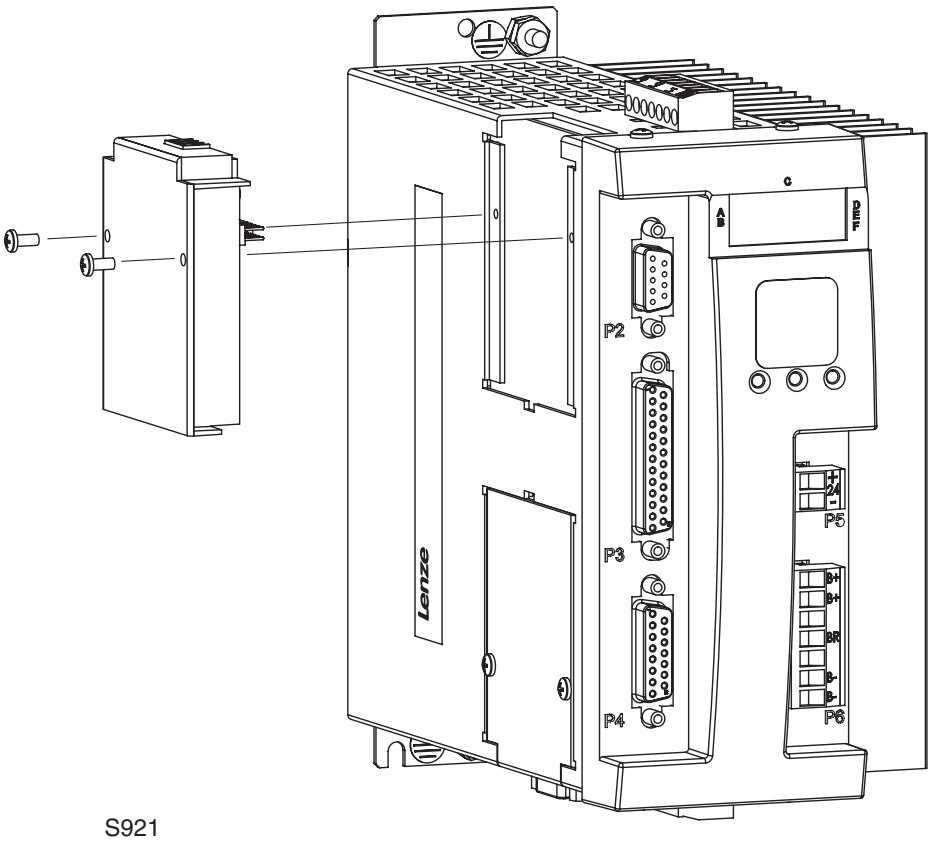
Lenze



S921

RS-485 module
Modbus

S94MOD01A

Table of Contents

1 Preface and General Information

1.1 How to use these Operating Instructions

- These Operating Instructions are intended for safety-relevent opertion on and with the module. They contain safety information which must be observed.
- All personnel working on and with the module must have these Operating Instructions available and observe the information and notes relevent for them.
- These instructions are only valid in combination with the Operating Instructions of the corresponding controller. They must always be compete and in a perfectly readable state.

2 Safety Information

2.1 Persons responsible for safety

Operator

- An operator is any natural or legal person who uses the drive system or on behalf of whom the drive system is used.
- The operator or his safety personnel is obliged to ensure
  - the compliance with all relevant regulations, instructions, and legislation.
  - that only skilled personnel works on and with the drive system.
  - that the personnel has the Operating Instructions available for all corresponding works.
  - that all unqualified personnel are prohibited from working on and with the drive system.

Qualified personnel

Qualified personnel are persons who - because of their education, experience, instructions, and knowledge about corresponding standards and regulations, rules for the prevention of accidents, and operating conditions - are authorized by the person responsible for the safety of the plant to perform the required actions and who are able to recognize potential hazards.

(Definition for qualified personnel to VDE 105 or IEC 364)

2.2 General safety information

- These safety notes do claim to be complete. In case of questions and problems please contact your Lenze representative.
- At the time of delivery the drive system meets the state of the art and ensures basically safe operation.
- The indications given in these Operating Instructions refer to the stated hardware and software versions of the controller.
- The controller is hazerdous if:
  - unqualified personnel works on and with the controller.
  - the controller is used inappropriately.
- Ensure by appropriate measures that neither personal injury nor damage to property may occur in the event of failure of the drive system.
- The drive system must only be operated when no faults occur.
- Retrofitting, modifications, or redesigns are basically prohibited. Lenze must be contacted in all cases.

3 Technical data

3.1 Related documents

- MODBUS Application Protocol Specification V1.1
  It can be found at: http://www.modbus.org/default.htm
- MODBUS over Serial Line Specification & Implementation guide V1.0

3.2 General Modbus protocol description

The MODBUS protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers.
There are some additional data unit (ADU) fields introduced by the network layer

ADU			
1 Byte Address	1 Byte Function Code	Data	2 Bytes CRC
	PDU		

Figure 3: RS485 Network MODBUS frame

The client (also named Master) that initiates a MODBUS transaction builds the MODBUS application data unit. The function indicates to the server what kind of action to perform. The MODBUS application protocol establishes the format of a request initiated by a client.

The function code field of a MODBUS data unit is coded in one byte. Valid codes are in the range of 1...255 decimal (128...255 reserved for exception responses). When a message is sent from a Client to a Server device the function code field tells the server what kind of action to perform.

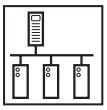
Sub-function codes are added to some function codes to define multiple actions.

The data field of messages sent from a client to server devices contains additional information that the server uses to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field.

The data field may be nonexistent (of zero length) in certain kinds request, in this case the server does not require any additional information. The function code alone specifies the action. If no error occurs related to the MODBUS function requested in a properly received MODBUS ADU the data field of a response from a server to a client contains the data requested. If an error related to the MODBUS function requested occurs, the field contains an exception code that the server application can use to determine the next action to be taken.

For example a client can read the ON / OFF states of a group of discrete outputs or inputs or it can read/write the data contents of a group of registers.

When the server responds to the client, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the server simply echoes the original function code.



3.3 Configuration

Drive 94 supports Modbus communication protocol through its RS485 optional card.

The following Communication parameters are available:

RS485 configuration – If the value of this parameter is ‘Modbus slave’ the modbus slave protocol is enabled on the RS485 port. If the value is ‘Normal’ the RS485 works in PPP mode.

Modbus baud rate – the RS485 baud rate in modbus mode. Note that modbus requires 2 stop bits and no parity.

Modbus reply delay – The delay introduced after receiving modbus request and before sending a reply. Note that this delay will always be >= 3.5 characters as required by the modbus specification. Some modbus master devices are slow and an increase of the ‘Modbus reply delay’ value might be required to successfully work with these devices.

Each device in the Modbus network must have its own unique network address.

The ‘Addr’ submenu on the drive display and the front panel buttons can be used to set the Modbus network address.

Drive 94 is Modbus slave devices in a single master multiple (or one) slaves network.
(Sometimes Slave devices are named servers because they wait for requests.)

Drive 94 communicates on the other end with Modbus Master.

Most Terminals can be configured as Masters with a Modbus Generic driver.

3.4 Supported Modbus features

Drive 94 uses the Modbus protocol delay of 3.5 characters time between packets to determine the end of packet.

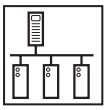
The following ModBus functions are supported:

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers

When the drive receives ModBus function it executes it and replies with Normal message, Exception Message or No message as described by the Modbus Application protocol Specification V1.1.

Modbus broadcast address is supported by the drive 94 also. In this case after receiving correct Modbus request the drive executes it but does not send any reply back.

The DWORD (double word) and FLOAT numbers are send with the LOW WORD FIRST convention by the 94 drive.
Note that some Terminals may have to be configured appropriately in order to accept such a WORD order.



4 Parameter setting

4.1 Register Memory

4.1.1 Read Only or Input Registers

The following read only ModBus registers (drive variables) accessible only through the ModBus function

- 03 (0x03) Read Holding Registers

4.1.2 Read/Write or Holding Registers

The following read/write ModBus registers (drive variables) accessible only through the ModBus functions

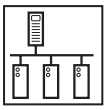
- 04 (0x04) Read Input Registers
- 06 (0x06) Write Single Register
- 16 (0x10) Write Multiple Registers

For all ModBus registers that are denoted in description field as 'W: EPROM & memory' exist another register with address+200 (203 for CurrentLimit, for example) to write only runtime memory location of that variable, without affecting EEPROM value. Read command returns the same result for both addresses.

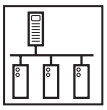
Parameter changes in EEPROM and memory using first set of addresses are permanent, however parameter changes in memory only using second set of addresses are valid only for one session. Session ends by powering the drive down.

Remember, when changing double word or float parameters, the change is applied to the drive only after second word is written.

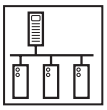
ModBus Register 16 bit address	Type	Controller Variable	Description
1	16 bit word	DriveMode	See Drive 94 manual W: EPROM & memory.
2	16 bit word	PwmIndex	See Drive 94 manual W: EPROM & memory.
3	32 bit float	CurrentLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
5	32 bit float	PeakCurrentLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
7	32 bit float	LowPeakCurrentLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
9	16 bit word	InvertAnalogInput	See Drive 94 manual W: EPROM & memory.
10	32 bit float	CurrentAnalogInput	See Drive 94 manual W: EPROM & memory. Low word starts first



ModBus Register 16 bit address	Type	Controller Variable	Description
12	32 bit float	VelocityAnalogInput	See Drive 94 manual W: EPROM & memory. Low word starts first
14	16 bit word	AccDecFlag	See Drive 94 manual W: EPROM & memory.
15	32 bit float	AccelLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
17	32 bit float	DecelLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
19	16 bit int	GearRatioTop	See Drive 94 manual W: EPROM & memory.
20	16 bit word	GearRatioBottom	See Drive 94 manual W: EPROM & memory.
21	16 bit word	StepInputType	See Drive 94 manual W: EPROM & memory.
22	16 bit word	FaultReset	See Drive 94 manual W: EPROM & memory.
23	16 bit word	FeedbackLoss	See Drive 94 manual W: EPROM & memory.
24	16 bit word	ReferenceType	See Drive 94 manual W: EPROM & memory.
25	16 bit word	MotorTempSensorRes	See Drive 94 manual W: EPROM & memory.
26	16 bit word	MotorTempSensorEnable	See Drive 94 manual W: EPROM & memory.
27	16 bit word	RegenDutyCycle	See Drive 94 manual W: EPROM & memory.
28	16 bit word	EncoderRepeatSource	See Drive 94 manual W: EPROM & memory.
29	32 bit float	TorqueVelocityLimit	See Drive 94 manual W: EPROM & memory. Low word starts first
31	16 bit int	SeGearRatioTop	See Drive 94 manual W: EPROM & memory.
32	16 bit word	SeGearRatioBottom	See Drive 94 manual W: EPROM & memory.
33	16 bit word	SecondEncoderEnable	See Drive 94 manual W: EPROM & memory.
34	32 bit float	SeIGain	See Drive 94 manual W: EPROM & memory. Low word starts first
36	16 bit word	SePosErr	See Drive 94 manual W: EPROM & memory.



ModBus Register 16 bit address	Type	Controller Variable	Description
37	32 bit float	SeMaxErrTime	See Drive 94 manual W: EPROM & memory. Low word starts first
39	16 bit word	AnalogOutput	See Drive 94 manual W: EPROM & memory.
40	32 bit float	CurrentScale	See Drive 94 manual W: EPROM & memory. Low word starts first
42	32 bit float	VelocityScale	See Drive 94 manual W: EPROM & memory. Low word starts first
44	16 bit word	AnalogBand	See Drive 94 manual W: EPROM & memory.
45	16 bit word	AnalogOffset	See Drive 94 manual W: EPROM & memory.
46	16 bit word	ZeroSpeed	See Drive 94 manual W: EPROM & memory.
47	16 bit word	SpeedWindow	See Drive 94 manual W: EPROM & memory.
48	16 bit word	AtSpeed	See Drive 94 manual W: EPROM & memory.
49	16 bit word	PosErr	See Drive 94 manual W: EPROM & memory.
50	32 bit float	MaxErrTime	See Drive 94 manual W: EPROM & memory. Low word starts first
52	32 bit float	VelPGain	See Drive 94 manual W: EPROM & memory.
54	32 bit float	VelIGain	See Drive 94 manual W: EPROM & memory.
56	32 bit float	PosPGain	See Drive 94 manual W: EPROM & memory.
58	32 bit float	PosIGain	See Drive 94 manual W: EPROM & memory. Low word starts first
60	32 bit float	DGain	See Drive 94 manual W: EPROM & memory.
62	32 bit float	ILimit	See Drive 94 manual W: EPROM & memory.
63	16 bit word	VelocityRegWnd	See Drive 94 manual W: EPROM & memory.
65	16 bit word	In2Func	See Drive 94 manual W: EPROM & memory.
66	16 bit word	In2Polarity	See Drive 94 manual W: EPROM & memory.
67	16 bit word	InBounceDelay for input 1	See Drive 94 manual W: EPROM & memory.



ModBus Register 16 bit address	Type	Controller Variable	Description
68	16 bit word	InBounceDelay for input 2	See Drive 94 manual W: EPROM & memory.
69	16 bit word	OutFunc for output 1	See Drive 94 manual W: EPROM & memory.
70	16 bit word	OutFunc for output 2	See Drive 94 manual W: EPROM & memory.
71	16 bit word	OutPolarity for output 1	See Drive 94 manual W: EPROM & memory.
72	16 bit word	OutPolarity for output 2	See Drive 94 manual W: EPROM & memory.
73	16 bit word	ControlWord	2 – Quick Stop 3 – Enable Drive 4 – Disable Drive 5 – Fault Reset 6 - Continue W: Memory only
75	32 bit dword	TargetVelocity	This register could provide the Target Velocity when the drive is in normal velocity mode. W: Memory only. Low word starts first
77	16 bit word	TargetTorque	This register could provide the Target Torque when the drive is in normal velocity mode. W: Memory only. Low word starts first

4.2 Discrete Memory

4.2.1 Discrete Inputs (Read Only Bits)

The following read only ModBus bit registers (drive bit variables) accessible only through the ModBus function

- 02 (0x02) Read Discrete Inputs

4.2.2 Coils (Read/Write Bits)

The following read/write ModBus bit registers (drive bit variables) accessible only through the ModBus functions

- 01 (0x01) Read Coils
- 05 (0x05) Write Single Coil
- 15 (0x0F) Write Multiple Coils

Modbus Register 16 bit address	Type	Description
1 (0x0001)	BIT	Digital OUT1
2 (0x0002)	BIT	Digital OUT2

Lenze

AC Technology Corporation • 630 Douglas Street • Uxbridge, MA 01569 • USA
☎ +1 (508) 278-9100