SIC184 MODBUS-RTU Protocol (v3.00)

Contents

1. Introduction

The SIC184 controller provides RS485 serial interface, which ensure communication with external devices via MODBUS-RTU protocol.
Connection with RS485 port between controllers and MASTER device, especially at bigger distances and speed transmissions (>38400bps, >10m) should be made using twisted pair cable (shielded twisted pair as best solution). It should be noticed to add terminator (120Ω…470Ω resistor, connected between A and B lines) at the beginning and the ending of RS485 bus.

The controller provides communication with Master devices with data rate up to 100 frames per second (fps).

Firmware version.
This data concerns driver with 3.00 software or latest.

Registers addresses can change at newer versions of driver software. Please check if driver software version correspond hereby described version.

2. Transmission parameters and MODBUS functions

Transmission parameters

- Default address: 1
- Default baudrate: **38400 b/s**
- Stop bits: **1**, Parity: **none**
- Timeout: **750µs** (maximum time interval between next bytes in frame)

Provided MODBUS function

Function no. (hex)	Description
0x01	Read output
0x02	Read input
0x03	Read N registers (for WORD, INT, DIN, REAL)
0x05	Write input
0x06	Write 1 register (for WORD, INT)
0x10	Write N registers (for DINT, REAL)

Description of variable types used in SIC184's MODBUS-RTU

Variable name	Description	Byte size / registers	Range
BYTE	8 bit (1-byte)	1 / 1	0-255
WORD	16 bit (2-byte)	2 / 1	0…32768
INT	Signed integer (2-byte)	2 / 1	-32768…32767
DINT	Double integer (4-byte)	4 / 2	-2^{31} …$(2^{31}-1)$
REAL	Floating points	4 / 2	$1.18*10^{-38}$…$3.40*10^{38}$, 0, $-3.40*10^{38}$ … $-1.18*10^{-38}$

Values entered into registers are not saved - values will return to default after re-turning on power supply.

3. List of registers

3.1 Write and read inputs, outputs and bit registers

Address	Name	Variable type	Mode (MODBUS function No.)	Description
6000 6000…6001	OUT	BYTE	R (0x01), W (0x05)	Read outputs OUT1…OUT2 Set outputs OUT1…OUT2
5000…5008	IN	BYTE	R (0x02)	Read inputs IN1…IN9
2000	ENABLE	BYTE	W (0x05)	Enable/disable drive (output EN =1/0)
2001	STOP	BYTE	W (0x05)	Drive stop
2002	POS_RESET	BYTE	W (0x05)	Reset drive position
2003	JOG_PLUS	BYTE	W (0x05)	JOG+ mode (velocity set by JOG_SPEED register)
2004	JOG_MINUS	BYTE	W (0x05)	JOG- mode

NOTE: In HMI devices, write/read functions of bit inputs are usually labeled as **x1**. Write/read functions of bit inputs are usually labeled as **x0**.

Table example of data for read input IN1 state (address = 5000)

Query (MASTER -> SIC184)		Response (SIC184-> MASTER)	
Device address	0x01	Device address	0x01
Function	0x02	Function	0x02
Initial HI address	**0x13**	No. of bytes	0x01
Initial LO address	**0x88**	Inputs state	BYTE
No. of HI inputs	0x00	CRC HI	BYTE
No. of LO inputs	**0x08**	CRC LO	BYTE
CRC HI	FD		
CRC LO	62		

Table example of data for output OUT2 settings (address = 6001)

Query (MASTER -> SIC184)		Response(SIC184-> MASTER)	
Device address	0x01	Device address	0x01
Function	0x05	Function	0x05
Initial HI address	**0x17**	Initial HI address	**0x17**
Initial LO address	**0x71**	Initial LO address	**0x71**
No. of HI inputs	0x00	No. of HI inputs	0x00
No. of LO inputs	**0x00**	No. of LO inputs	**0x00**
CRC HI	98	CRC HI	98
CRC LO	65	CRC LO	65

3.2 Write and read numerical registers

User registers for general purpose

Address	Name	Variable type	Mode (MODBUS function No.)	Description
0…498	USER_REGISTER	WORD DIN REAL	R (0x03), W (0x06) R (0x03), W (0x10) R (0x03), W (0x10)	User registers.

Control registers WORD

Address	Name	Variable type	Mode (MODBUS function No.)	Description
1000	PROG_BANK_SEL	WORD	R (0x03), W (0x06)	Selection of program bank number to run
1001	PROG_CTRL	WORD	R (0x03), W (0x06)	Program control: 0 – STOP, 1 – START, 2 - PAUZA
1002	STATUS	WORD	R (0x03)	Drive state: 0 - drive turned off (EN signal inactive) 1 – drive turned on, no motion (EN signal active)

				2 – drive in set velocity mode 3 – drive in motion to set position mode 4 – drive achieved the set position 5 – error of achieving set position (for operation with encoder) 6 – drive in homing mode 8 – drive in position correction mode (for operation with encoder) 9 - drive achieved limit position L while motion towards negative position value (by program or proximity sensor signal KL) 10 - drive achieved limit position R while motion towards positive position value (by program or proximity sensor signal KR)
1003	POWER	WORD	R (0x03), W (0x06)	Drive current (2…100% of max. value)
1004	POWER_RED	WORD	R (0x03), W (0x06)	Drive current reduction (2…100% of max. value)
1005	JOG_SPEED	WORD	R (0x03), W (0x06)	Percentage velocity for JOG mode (1..100% VMAX)

NOTE: In HMI devices, write/read functions of numerical values are usually labeled as **x4**.

Control registers DINT, REAL

Address	Name	Variable type	Mode (MODBUS function No.)	Description
REGISTERS WITH DINT VALUES Values for drive in pulses.				
Current values and ramp parameters				
1020	POS_ACT_INT	DINT	R (0x03)	Drive current position in pulses
1022	POS_NEW_INT	DINT	R (0x03), W (0x10)	Save current position to drive. **Setting 0 value causes reset of current drive position.**
1024	VEL_ACT_INT	DINT	R (0x03)	Read current drive velocity.
1026	ACC_INT	DINT	R (0x03), W (0x10)	Write/read acceleration in position mode and acceleration | braking in velocity mode.
1028	DEC_INT	DINT	R (0x03), W (0x10)	Write/read braking in velocity mode.
1030	VMAX_INT	DINT	R (0x03), W (0x10)	Write/read maximum velocity for position mode.
1032	POSLIM_L_INT	DINT	R (0x03), W (0x10)	Write/read L end position (for motion towards "-")
1034	POSLIM_R_INT	DINT	R (0x03), W (0x10)	Write/read R end position (for motion towards "+")
Registers for motion setting				
1036	HOME_INT	DINT	W (0x10)	Homing execution. Value of register determines homing velocity.
1038	VEL_ABS_INT	DINT	W (0x10)	Set absolute velocity (drive velocity will be the same as entered value).
1040	VEL_REL_INT	DINT	W (0x10)	Set relative velocity (drive velocity will be equal to current + entered velocity).
1042	POS_ABS_INT	DINT	W (0x10)	Set absolute position (drive motion occur until achieved set position).
1044	POS_REL_INT	DINT	W (0x10)	Set relative position (drive motion occur until position will be equal to current + set value).
Registers for encoders				
1046	ENC_XPOS_INT	DINT	R (0x03), W (0x10)	Counter value of pulses from encoder converted into drive rotation.
1048	ENC_IMP_INT	DINT	R (0x03), W (0x10)	Counter value of pulses from encoder.
REGISTERS WITH REAL VALUES Values for drive in set units.				
Current values and ramp parameters and motion				
1060	POS_ACT_REAL	REAL	R (0x03)	Current position.
1062	POS_NEW_ REAL	REAL	R (0x03), W (0x10)	Save current position to drive. **Setting 0 value causes reset of current drive position.**
1064	VEL_ACT_ REAL	REAL	R (0x03)	Read current drive velocity.
1066	ACC_ REAL	REAL	R (0x03), W (0x10)	Write/read acceleration in position mode and acceleration | braking in velocity mode.
1068	DEC_ REAL	REAL	R (0x03), W (0x10)	Write/read braking in velocity mode.

1070	VMAX_ REAL	REAL	R (0x03), W (0x10)	Write/read maximum velocity for position mode.
1072	POSLIM_L_ REAL	REAL	R (0x03), W (0x10)	Write/read L end position (for motion towards "-")
1074	POSLIM_R_ REAL	REAL	R (0x03), W (0x10)	Write/read R end position (for motion towards "+")
Registers for motion setting				
1076	HOME_ REAL	REAL	W (0x10)	Homing execution. Value of register determines homing velocity.
1078	VEL_ABS_ REAL	REAL	W (0x10)	Set absolute velocity (drive velocity will be the same as entered value).
1080	VEL_REL_ REAL	REAL	W (0x10)	Set relative velocity (drive velocity will be equal to current + entered velocity).
1082	POS_ABS_ REAL	REAL	W (0x10)	Set absolute position (drive motion occur until achieved set position).
1084	POS_REL_ REAL	REAL	W (0x10)	Set relative position (drive motion occur until position will be equal to current + set value).
Registers for encoders				
1086	ENC_XPOS_ REAL	REAL	R (0x03), W (0x10)	Counter value of pulses from encoder converted into drive rotation.

The SIC184 is addressing registers starts with 0. For MASTER devices which addressing starts with 1, register values should be entered with shift by 1 e.g. POS_ACT_INT = 1020 + 1 = 1021

Table example of settings for device M1 absolute velocity. Function: 0x10, Register address: 1078 (VEL_ABS_REL)

Write (MASTER -> SIC84)		Response (SIC184 -> MASTER)	
Device address	0x01	Device address	0x01
Function	0x10	Function	0x10
Register Hi address	**0x04**	Initial HI address	0x04
Register Lo address	**0x36**	Initial LO address	**0x36**
No. of HI registers	0x00	No. of HI registers	0x00
No of LO registers	0x02	No. of LO registers	**0x02**
No. of bytes	0x04	CRC	16 bit
Register 0x06 Hi	REAL/DINT* (Byte 1)		
Register 0x06 Lo	REAL/DINT* (Byte 0)		
Register 0x06 +1 Hi	REAL/DINT* (Byte 3)		
Register 0x06 +1 Lo	REAL/DINT* (Byte 2)		
CRC	**16 bit**		

Table example with read current position of M1 drive. Function: 0x03, Register address: 1060 (POS_ACT_REAL)

Query (MASTER -> SIC184)		Response (SIC184-> MASTER)	
Device address	0x01	Device address	0x01
Function	0x03	Function	0x03
Register HI address	**0x04**	No. of bytes	**0x04**
Register LO address	**0x24**	Register 0x03 HI	REAL/DINT* (Byte 1)
No. of HI registers	0x00	Register 0xFC LO	REAL/DINT* (Byte 0)
No of LO registers	0x02	Register 0x03+1 HI	REAL/DINT* (Byte 3)
CRC HI	0x85	Register 0xFC+1 LO	REAL/DINT* (Byte 2)
CRC LO	0x30	CRC	16 bit

All 4-bytes types: **DINT**, **DWORD**, **REAL** are always included in **two registers**. Furthermore for DINT, first register include least significant part while second register – most significant part. For example, to read current position of M1 drive, 1020 and 1021 registers should be read and then make proper conversion (if there is not proper Modbus function in MASTER driver).

2 registers conversion (4 bytes) to 32-byte type (DINT, DWORD, FLOAT).

RegisterX HI <-> Byte1
RegisterX LO <-> Byte0
RegisterX+1 HI <-> Byte3
RegisterX+1 LO <-> Byte2

32_bit_integer = **Byte3**<<24 + **Byte2**<<16 + **Byte1**<<8 + **Byte0,**
or **32_bit_integer** = Register X + Register(X + 1)<<16

4. Modbus control description

4.1 DINT/REAL registers

Part of registers for drive control is lapped and exist as **DINT** type (for entering value without comma, then values are set without unit conversion) and as REAL type (for entering floating point values in set motion units).

It is useful when is required direct position setting in pulses or master driver doesn't support register's conversion into floating point numbers **REAL** type.

Unit conversion example into pulses for DINT registers

Drive: stepper motor 200 pulses/rev. with step resolution 1/64.
Unit calculation: 200 * 64 = **12800** pulses / motor revolution.

Setting velocity 2,5 rps
To register ***VEL_ABS_INT*** (1038) write value 2,5 * 12800 = 32000 [pulses]

Read of current position:
Read register ***POS_ACT_INT*** (1020) which for example sends 57600 value.
Current drive position after calculation into motor revolution is: 57600/12800 = 4,5 [revolutions]

4.2 Program control in controller memory

Program selection from memory of the controller which have to be controlled (run, stop, holdup) is possible with ***PROG_BANK_SEL*** (1000) register. The corresponding value of selected program (0…5 value) need to be write into register.

Register ***PROGRAM_CTRL*** (1001) control program operation:

- Write 1 value causes run program
- Write 2 value causes hold current program
- Write 0 value causes stop program

If running program comes from empty program or position bank, controller will signals error.

4.3 Turning on/off and stopping drive

- ***ENABLE*** (2000) – drive on/off
- ***STOP*** (2001) – drive stop
- ***RESET*** (2002) – reset of drive position

4.4 Motion parameters setting (ramp)

Motion parameters setting which control ramp is released using registers:

- ***ACT_INT*** (1026) / ***ACT_REAL*** (1066) –acceleration in position and velocity mode
- ***DEC_INT*** (1028) / ***DEC_REAL*** (1068) – slowing down for position mode
- ***VMAX_INT*** (1030) / ***VMAX_REAL*** (1070) – maximal velocity for position mode

Motion parameters should be entered as positive values.

4.5 Drive homing

Drive homing uses ***HOME_INT*** (1036) ***/ HOME_REAL*** (1076) registers. Entered value into register defines homing velocity. Homing occur immediately after write into register. When homing is end, drive position is automatically reset.

Homing to **the limit switch** (KL) is release by write velocity as **negative value**.
Homing to **the limit switch** (KR) is release by write velocity as **positive value**.

4.6 Velocity and position setting

Absolute and relative velocity/position

Velocity and position could be set **absolutely** (e.g. drive achieve velocity/position value the same as entered register value) or **relatively** (drive will increase or decrease velocity/position by entered register value). Registers which contain **ABS** in their names set absolute values, while **REL** – relatively values.

Motion registers

- ***VEL_ABS_INT*** (1038) / ***VEL_ABS_REAL*** (1078) – set absolute velocities
- ***VEL_REL_INT*** (1040) / ***VEL_REL_REAL*** (1080) – set relative velocities
- ***POS_ABS_INT*** (1042) / ***POS_ABS_REAL*** (1082) – set absolute positions
- ***POS_REL_INT*** (1044) / ***POS_REL_REAL*** (1084) – set relative positions

4.7 Position reset, position and velocity read

Read current velocity and position

Current velocity is available in ***VEL_ACT_INT*** (1024) / ***VEL_ACT_REAL*** *(1064)* registers.
Current position in available in ***POS_ACT_INT*** *(1020)* / ***POS_ACT_REAL*** *(1060)* registers.

Current position reset

Current position can be reset by record to ***POS_RESET*** (2000) register. Current position can be also overwrite by another by record to ***POS_NEW_INT*** *(1022)* / ***POS_NEW_REAL*** *(1062)* register.

4.8 Drive status

Drive operation monitoring is possible with ***STATUS*** (1002) register. Depend on register value it takes following values:

MX_STATUS register value	Description
0	Drive turned off (EN signal inactive)
1	Drive turned on, no motion (EN signal active)
2	Drive in set velocity mode
3	Drive in motion to set position mode
4	Drive achieved the set position
5	Error of achieving set position (for operation with encoder)
6	Drive in homing mode
7	-
8	Drive in position correction mode (for operation with encoder)
9	Drive achieved limit position L while motion towards negative position value

	(by program or proximity sensor signal KL)
10	Drive achieved limit position R while motion towards positive position value (by program or proximity sensor signal KR)

Drive status could be used i.e. to define: is drive achieved set position before next position setting.

4.9 JOG control mode

The JOG mode could be used for manual drive position control with e.g. HMI touch panel. Motion of the drive occur by setting proper JOG register. Motion velocity in JOG mode is defined in ***JOG_SPEED*** (1005) register as percentage value (2..100) of max. velocity set by ***VMAX_INT*** (1030) / ***VMAX_INT*** (1070) register.

Motion for each drive is triggered by bit registers:

- ***JOG_PLUS*** (2003) - set bit: motion towards positive direction, reset bit: drive stop
- ***JOG_MINUS*** (2004) – set bit: motion towards negative direction, reset bit: drive stop

4.10 User registers

The driver contains 500 general-purpose registers (**0-499 addresses**). User can store in these registers values, which could be read or write by current controller program. Values type as INT, DINT and REAL could be write into registers. In order to write DINT and REAL value type, which are always included in two adjacent registers, they should be write into even addresses.

Register	0	1	2	3	...	498	499
	INT 0	INT 1	INT 2	INT 3		INT 498	INT 499
	DINT 0		DIN 2			DIN 498	
	REAL 0		REAL 2			REAL 498	

In order to reference to user registers in WBC language, commands which should be used are:
$I**X** – reference to register with INT type value
$D**X** – reference to register with DINT type value (X only as even value)
$R**X** – reference to register with REAL type value (X only as even value)
where X is register address 0..499

5. Documentation revision

v1.01:
- initial version