RSGD 75mm
Modbus Manual

Rev 1.0

Contents

Chapter 1 Introduction

1.1 Foreword

RSGD is a 2-phase controlled soft starter with a dedicated algorithm for general purpose applications. RSGD is equipped with Modbus RTU communication over RS485.

The purpose of this document is to outline information on the functionalities that are provided by Modbus. Modbus can be used to initialise, control and monitor RSGD general purpose soft starters. Should there be any problems that cannot be solved with the information provided in this guide, contact our technical representative who will be willing to help you.

1.2 Product inspection

Please check the following when receiving and unpacking RSGD units:

- The product is the one specified in your purchase order
- Check if there are any damages caused by transportation. In case of any problem, do not install the product and contact Carlo Gavazzi sales representative.

We suggest keeping the original packing in case it is necessary to return the instrument to our After Sales Department. In order to achieve the best results with your product, we recommend reading the instruction manual carefully. If the product is used in a way not specified by the producer, the protection provided by the product may be impaired.

1.3 Precautions

For your safety, the following symbol is to remind you to pay attention to safety instructions on configuring and installing RSGD. Be sure to follow the instructions for higher safety.

 This symbol indicates a particularly important subject or information

Please read this manual thoroughly before using the device. Should there be any problem using the product which cannot be solved with the information provided in the manual, contact your nearest Carlo Gavazzi distributor or our sales representatives to help you. Check that the device is installed in accordance with the procedures as described in this manual.

The manufacturer accepts no liability for any consequence resulting from inappropriate, negligent or incorrect installation or adjustment of the optional parameters of the equipment. The contents of this guide are believed to be correct at the time of printing. In the interests of commitment to a policy of continuous development and improvement, the manufacturer reserves the right to change the specification of the product or its performance, or the content of the guide without notice.

Chapter 2 Establishing Communication

2.1 Introduction

The RSGD can be controlled either by a PC or by a controller using Modbus RTU protocol, with one-to-one or one-to-many communication. The Modbus link between the master and slaves can be established on a 3-wire RS485 communication port.

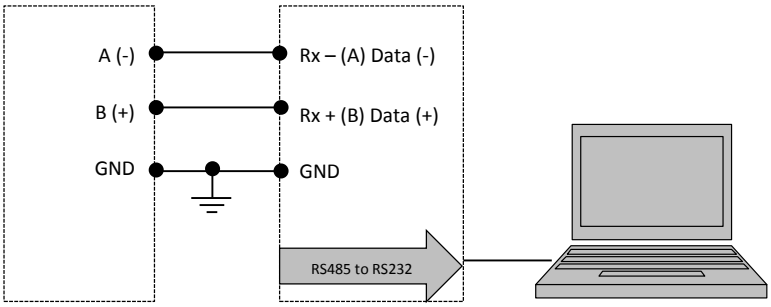
The RSGD soft starters leave the factory with default communication parameters as listed below:

Default communication parameters	
Parameter	**Default Value**
Device address	1
Baud rate	9600
Parity	No parity
Stop Bit	2

The factory default communication parameters can be modified.

2.2 Installation

In order to be able to establish communication between a PC (or a controller) and the RSGD, you will need to connect a raw cable between the communicating device and the screw terminal type terminal box available on the RSGD unit.

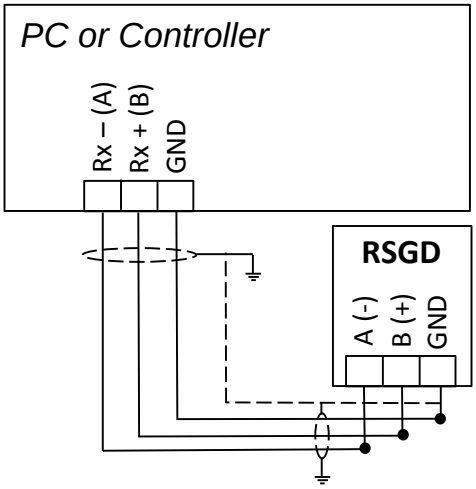


The A (-) and the B(+) connections from the soft starter need to be connected to the Rx- (A) and Rx+ (B) line of the communicating device respectively. If this connection is not followed, communication is not established.

To reduce noise on the RS485 communication raw cable, use a twisted pair and shielded cable. In addition, connect the shield to the GND terminal to further minimize the noise on the RS-485 cable.

2.3 One-to-one communication

One-to-one communication occurs between a PC (or a controller) and one RSGD.

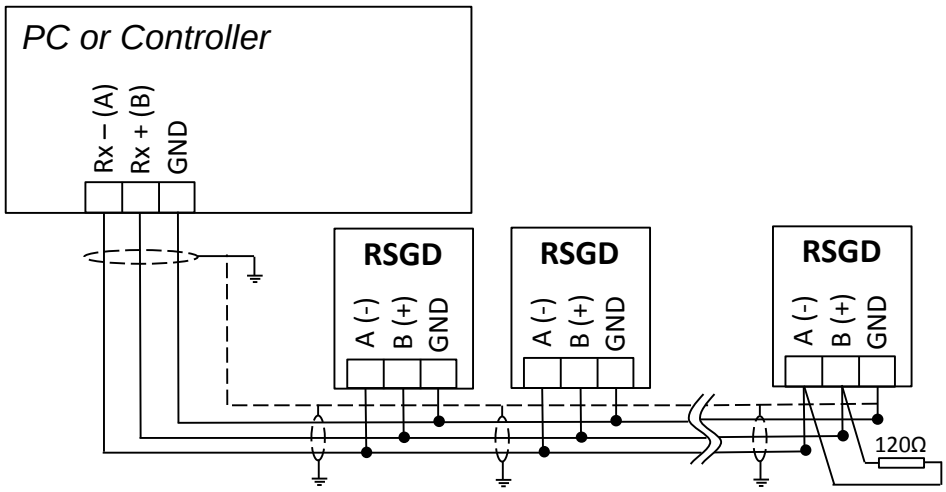


In order to establish one-to-one communication, the RSGD unit must be first powered-up with the specified supply voltage.

If the supply LED is green fixed on the soft starter, you can establish communication with the soft starter.

2.4 One-to-many communication

One-to-many communication occurs between a PC (or a controller) and multiple general purpose soft starters.



In order to establish one-to-many communication, the RSGD units must be first all powered-up with the specified supply voltage.

If the supply LED is green fixed on the soft starter, you can establish communication with the soft starter.

For one-to-many communication, the device address of each RSGD should be different.

For one-to-many communication, the baud rate and parity bit of each RSGD should be the same.

For large networks, it is required to place a 120 Ω ¼ W resistor between A (-) and the B(+) connections on the last soft starter, to avoid possible communication problems.

Chapter 3 Modbus RTU Protocol

3.1 Introduction

Modbus RTU protocol is a messaging structure used to establish master-slave communication between devices in which only one device (called master) can initiate transactions (called queries); the other devices (called slaves) respond with the requested data to the master.

3.2 Modbus RTU functions

The following Modbus functions are available on the RSGD soft starters:

- Reading of n "Input register" (code 04h)
- Writing of one "holding register" (code 06h)
- Broadcast mode (code 00h)

In this document, the Modbus address field is indicated in two modes:

- *Modicon address*: it is the 6-digit Modicon representation with Modbus function code 04h (Read input registers).
- *Physical address*: it is the word address value included in the communication frame.

Read Input Registers (04h):

This function code is used to read the contents of a 1 input register (word). The request frame specifies the starting register address and the number of registers to be read.

The register data in the response message is packed as two bytes per register (word), with the binary contents right justified within each byte. For each register, the first byte contains the high order bits (MSB) and the second contains the low order bits (LSB).

The only exceptions are:

- History file readout

Request Frame:

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	04h	-
Starting Address	2 bytes	000Bh to 00E8h	Byte order: MSB, LSB
Quantity of Registers (N word)	2 bytes	1h to 78h (1 to 120)	Byte order: MSB, LSB – As stated above no contiguous registers can be read. The values 1 to 78h are the minimum and maximum numbers respectively that are accepted. Each read function should be separately called using the number stated in the field named 'Length (words)'.
CRC	2 bytes	-	-

Response Frame (correct action):

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	04h	-
Byte Count	1 byte	N word * 2	-
Register Value	N* 2 bytes	-	Byte order: MSB, LSB
CRC	2 bytes	-	-

Response Frame (incorrect action):

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	84h	-
Exception Code	1 byte	01h, 02h, 03h, 06h	Possible exception: 01h: illegal function 02h: illegal data address 03h: illegal data value 06h: slave device busy
CRC	2 bytes	-	-

Write Single Holding Register (06h):

This function code is used to write a single holding register. The Request frame specifies the address of the register (word) to be written and its contents.

The correct response is an echo of the request, returned after the register contents have been written.

Request Frame:

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	06h	-
Starting Address	2 bytes	0000h to FFFFh	Byte order: MSB, LSB
Quantity of Registers (N word)	2 bytes	0000h to FFFFh	Byte order: MSB, LSB
CRC	2 bytes	-	-

Response Frame (correct action):

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	06h	-
Starting Address	2 bytes	0000h to 00E3h	Byte order: MSB, LSB
Register Value	2 bytes	0000h to FFFFh	Byte order: MSB, LSB
CRC	2 bytes	-	-

Response Frame (incorrect action):

Description	Length	Value	Note
Physical Address	1 byte	1h to F7h (1 to 247)	-
Function Code	1 byte	86h	-
Exception Code	2 bytes	01h, 02h, 03h, 06h	Possible exception: 01h: illegal function 02h: illegal data address 03h: illegal data value 06h: slave device busy
CRC	2 bytes	-	-

Broadcast Mode (00h)

In broadcast mode the master can send a request (command) to all the slaves. No response is returned to broadcast requests sent by the master. It is possible to send the broadcast message only with function code 06h and using address 00h.

3.3 Registers Map

Data Format Representation

Format	IEC data type	Description	Bits	Range
UINT16	UINT	Unsigned integer	16	0…65535

Group Description

Group	Description
Communication Parameters	Includes the communication parameters of the device
Device Settings	Indicates the settings of the three selector knobs found on the device
Device Status	Describes the status of the soft starter and other parameters of the device
Control	Includes several functions to control the device
Delays	Includes the delays related to stop-to-start and start-to-start intervals
Protection Settings	Indicates the default alarm limits
History File	Contains information about the last 32 starts performed. For further information on the history file refer to Appendix
Alarm Counters	Lists the number of times a particular alarm has occurred
General Counters	Includes counters related to operational use
Instantaneous Voltage and Current	Lists the instantaneous electrical variables (voltage and current)
Maximum Current Variables	Lists the maximum current measured on each phase during ramp-up, bypass and ramp-down
Instantaneous Power Variables	Lists all information related to power

Communication Parameters

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
308193	2000h	1	Device Address	UINT16	Device Address [x1] 0001h: Device Address 1 0002h: Device Address 2 . . . 00F7h: Device Address 247
308194	2001h	1	Baud Rate	UINT16	Baud Rate [x1] 0000h: 9600bps 0001h: 19200bps 0002h: 38400bps 0003h: 57600bps
308195	2002h	1	Parity	UINT16	Parity [x1] 0000h: No Parity, 2 stop bits 0001h: Odd Parity, 1 stop bit 0002h: Even Parity, 1 stop bit

Write only mode (function 06h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
408193	2000h	1	Device Address	UINT16	Range: 0001h to 00F7h [x1]
408194	2001h	1	Baud Rate	UINT16	0000h: 9600bps [x1] 0001h: 19200bps [x1] 0002h: 38400bps [x1] 0003h: 57600bps [x1]
408195	2002h	1	Parity	UINT16	0h: No Parity, 2 stop bits [x1] 1h: Odd Parity, 1 stop bit [x1] 2h: Even Parity, 1 stop bit [x1]

Device Settings

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
332769	8000h	1	Ramp-up (s)	UINT16	Ramp-up time [x1000]
332770	8001h	1	Ramp-down (s)	UINT16	Ramp-down time [x1000]
332771	8002h	1	Full load current (A_{RMS})	UINT16	Full load current [x10]
332772	8003h	1	Current Limit Ratio	UINT16	Ratio between rated soft starter current and maximum current limit [x10]
332776	8007h	1	System Voltage	UINT16	System voltage [x10]

Device Status

Read only mode (function 04h):

<table>
<tr><th>Modicon Address</th><th>Physical Address</th><th>Length (words)</th><th>Description</th><th>Data Format</th><th colspan="2">Notes [Scaling Factor]</th></tr>
<tr><td>320481</td><td>5000h</td><td>1</td><td>Soft Starter Status</td><td>UINT16</td><td colspan="2">0000h: Idle
0001h: Ramp-up
0002h: Bypass
0003h: Ramp-down
0004h: Alarm
0005h: Alarm Recovery</td></tr>
<tr><td>320482</td><td>5001h</td><td>1</td><td>Top of ramp (TOR) relay status</td><td>UNIT16</td><td colspan="2">0000h: TOR relay is OFF
0001h: TOR relay is ON</td></tr>
<tr><td>320483</td><td>5002h</td><td>1</td><td>Alarm relay status</td><td>UNIT16</td><td colspan="2">0000h: Alarm relay is OFF
0001h: Alarm relay is ON</td></tr>
<tr><td>320484</td><td>5003h</td><td>1</td><td>Run relay status</td><td>UINT16</td><td colspan="2">0000h: Run relay is OFF
0001h: Run relay is ON</td></tr>
<tr><td>320485</td><td>5004h</td><td>1</td><td>PTC status</td><td>UINT16</td><td colspan="2">0000h: PTC is open
0001h: PTC is short</td></tr>
<tr><td>320486</td><td>5005h</td><td>1</td><td>Remote Reset (RRST) status</td><td>UINT16</td><td colspan="2">0000h: RRST is open
0001h: RRST is short</td></tr>
<tr><td>320487</td><td>5006h</td><td>1</td><td>Control Input – Status A1-A2 / Modbus</td><td>UINT16</td><td colspan="2">Control Input Status [x1]
0000h: Switch OFF
0001h: Switch ON</td></tr>
<tr><td rowspan="3">320488</td><td rowspan="3">5007h</td><td rowspan="3">1</td><td rowspan="3">Alarm status</td><td rowspan="3">UINT16</td><td colspan="2">Outputs a number equal to number of flashes of alarm issued.</td></tr>
<tr><td>No of Flashes</td><td>Alarm Status</td></tr>
<tr><td>2</td><td>Wrong phase sequence</td></tr>
</table>

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]	
					No of Flashes	*Alarm Status*
					3	Line voltage out of range
					4	Phase loss (motor side)
					5	Locked rotor
					7	Over temperature
					8	Overload
					9	Supply voltage unbalance
					10	Shorted SCR
					0	Internal Fault
320489	5008h	1	Alarm reset mode	UINT16	0000h: Auto alarm reset 0001h: Manual alarm reset	

Device Status

Write only mode (function 06h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
420489	5008h	1	Alarm Reset Mode	UINT16	Set the alarm reset mode 0000h: Auto alarm reset 0001h: Manual alarm reset
420490	5009h	1	Soft Alarm Reset	UINT16	Reset the alarm: 0000h: No action 0001h: Reset alarm

Control

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
328673	7000h	1	Control Mode	UINT16	Control Mode [x1] 0000h: A1, A2 control mode 0001h: Modbus control mode
328674	7001h	1	Control Input Status - Modbus	UINT16	Control Input Status [x1] 0000h: Switch OFF 0001h: Switch ON
328675	7002h	1	Force Refresh Signal mode	UINT16	Force Refresh Signal mode [x1] 0000h: Disable 0001h: Enable
328676	7003h	1	Refresh Interval (s)	UINT16	Refresh Interval [x1]
328677	7004h	1	Force Refresh Signal (Heartbeat Signal)	UINT16	Refresh Signal [x1]

Write only mode (function 06h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
428673	7000h	1	Set the Control Mode	UINT16	0000h: A1, A2 control mode 0001h: Modbus control mode
428674	7001h	1	Start/Stop Device	UINT16	0000h: Switch OFF 0001h: Switch ON

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
428675	7002h	1	Force Refresh Signal mode	UINT16	Enable or Disable the force refresh signal 0000h: Disable 0001h: Enable
428676	7003h	1	Refresh Interval (s)	UINT16	Range: 0001h to 0258h
428677	7004h	1	Force Refresh Signal (Heartbeat Signal)	UINT16	0001h: To send force refresh signal. If force refresh signal mode is enabled, this register has to be set to 1 within every refresh interval otherwise the RSGD unit will switch OFF the output.

Delays

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
336865	9000h	1	Minimum Stop to Start Delay (s)	UINT16	Stop to Start Delay [x1]
336866	9001h	1	Minimum Start to Start Delay (s)	UINT16	Start to Start Delay [x1]
336867	9002h	1	Time from Last Stop (s)	UINT16	Time from Last Stop [x1]
336868	9003h	1	Time from Last Start (s)	UINT16	Time from Last Start [x1]

Write only mode (function 06h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
436865	9000h	1	Minimum Stop to Start Delay (s)	UINT16	Range: 0000h to FFFFh [x1]

Protection Settings

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
340961	A000h	1	Supply Voltage Unbalance Limit (%)	UINT16	Supply Voltage Unbalance Limit [x10]
340962	A001h	1	Over Voltage Supply Limit (%)	UINT16	Over Voltage Limit [x10]
340963	A002h	1	Under Voltage Supply Limit (%)	UINT16	To read Under Voltage Limit [x10]
340964	A003h	1	Load Current Unbalance Limit (%)	UINT16	Load Current Unbalance Limit [x10]
340965	A004h	1	I_{MAX} Bypass (A_{rms})	UINT16	Maximum current in bypass [x10]
340966	A005h	1	Phase Sequence alarm mode	UINT16	Phase Sequence alarm mode [x1] 0000h: Enable 0001h: Disable
340967	A006h	1	Motor Overload alarm mode	UINT16	Motor Overload alarm mode [x1] 0000h: Enable 0001h: Disable

Write only mode (function 06h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
440966	A005h	1	Phase Sequence alarm mode	UINT16	Enable or Disable the phase sequence alarm 0000h: Enable 0001h: Disable
440967	A006h	1	Motor overload alarm mode	UINT16	Enable or Disable the motor overload alarm 0000h: Enable 0001h: Disable

History File

The history file allows the user to download a series of data related to the last 32 starts done by the device.

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
349153	C000h	64	Start 1 to Start 4	UINT16	Data of the first set of 4 starts present in history [x1]
349154	C001h	64	Start 5 to Start 8	UINT16	Data of the second set of 4 starts present in history [x1]
349155	C002h	64	Start 9 to Start 12	UINT16	Data of the third set of 4 starts present in history [x1]
349156	C003h	64	Start 13 to Start 16	UINT16	Data of the fourth set of 4 starts present in history [x1]
349157	C004h	64	Start 17 to Start 20	UINT16	Data of the fifth set of 4 starts present in history [x1]
349158	C005h	64	Start 21 to Start 24	UINT16	Data of the sixth set of 4 starts present in history [x1]
349159	C006h	64	Start 25 to Start 28	UINT16	Data of the seventh set of 4 starts present in history [x1]
349160	C007h	64	Starts 29 to Start 32	UINT16	Data of the eighth set of 4 starts present in history [x1]

For further information on the history file refer to Appendix

Alarm Counters

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
324577	6000h	1	Internal fault	UINT16	Internal fault [x1]
324578	6001h	1	Shorted SCR	UINT16	Shorted SCR [x1]
324579	6002h	1	Wrong phase sequence	UINT16	Wrong phase sequence [x1]
324580	6003h	1	Line voltage out of range	UINT16	Line voltage out of range [x1]
324581	6004h	1	Phase loss (motor side)	UINT16	Phase loss (motor side) [x1]
324582	6005h	1	Locked Rotor	UINT16	Locked Rotor [x1]
324583	6006h	1	Excess ramp-up time	UINT16	Excess ramp-up time [x1]
324584	6007h	1	Over temperature	UINT16	Over temperature [x1]
324585	6008h	1	Overload	UINT16	Overload [x1]

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
324586	6009h	1	Supply Voltage Unbalance	UINT16	Supply Voltage Unbalance [x1]

General Counters

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
316385	4000h	1	kWh	UINT16	Power consumption [x10]
316386	4001h	1	Overflow of kWh counter	UINT16	Power consumption when value > 65,535 [x10]
316387	4002h	1	Number of starts	UINT16	Number of starts performed [X1]
316388	4003h	1	Overflow of number of starts counter	UINT16	Number of starts when value > 65,535 [x10]
316389	4004h	1	Running hours (hr)	UINT16	Running hours [x1]
316390	4005h	1	Running seconds (s)	UINT16	Running seconds [x1]
316391	4006h	1	Maximum start time (ms)	UINT16	Maximum start time [x1]
316392	4007h	1	Number of HP starts	UINT16	Number of HP starts performed [x1]
316393	4008h	1	Number of power up	UINT16	Number of power up [x1]
316394	4009h	1	Number of power down	UINT16	Number of power down [x1]

Instantaneous Voltage and Current

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
312289	3000h	1	V L1-L2 (V_{RMS})	UINT16	Line voltage (L1-L2) [X10]
312290	3001h	1	V L2-L3 (V_{RMS})	UINT16	Line voltage (L2-L3) [X10]
312291	3002h	1	V L1-L3 (V_{RMS})	UINT16	Line voltage (L3-L1) [X10]
312292	3003h	1	I L1 (A_{RMS})	UINT16	Line current (L1) [X10]
312293	3004h	1	I L2 (A_{RMS})	UINT16	Line current (L2) [X10]
312294	3005h	1	I L3 (A_{RMS})	UINT16	Line current (L3) [X10]

Maximum Current Variables

Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
316641	4100h	1	I L1 Ramp-up (A_{RMS})	UINT16	Line current (L1) during ramp-up [X10]
316642	4101h	1	I L2 Ramp-up (A_{RMS})	UINT16	Line current (L2) during ramp-up [X10]
316643	4102h	1	I L3 Ramp-up (A_{RMS})	UINT16	Line current (L3) during ramp-up [X10]
316644	4103h	1	I L1 Bypass (A_{RMS})	UINT16	Line current (L1) during bypass [X10]
316645	4104h	1	I L2 Bypass (A_{RMS})	UINT16	Line current (L2) during bypass [X10]

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
316646	4105h	1	I L3 Bypass (A_{RMS})	UINT16	Line current (L3) during bypass [X10]
316647	4106h	1	I L1 Ramp-down (A_{RMS})	UINT16	Line current (L1) during ramp-down [X10]
316648	4107h	1	I L1 Ramp-down (A_{RMS})	UINT16	Line current (L2) during bypass [X10]
316649	4108h	1	I L1 Ramp-down (A_{RMS})	UINT16	Line current (L3) during bypass [X10]

Instantaneous Power Variables

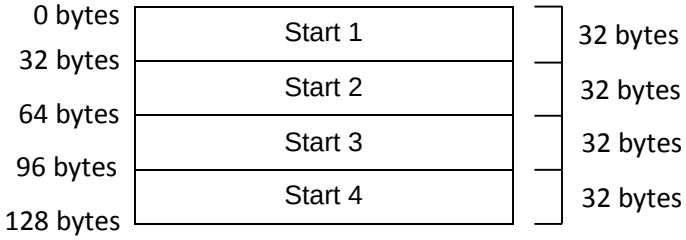
Read only mode (function 04h):

Modicon Address	Physical Address	Length (words)	Description	Data Format	Notes [Scaling Factor]
312545	3100h	1	P_{output} (kW)	UINT16	Average active power output [x10]
312546	3101h	1	Q_{output} (kVAr)	UINT16	Average reactive power output [x10]
312547	3102h	1	S_{output} (kVA)	UINT16	Average apparent power output [x10]
312548	3103h	1	PF Total	UINT16	Power factor [X1000]
312549	3104h	1	Hz	UINT16	Supply frequency [X100]
312550	3105h	1	Phase sequence	UINT16	7FFFh: -ve phase sequence 8000h: Undefined 8001h: +ve phase sequence
312551	3106h	1	Supply Voltage Unbalance (%)	UINT16	Supply Voltage Unbalance [x10]
312552	3107h	1	Load Current Unbalance (%)	UINT16	Load Current Unbalance [x10]
312553	3108h	1	TCU (%)	UINT16	Thermal Capacity Used [x10]
312554	3109h	1	NTC Temperature	UINT16	NTC Temperature [x100]

Appendix

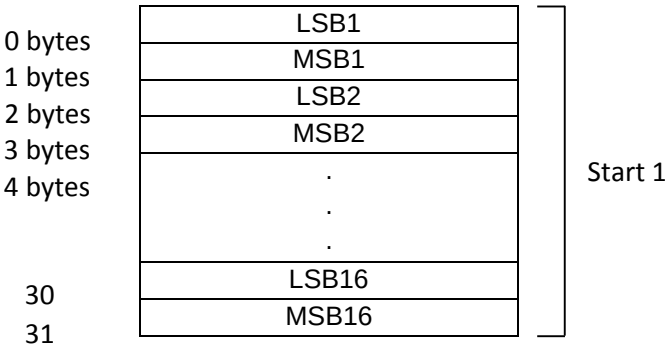
History File

When reading data from group C0h (addresses 00h to 07h), a block of **128bytes** of data is received (for every address) containing the history of 4 consecutive starts.



Block of data when reading data from C0000h

Therefore, the history of each start consists of **32bytes** (0-255 bits) in **Little Endian format** as shown below:



Data of Start 1 in Little Endian format

** All data received is expressed in hexadecimal.*

Example:

Read the data of the last 32 starts recorded in the RSGD memory.

Step 1: Read data from address C000h

Request Frame:

Description	Value	
Physical Address	01h	
Function Code	04h	
Starting Address	C0h	00h
Number of words	00h	40h
CRC	-	-

Response Frame (correct action):

Description	Value	
Physical Address	01h	
Function Code	04h	
Byte Count	80h	
Register Value	0100AF8741020F91008030080000990400000000000750C0080200848490020020 0AF8741020F91008030080000990400000000000750C0080200848490020030 0AF8741020F91008030080000990400000000000750C0080200848490020040 0AF8741020F91008030080000990400000000000750C0080200848490020h	
CRC	-	-

Response Frame (incorrect action):

Description	Value	
Physical Address	01h	
Function Code	84h	
Exception Code	01h, 02h, 03h	
CRC	-	-

Step 2: Divide the received data into four blocks of 32bytes of data

Start 1: 0100AF8741020F91008030080000990400000000000750C0080200848490020h
Start 2: 0200AF8741020F91008030080000990400000000000750C0080200848490020h
Start 3: 0300AF8741020F91008030080000990400000000000750C0080200848490020h
Start 4: 0400AF8741020F91008030080000990400000000000750C0080200848490020h

Step 3: Tag each byte of the first block as LSB1, MSB1, LSB2, …, MSB16

LSB1	01h
LSB2	AFh
LSB3	41h
LSB4	0Fh
LSB5	00h
LSB6	30h
LSB7	00h
LSB8	99h
LSB9	00h
LSB10	00h

MSB1	00h
MSB2	87h
MSB3	02h
MSB4	91h
MSB5	80h
MSB6	08h
MSB7	00h
MSB8	04h
MSB9	00h
MSB10	00h

LSB11	00h
LSB12	75h
LSB13	00h
LSB14	20h
LSB15	48h
LSB16	00h

MSB11	00h
MSB12	0Ch
MSB13	80h
MSB14	08h
MSB15	49h
MSB16	20h

** Two hexadecimal digits represent one byte.*

Step 4: By using the conversion formula convert each byte to display information of the respective start.

Code	Variable	Description
A	Start number	The start number to which the data belongs
B	System voltage & phase sequence	The system voltage and phase sequence during power-up
C	Ramp-up time	Ramp-up time setting
D	Initial firing angle	The initial firing angle at which the SCR turned on
E	Current balancing setting	Variable related to current balancing algorithm
F	Spare	Reserved
G	Alarm reset mode	Auto or manual alarm reset setting
H	Phase sequence alarm mode	Phase sequence alarm setting
I	Motor overload alarm mode	Motor overload alarm setting
J	Current Limit Setpoint	The value of current limit
K	Ramp-down time	Ramp-down time setting
L	FLC setting	Full load current setting
M	Max IL1 during ramp-up	The maximum current measured on phase 1 during ramp-up
N	Max IL2 during ramp-up	The maximum current measured on phase 2 during ramp-up
O	Max IL3 during ramp-up	The maximum current measured on phase 3 during ramp-up
P	HP mode	Indicates if start was done in HP
Q	Time to reach full speed	The time duration that the motor took to reach full speed
R	Time to reach continuous current	The time duration that the device took to reach continuous current
S	Max IL1 during bypass	The maximum current measured on phase 1 during bypass
T	Max IL2 during bypass	The maximum current measured on phase 2 during bypass
U	Max IL3 during bypass	The maximum current measured on phase 3 during bypass
V	Torque on leaving bypass	The measured torque when the soft starter entered into ramp-down mode
W	Ramp-down duration	The time duration that the soft starter took to stop the motor
X	Max IL1 during ramp-down	The maximum current measured on phase 1 during ramp-down
Y	Max IL2 during ramp-down	The maximum current measured on phase 2 during ramp-down
Z	Max IL3 during ramp-down	The maximum current measured on phase 3 during ramp-down
AA	NTC temperature (max)	The maximum internal temperature measured by the temperature sensor
AB	Junction temperature	The maximum junction temperature
AC	Estimated motor temperature	The temperature of the motor based on measurement of the current consumed
AD	Soft starter status after stopping	The soft starter status after ramp-down

** The information found in the following table is obtained from the next section.*

Code	Conversion				Display	
	Conversion Result	Multiply	Add	Result		
A	1	-	-	1	Start no. 1	
B	21	-	-	21	400V	+ve ph seq
C	7	-	-	7	30s ramp-up time setting	
D	135	-	-	135	Initial firing angle of 135°	
E	577	-	+32423	33000	33000 current balancing setting	
F	-	-	-	-	-	
G	0	-	-	0	Auto alarm reset mode	
H	0	-	-	0	Phase sequence alarm enabled	
I	0	-	-	0	Motor overload alarm enabled	
J	62	-	-	62	Current limit setpoint = 62A	
K	2	-	-	2	5s ramp-down time setting	
L	1	-	-	1	Position 1	
M	2	-	-	2	Max IL1 during ramp-up = 2A	
N	3	-	-	3	Max IL2 during ramp-up = 3A	
O	2	-	-	2	Max IL3 during ramp-up = 2A	
P	0	-	-	0	HP mode = 0	
Q	0	x0.001	-	0.000	Full speed reached at 0.000s	
R	1177	x0.001	-	1.177	Cont. current reached at 1.177s	
S	0	-	-	0	Max IL1 during bypass = 0A	
T	0	-	-	0	Max IL2 during bypass = 0A	
U	0	-	-	0	Max IL3 during bypass = 0A	
V	0	x0.025	-	0	Torque on leaving bypass = 0Nm	
W	3189	x0.001	-	3.189	Ramp-down duration = 3.189s	
X	2	-	-	2	Max IL1 during ramp down = 2A	
Y	2	-	-	2	Max IL2 during ramp down = 2A	
Z	2	-	-	2	Max IL3 during ramp down = 2A	
AA	72	-	-50	22	Tntc(max) = 22°C	
AB	73	-	-50	23	Tvj(max) = 23°C	
AC	0	-	+40	40	Tvm(max) = 40°C	
AD	32	-	-	32	Idle	

Step 5: Repeat the above procedure for another 3 times to convert the data of the first 4 starts recorded in the RSGD memory.

Step 6: Repeat steps 1 to step 5 for another 7 times to convert the data of the last 32 starts recorded in the RSGD memory.

Data Representation:

Code	A															
Format	LSB1								MSB1							
Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0 bytes							1 bytes								2 bytes

Code	***Variable***	***Conversion Statement***	***Comments***
A	Start Number	- Convert MSB1 to decimal, multiply by 256 and store the result in variable x - Convert LSB1 to decimal and store the result in variable y - Conversion result = $x + y$	N/A

Code	B					C			D							
Format	LSB2								MSB2							
Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	2 bytes							3 bytes								4 bytes

Code	***Variable***	***Conversion Statement***	***Comments***		
B	System Voltage & Phase Sequence	- Convert LSB2 to binary, right arithmetic shift by 3 and store the result in variable x - Convert x to decimal and store the result in variable y - Conversion result = y	*Value*	*System Voltage (V)*	*Phase Sequence*
			10	220	-ve ph seq
			11	220	Undefined
			12	220	+ve ph seq
			19	400	-ve ph seq
			20	400	Undefined
			21	400	+ve ph seq
			23	480	-ve ph seq
			24	480	Undefined
			25	480	+ve ph seq
			29	600	-ve ph seq
			30	600	Undefined
			31	600	+ve ph seq
C	Ramp-up time	- Convert LSB2 to binary, bitwise-AND by 0b00000111 and store the result in variable x - Convert x to decimal and store the result in variable y - Conversion result = y	*Value*	*Ramp-up time (s)*	
			1	1	
			2	2	
			3	5	
			4	10	
			5	15	
			6	20	
			7	30	
D	Initial Firing Angle	- Convert MSB2 to decimal and store the result in variable x - Conversion result = x	N/A		

Code	E								F	G	H	I	E			
Format	LSB3								MSB3							
Bits	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	4 bytes								5 bytes				6 bytes			

Code	Variable	Conversion Statement	Comments	
E	Current Balancing Setting	- Convert MSB3 to binary, bitwise-AND by 0b00001111 and store the result in variable x - Convert x to decimal, multiply by 256 and store the result in variable y - Convert LSB3 to decimal and store the result in variable z - Conversion result = $y + z$	Scaling Factor: +32423	
F	Spare	N/A	N/A	
G	Alarm Reset Mode	- Convert MSB3 to binary, bitwise-AND by 0b01000000 and store the result in variable x - Right arithmetic shift x by 6 and store the result in variable y - Convert y to decimal and store the result in variable z - Conversion result = z	*Value*	*Mode*
			0	Auto alarm reset
			1	Manual alarm reset
H	Phase Sequence Alarm Mode	- Convert MSB3 to binary, bitwise-AND by 0b00100000 and store the result in variable x - Right arithmetic shift x by 5 and store the result in variable y - Convert y to decimal and store the result in variable z - Conversion result = z	*Value*	*Mode*
			0	Enable
			1	Disable
I	Motor Overload Alarm Mode	- Convert MSB3 to binary, bitwise-AND by 0b00010000 and store the result in variable x - Right arithmetic shift x by 4 and store the result in variable y - Convert y to decimal and store the result in variable z - Conversion result = z	*Value*	*Mode*
			0	Enable
			1	Disable

Code	J								K				L			
Format	LSB4								MSB4							
Bits	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	6 bytes								7 bytes				8 bytes			

Code	Variable	Conversion Statement	Comments
J	Current Limit Setpoint	- Convert LSB4 to binary, left arithmetic shift by 2 and store the result in variable x - Convert MSB4 to binary, right arithmetic shift by 6 and store the result in variable y - Convert ($x + y$) to decimal and store the result in variable z - Conversion result = z	N/A

Code	Variable	Conversion Statement	Comments	
K	Ramp-down time	- Convert MSB4 to binary, bitwise-AND by 0b00111000 and store the result in variable x - Rhift arithmetic right x by 3 and store the result in variable y - Convert y to decimal and store the result in variable z - Conversion result = z	*Value*	*Ramp-down time (s)*
			1	0
			2	5
			3	10
			4	15
			5	20
			6	25
			7	30
L	FLC setting	- Convert MSB4 to binary, bitwise-AND by 0b00000111 and store the result in variable x - Convert x to decimal and store the result in variable y - Conversion result = y	*Value*	*FLC Setting*
			1	Position 1
			2	Position 2
			3	Position 3
			4	Position 4
			5	Position 5
			6	Position 6
			7	Position 7

Code	M								N							
Format	LSB5								MSB5							
Bits	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	8 bytes								9 bytes							10 bytes

Code	N				O										P	
Format	LSB6								MSB6							
Bits	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
	10 bytes								11 bytes							12 bytes

Code	Variable	Conversion Statement	Comments
M	Max IL1 during ramp-up	- Convert LSB5 to binary, left arithmetic shift by 2 and store the result in variable x - Convert MSB5 to binary, right arithmetic shift by 6 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
N	Max IL2 during ramp-up	- Convert MSB5 to binary, bitwise-AND by 0b00111111, left arithmetic shift by 4 and store the result in variable x - Convert LSB6 to binary, right arithmetic shift by 4 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
O	Max IL3 during ramp-up	- Convert LSB6 to binary, bitwise-AND by 0b00001111, left arithmetic shift by 6 and store the result in variable x	N/A

		- Convert MSB6 to binary, right arithmetic shift by 2 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z		
P	HP mode	- Convert MSB6 to binary, bitwise-AND by 0b00000011 and store the result in variable x - Convert x to decimal and store the result in variable y - Conversion result = y	*Value*	*Mode*
			0	HP mode 0
			1	HP mode 1
			2	HP mode 2

Code	Q															
Format	LSB7								MSB7							
Bits	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
	12 bytes							13 bytes							14 bytes	

Code	***Variable***	***Conversion Statement***	***Comments***
Q	Time to reach full speed	- Convert MSB7 to decimal, multiply by 256 and store the result in variable x - Convert LSB7 to decimal and store the result in variable y - Conversion result = x + y	Scaling Factor: x1000

Code	R															
Format	LSB8								MSB8							
Bits	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
	14 bytes							15 bytes							16 bytes	

Code	***Variable***	***Conversion Statement***	***Comments***
R	Time to reach continuous current	- Convert MSB8 to decimal, multiply by 256 and store the result in variable x - Convert LSB8 to decimal and store the result in variable y - Conversion result = x + y	Scaling Factor: x1000

Code	S								T							
Format	LSB9								MSB9							
Bits	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	16 bytes								17 bytes							18 bytes

Code	T				U										F	
Format	LSB10								MSB10							
Bits	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	18 bytes								19 bytes							20 bytes

Code	*Variable*	*Conversion Statement*	*Comments*
S	Max IL1 during bypass	- Convert LSB9 to binary, left arithmetic shift by 2 and store the result in variable x - Convert MSB9 to binary, right arithmetic shift by 6 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
T	Max IL2 during bypass	- Convert MSB9 to binary, bitwise-AND by 0b00111111, left arithmetic shift by 4 and store the result in variable x - Convert LSB10 to binary, right arithmetic shift by 4 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
U	Max IL3 during bypass	- Convert LSB10 to binary, bitwise-AND by 0b00001111, left arithmetic shift by 6 and store the result in variable x - Convert MSB10 to binary, right arithmetic shift by 2 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
F	Spare	N/A	N/A

Code	V															
Format	LSB11								MSB11							
Bits	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
	20 bytes								21 bytes							22 bytes

Code	Variable	Conversion Statement	Comments
V	Torque on leaving bypass	- Convert MSB11 to decimal, multiply by 256 and store the result in variable x - Convert LSB11 to decimal and store the result in variable y - Conversion result = $x + y$	Scaling Factor: x40

Code	W															
Format	LSB12								MSB12							
Bits	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
	22 bytes								23 bytes							24 bytes

Code	Variable	Conversion Statement	Comments
W	Ramp-down duration	- Convert MSB12 to decimal, multiply by 256 and store the result in variable x - Convert LSB12 to decimal and store the result in variable y - Conversion result = $x + y$	Scaling Factor: x1000

Code	X								Y							
Format	LSB13								MSB13							
Bits	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
	24 bytes								25 bytes							26 bytes

Code	Y				Z										F	
Format	LSB14								MSB14							
Bits	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
	26 bytes								27 bytes							28 bytes

Code	Variable	Conversion Statement	Comments
X	Max IL1 during ramp-down	- Convert LSB13 to binary, left arithmetic shift by 2 and store the result in variable x - Convert MSB13 to binary, right arithmetic shift by 6 and store the result in variable y	N/A

		- Convert (x + y) to decimal and store the result in variable z - Conversion result = z	
Y	Max IL2 during ramp-down	- Convert MSB13 to binary, bitwise-AND by 0b00111111, left arithmetic shift by 4 and store the result in variable x - Convert LSB14 to binary, right arithmetic shift by 4 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A
Z	Max IL3 during ramp-down	- Convert LSB14 to binary, bitwise-AND by 0b00001111, left arithmetic shift by 6 and store the result in variable x - Convert MSB14 to binary, right arithmetic shift by 2 and store the result in variable y - Convert (x + y) to decimal and store the result in variable z - Conversion result = z	N/A

Code	AA								AB							
Format	LSB15								MSB15							
Bits	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
	28 bytes								29 bytes							30 bytes

Code	***Variable***	***Conversion Statement***	***Comments***
AA	NTC Temperature (max)	- Convert LSB15 to decimal and store the result in variable x - Conversion result = x	Scaling Factor: +50
AB	Junction Temperature	- Convert MSB15 to decimal and store the result in variable x - Conversion result = x	Scaling Factor: +50

Code	AC								AD							
Format	LSB16								MSB16							
Bits	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
	30 bytes								31 bytes							32 bytes

Code	***Variable***	***Conversion Statement***	***Comments***
AC	Estimated Motor Temperature	- Convert LSB16 to decimal and store the result in variable x - Conversion result = x	Scaling Factor: -40

Code	Variable	Conversion Statement	Comments	
AD	Soft starter Status after stopping	- Convert MSB16 to decimal and store the result in variable x - Conversion result = x	*Value*	*State*
			0	Internal fault
			1	Spare
			2	Spare
			3	Short circuit during idle
			4	Short circuit during ramp
			5	Spare
			6	Negative phase sequence
			7	Spare
			8	Spare
			9	Synchronisation loss
			10	System voltage not detected
			11	Line voltage out of range
			12	Current unbalance
			13	Spare
			14	Spare
			15	Locked rotor
			16	Spare
			17	Spare
			18	Excess ramp-up time
			19	Spare
			20	Spare
			21	Internal over temperature
			22	Spare
			23	Spare
			24	Motor overload
			25	Maximum current in bypass
			Value	*State*
			26	PTC alarm
			27	Supply voltage unbalance
			28	Spare
			29	Spare
			30	Low internal voltage
			31	No reset
			32	Idle