	3	-	-	R	Positionswert einfrieren. Positionswert wird eingefroren. Zustand wird durch Auslesen des Positionswertes zurückgesetzt. Dient zum synchronisierten Auslesen mehrerer Sensoren.

## Fehlermeldungen

Das Funkmodul erkennt Übertragungs- bzw. Eingabefehler und sendet folgende Fehlermeldungen:

Hex	TX	RX	S	P	R	Funktion
0x82	-	3	-	-	-	Datenübertragungsfehler Prüfsumme
0x83	-	3	-	-	-	Unzulässiger oder unbekannter Befehl
0x85	-	3	-	-	-	Unzulässiger Wert (Parameter Programmierung)

## Synchronisation:

Eine Byte-/Telegrammsynchronisation erfolgt über "Timeout": Der Abstand der einzelnen Bytes eines Telegramms dürfen einen Wert von **10ms** nicht übersteigen. Falls ein angesprochener Sensor nicht antwortet, so darf der Master frühestens nach **30ms** erneut ein Telegramm senden.

## Telegrammbeispiel:

Master fordert Positionswert des Slaves 7 an.

Master sendet (hex): 87 16 91

Kurztelegramm an Adresse 7; Befehl 16; Prüfbyte 91H.

Slave antwortet (hex): 07 16 03 02 00 10

Langtelegramm von Adresse 7; Befehl 16; Pos. Wert 203H = 515d; Prüfsumme 10H.



See enclosed page with user information RTX500

# Software S (Standard)

ENGLISH

## 1. Data transmission

The data is transmitted from the radio transmitter/receiver to a control unit (PC, terminal) via a 1:1 connection cable and the RS232 interface. The interface should be configured as follows:

- 19200 Baud
- 8 Datenbit
- 1 Stopbit
- no Parity
- no Handshake

Parameterization of the radio transmitter/receiver, i.e., selection of a radio channel number (0...49) is done via the RS232 interface and is stored in the non-volatile memory of the radio transmitter/receiver. The radio transmitter/receiver is permanently ready to receive.

The radio transmitter/receiver, RTX500, can be operated with 2 different communication protocols, the Service Standard Protocol or the SIKONETZ3 protocol. Before switching on the voltage supply, it is necessary to determine by means of the solder bridge BR4 (see user information RTX500, 5.1 Connection) which interface protocol is to be used. The Service Standard Protocol is used when the bridge is closed (default setting) and the SIKONETZ3 protocol is used when the bridge is open).

As soon as a valid radio telegram has been received, the desired information can be read out via the RS232 interface. The control unit (PC, terminal) outputs a command, possibly associated with additional numeral parameters. The radio transmitter/receiver replies in compliance with the set protocol according to the tables below.

### 1.1 Channel selection



For correct initialization of the module, care should be taken that the control unit activates the RS232 interface only after applying the 24V supply!

The reception frequency of the system is set via the channel number. It must correspond with the channel number set at the receiver side. At the same time, this parameter determines the system's transmitting power.

## Frequency table 868MHz-ISM band:

Transmit power dbm (mW)	Channel no.	Transmitting frequency [MHz]
+15 (32)	00...09	Start: 869.475 End: 869.575 channel spacing: 9kHz CE-standard: 869.400 ... 869.650 (500mW <10% dutycycle) <b>Attention!</b> With this setting some single harmonics may be higher than the accredited range of performance.
+10 (10)	10...19	Start: 868.075 End: 868.525 channel spacing: 50kHz CE-standard: 868.000 ... 868.600 (25mW <1% dutycycle)
	20...29	Start: 869.350 End: 869.395 channel spacing: 5kHz CE-standard: 869.300 ... 869.400 (10mW open dutycycle)
+5 (3.2)	30...39	Start: 869.750 End: 869.975 channel spacing: 25kHz
+0 (1)	40...49	CE-standard: 869.700 ... 870.000 (5mW 10 ... 100% dutycycle)

## Frequency table 915MHz-ISM band:

Transmit power dbm (mW)	Channel no.	Transmitting frequency [MHz]
	00...09	Start: 903.000 End: 907.500 channel spacing: 500kHz
	10...19	Start: 908.000 End: 912.500 channel spacing: 500kHz
+10 (10)	20...29	Start: 913.000 End: 917.500 channel spacing: 500kHz
	30...39	Start: 918.000 End: 922.500 channel spacing: 500kHz
	40...49	Start: 923.000 End: 927.500 channel spacing: 500kHz

FCC-Norm: 902.000 ... 928.000MHz.

## 2. Data transfer to the control unit

### 2.1 Service-Standard-protocol

All reply telegrams are completed by CR (= 0x13). Invalid entries are replied to with "?".

Com.	Length	Response	Description
A0	2/12	xxxxxxxxxx>	Issue hardware version
A1	2/9	xxxxxxx>	Issue firmware version
A2	2/9	xxxxxxx>	Issue transmitting frequency
A3	2/12	xxxxxxxxxx>	Issue application
C from software release V0.05	1/15	+yyyyyyyyy_ zzz_0xxx>	y = actual position value _ = blank z = sender address s = Status byte in hex design of the last received telegram.
Oy	2/5	xxx>	Issue parameters y = 5: radio channel number x = decimal value
Pyxx	5/2	>	Enter parameter y = 5: radio channel number x = decimal value (000 .. 049)
Syyyyy	6/2	>	y = 11100: restore factory settings
U	1/6	0xyy>	output status register of the last received telegram. Dropped from software release V0.05



Com.	Length	Response	Description
Z	1/11	+xxxxxxx	output position value of last received telegram.

## 2.2 List of commands SIKONETZ3

The protocol is set up as a master-slave system. The sensor has only slave functions. There exist 2 telegram lengths:

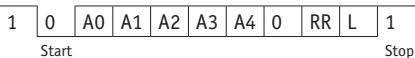
3 Byte:

Address Byte	Command	Check Byte

6 Byte:

Address Byte	Command	Data Byte Low	Data Byte Middle	Data Byte High	check Byte

The address byte is composed as follows:



The test byte results from an EXOR-interconnection of the remaining two or five bytes of the telegram.

A0 ... A4: binary coded address 1 ... 31;

address 0 defined for master

RR: broadcast Bit = 1; command is effective for all connected slaves, slaves do not reply

L: length Bit: 1 = short telegram (3 bytes)  
0 = long telegram (6 bytes)

Column	Signification
Hex:	Hexadezimal value of the command.
TX:	Length of telegram from master to sensor.
RX:	Length of telegram from sensor to master.
S:	Transmitted parameter is permanently stored in the sensor.
P:	For this command programming mode has to be activated (command 0x32; 0x33).
R:	This command can be broadcasted.

Hex	TX	RX	S	P	R	Function
0x16	3	6	-	-	-	read out position value
0x18	3	6	-	-	-	read out calibration value
0x1b	3	6	-	-	-	read out device's characteristics D-Byte 1: identifier 23 D-Byte 2: software version D-Byte 3: hardware version
0x1d	3	6	-	-	-	read out counting direction value = 0: "up" (+) value = 1: "down" (-)
0x28	6	6	S	P	-	program calibration value Value to which the position value is set when the device is calibrated (command 0x48).
0x2d	6	6	S	P	-	program counting direction value = 0: "up" (+) value = 1: "down" (-)

Hex	TX	RX	S	P	R	Function
0x32	3	3	-	-	-	programming mode "ON"; Programming mode must be "ON" to be able to program (0x28 nd 0x2d).
0x33	3	3	-	-	-	programming mode "OFF" Default
0x3a	3	6	-	-	-	send system status
0x3b	3	3	-	-	-	cancel system status System status bytes 2 and 3 are being deleted.
0x48	3	3	S	P	-	zero the sensor; Position value is set to calibration value.
0x4f	3	3	-	-	R	freeze position value Position value is frozen; deactivated when positional value is read out. Serves for synchronized reading out of several sensors.

## Error messages

The radio transmitter/receiver recognizes transmission or input errors and then issues the following error messages:

Hex	TX	RX	S	P	R	Function
0x82	-	3	-	-	-	check sum data transmission error
0x83	-	3	-	-	-	invalid or unknown command
0x85	-	3	-	-	-	invalid value (parameter programming)

## Synchronisation:

Byte/ telegram synchronisation is made via "Time-out": the distance between each byte of a telegram must not exceed **10ms**. If a sensor does not respond, the master may only send another telegram after **30ms** at the earliest.

## Telegram example:

Master requests position value of Slave 7.

*Master sends (hex):* 87 16 91

Short telegram to address 7; command 16; check byte 91H.

*Slave replies (hex):* 07 16 03 02 00 10

Long telegram from address 7; command 16; position value 203H = 515d; checksum 10H.

## SIKO GmbH

### Werk/ Factory:

Weihermattenweg 2  
79256 Buchenbach-Unteribental

### Postanschrift/ Postal address:

Postfach 1106  
79195 Kirchzarten

**Telefon/Phone** +49 7661 394-0

**Telefax/Fax** +49 7661 394-388

**E-Mail** info@siko.de

**Internet** www.siko.de

**Service** support@siko.de

