

Modbus Software Specification

Status actual running SST firmware, release 12-01-16

Company: intecma
File : MODBUS RTU EN Version.doc
Date of Origin: 15-04-14
Revision: 05-02-16
Author: EJW / PP

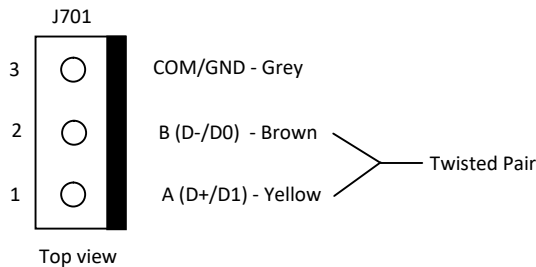
	SST330i-M(-ES)	SST130i-xxx-M(-ES)	SST360i-M
<u>SW</u>	4.03 4.04 (not yet released)	2.04 (12-01-16)	4.04
<u>HW</u>	6.1	4.4	-6.1 (not yet released)

MODBUS RTU Hardware

- RS485 transceiver
- By Default no termination, Pup and Pdown resistors are placed.
- The following parts can be mounted on the board on request:
 - Termination resistor between D+/D1 and D-/D0 RT = 150R .05W (R705)
 - R Pull-up on D+/D1 Line Rpu = 560-680R 0.25W (R703)
 - R Pull down on D-/D0 Line Rpd = 560-680R 0.25W (R704)

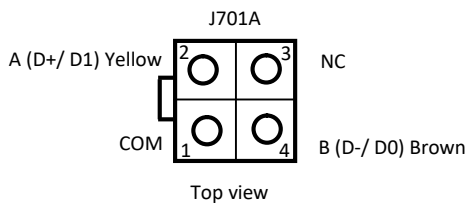
Connector on the Control Board.

- 3-pole vert. pinheader, pitch 2.54mm with friction lock, wire part with crimp contacts, AWG26-24 Molex KK-6471 - 22-01-2035 series and / or second sources



Connector on the Power board:

- 4-pole vert. pinheader (2x2), pitch 4.2mm Mini Fit (molex 39-28 8040, Würth 649 004 211 22)



MODBUS RTU Software and protocol:

A Modbus RTU frame looks as follows;

address	Function Code	Data	CRC control
1 Byte	1 Byte MSb = 0	N Byte(s)	2 bytes

Every request of the master will be send back "filled-in" when the Slave accept this request. If an error occurs, than the most significant bit of the Function Code will be set and the data will be replaced by the exception code. The exception Code will be send by the slave.

Not all Exception Codes of each Function Code will be supported.

Error frame

address	Function Code	Exeption Code	CRC control
1 Byte	1 Byte MSb = 1	N Byte(s)	2 bytes

Description	Exception Code
Function code not suported	0x01
IllegalDataAdress	0x02
IllegalDataValue	0x03
SlaveDeviceFailure	0x04

Each byte is placed in a separate subframe. This subframe exists of 11 bits, named 1 startbit, 8 databits and 2 stopbits.

Between these subframes a maximal silence period of 1.5x the subframe length may occur, otherwise the frame is said to be incomplete.

After the complete Function Code has been send, a silence period of 3.5 times the duration of the subframe must be counted untill the slave is allowed to send a response.

The MSB must be send first (big endian).

The duration of the subframes is also called the inter-character Time-out.

The duration of the frames is called the inter-frame delay.

The address of each device in a MODBUS network has a scope from 0 to 247 (247 included).

The address 0 is reserved for the master.

The Function Codes (range 1-255) used, are to be constructed according the MODBUS standard.

The size of the datafield depends on the used Function Code.

The CRC check is to be executed with a polynome of 0xA001.

Defined registers and coils

The MODBUS registers (that are also the separate registers and /or references to the internal addressing, so that only the required registers can be written or read) are to be configured as follows:

registers slave	0	1	2	3	4	5	6	7	type	
master read/write bit 0-2 (read and write allowed) (coils)										
(1)start/(0)stop motor	■								coil 1	ONLY allowed when starting is released (Coil5)
1 goto configure mode (and motor will be stopped, if not already) 0 exit configure mode		■							coil 2	ONLY write to zero When Master is sure configuration is complete (all registerd are written)
Reset errors			■						coil 3	When set Reset errors (use with care) Bit is cleared afterwards
Reset the entire starter				■					Coil 4	Reset the whole starter
read-only (coils)										
configuration complete					■				coil 5	1=configuration done
Phase sequence						■			coil 6	0=Left 1=right
Usb is controller							■		Coil 7	When 1 Modbus write function is replied as Illegal function code
Physical start/stop is controller								■	Coil 8	When 1 Modbus write function is replied as Illegal function code
error	■	■	■	■	■	■	■	■	coil 9-16	Softstarter errors *
error2	■	■	■	■	■	■	■	■	coil 17-25	17 -> phase loss 18 ->capacitor failure (130) 19 ->overvoltage error (130) 330 SW ≥ 4.04 and 130 SW ≥ 2.02; 20 ->No Successful start (could not start the motor with current settings) 21 -> Shearpin-Overcurrent 22-25 To be defined
Ready to receive a start command	■								26	1 = the ICL has detected mains supply and a power source. and initialized on mains input and parameters

read-only (registers)		register number	unit	SST330i-M (-ES)	SST130i-75-M (-ES)	SST130i-140-M (-ES)
Starterversion	low	0		330->0x4A 360->0x68	0x82 full Type name can be found at the registers 20-29	
Starterversion	high	0		330 & 360 -> 0x01	0x00	
HWversion Softstarter(prt0)	low	1		=0x06 at HW <u>6.0</u>	0x03 at HW <u>3.0</u>	
HWversion Softstarter(prt1)	high	1		=0x00 at HW <u>6.0</u> =0x01 at HW <u>6.1</u>	0x00 at HW <u>3.0</u> 0x01 at HW <u>3.1</u>	
SWversion Softstarter(prt0)	low	2		=0x04 at SW 4.00	0x02 at SW 2.00	
SWversion Softstarter(prt1)	high	2		=0x02 at SW 4. <u>21</u>	0x02 at SW 2. <u>21</u>	
SWversion Softstarter(prt2)	low	3		=0x01 at SW 4. <u>21</u>	0x01 at SW 2. <u>01</u>	
Serial number	Low +high	8-11		Most significant Bytes in register 8 (Uint_32)		
ImRMS	low	12	Ampere	0x05 at 5 Ampère	0x05 at 5 Ampere Not total current*****	
Heatsinktemperature	low	13	° C	0x1D at 29 °C 0xFF at -1 °C 0xFE at -2 °C Two's complement		
Alpha	low	14	%	0-100		
AutoResetTimer (decrementing over time)	low	15	seconds	Timer itself		
AutoResetTimer (decrementing over time)	low	16	minutes	Timer itself		
supplyVoltage (Usupply)	low	17	Volt	0x1A at 26V		
Mains voltage	Low+high	18	Volt			
motorstatus **	low	19		See **		
startcapacitor Voltage	low+high	20	Volt	not used	Voltage Startcapacitor	
not used	-	21				
Full Type description (16 char string)	Low+high	22-29		SW version 4.04 and up "SST330i-M" OR "SST330i-M-ES" when "ES" version	"SST130i-75-M" OR "SST130i-75-M-ES" when "ES" version	"SST130i-140-M" OR "SST130i-140-M-ES" when "ES" version

* |errorstate|wrong frequency|Phase-sequence(not for 130)|overcurrent|overtemp|undervoltage|no Motor|Under Voltage Supply |Phase loss|

registers for configuration (write enabled when in configuration mode) when trying to write when not in configuration mode, error will be generated and motor will be stopped				when the master try to write a value which is not allowed, a error frame with exceprion code 03 will be returned		
IMaxLow	low	30	Ampere	5-47	75	5-140
IMaxHigh	low	31	Ampere	5-47	75	5-140
IMaxNom	low	32	Ampere	5-30		
IMaxLowTimer	low	33	*0,1 sec	1-100	1-6	
IMaxHighTimer	low	34	*0,1 sec	0-100	0-6	
IMaxNomTimer	low	35	Seconds	1-10		
StartVoltageConfiguration	low	36	% of line	20-55		
Acceleration****	low	37	% of max acceleration	Register see ****		
PhaseDirectionAllowed	low	38		0=Left 1=right 2=both	not used	
decelerationTime	low	39	* 0,1 sec	0-200	0-10	
AutoResettimer	Low	40	minutes	Register 0x01-0x61 0x61 will be timer off		
Modbus Time-out time (sec)(after a communication time-out, it stops the motor) (USE FOR COMMUNICATION FAILURE)	Low	41	seconds	0-255 when 0 no time-out will be set. *****		
Modbus Time-out time (sec) (time-out to fully reset the starter) (USE FOR COMMUNICATION FAILURE)	Low	42	seconds	0-255 when 0 no time-out will be set. *****		
version and Type configured (read+write)						
Master Type number configured	low +high	4+5	user defined	Register (type of Heatpump) (to be defined by costumer)		
Master Type configuration number	low +high	6	user defined	Register (version number parameter set configured by heatpump) (to be defined by costumer)		
Read-only						
Register configuration version *****	low	7		Register = 0x00 1.00 0x01 @SW4.03	Register = 0x01 1.01	
Register configuration version revision	high	7		Register = 0x01 1.00	Register = 0x01 1.01	

**motorstatus	motor stopped	0
	motor starting	1
	motor at speed	2
	motor stopping	3

**** Acceleration	25%	1
	50%	2
	75%	3
	100%	4

The Acceleration is the count of periods $100/ACC$ where after the phase cut is made greater. The possible values are 25,50,75,100% in software. In firmware the sinus these values are defined as 1,2,3,4 in Modbus we will define it as 1(25%),2(50%),3(75%),4(100%) . When the ACC is 25%, each 4th sinewave period gets a new phase angle control value. When the ACC value is 100%, each 20ms Sinewave period get a new phase angle control value. Actually the phase angle step-size is controlled with it. The higher the value, the faster the phase angle goes to it final value as long as the max. current level is not exceeded.

***** when a modbus Command is received the time-out timer will be reset. A time-out will occur after the time in register 41 or 42 (in seconds) is reached without receiving a MODBUS command. After this (coil 1) will be cleared (motor will be stopped) when the time of register 41 is reached. When the time of register 42 is reached, (coil 1) will be cleared (motor will be stopped), and the starter reboots, this is for use like a watchdog timer.

***** this is the register specification revision which represents the position and available registers.

******* the currents which is read and limited on, at the 1 phase softstarter (which is the stator current) will be higher than the total current (stator and rotor current). This is due to reactive power which cancel each other partially.**

The registers 0 - 42 (42 included) must be readable and the registers 1 and 2 must be writeable. Registers 30 to 42 (42 included) are only to be written if the configuration mode is active.

If the Master tries to write to this register if the configuration mode is not switched-on, than the motor will be stopped and an Exception Code will be send as a response on this request to the master.

Further it is important that a counter have to be implemented in which is stored what subframe has been received. Also a flag is needed that is set when the total frame has been received.

After that The CRC check can be started, a response be generated and sended.

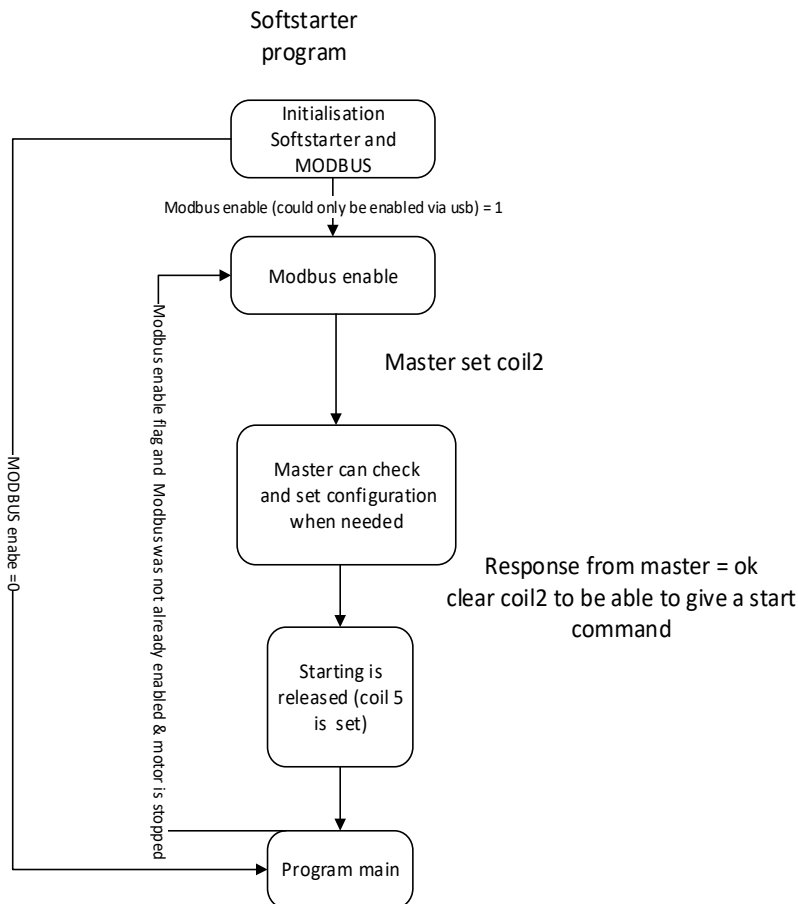
Before sending the response, the data must be evaluated firdt in the CRC check routine.

Also a counter is needed for counting the subframes. Also here this must be controlled by flags .

When the motor is stopped once by USB, when returning to Modbus control mode de motor will stop til a start commando is sent after returning to modbus mode or after a USB start command which is not followed by a stop command.

Initialisation sequence

Before the starter can be used , an initialisation must be done. Before initialisation the starter give the exception response when the master set the coil to start. The status of the starter can be seen at coil 5. To get the starter in initialisation mode, coil2 has to be set. After this the master can check the settings. When the settings are not correct the master can change it. When they are correct, don't set them again, because every time they are set , these settings will be written to flash after leaving the configuration mode. The next step will be to exit the configuration mode and give a start command when needed.



Function Codes

The read Function Codes that are supported;

FC1 Read Coil Status

parameters (data field) M -> S

- **Data-address first coil (= coilnumber-1)** **2 Bytes**
- **Number of coils to read** **2 Bytes**

parameters (data field) S -> M

- **Number of data Bytes that follows = N (quantity coils/8 (round to next higher integer))** **1 Byte**
- **Bytes with needed data** **N Byte (s)**

FC4 Read Input Registers

parameters (data field) M -> S

- Data-address first address 2 Bytes
- Number of coils to read 2 Bytes

parameters (data field) S -> M

- Number of data Bytes that follows = N (quantity coils/8 (round to next higher integer))
1 Byte
- Bytes with needed data N Byte (s)

The supporting write operations are;

FC05 Force Single Coil

parameters (data field) M -> S

- Data-address coil (= coilnumber-1) 2 Bytes
- status to be written (FF00 = ON, 0000 = OFF) 2 Bytes

parameters (data field) S -> M

- Data-address coil (= coilnumber-1) 2 Bytes
- status to be written (FF00 = ON, 0000 = OFF) 2 Bytes

FC16 Preset Multiple registers

parameters (data field) M -> S

- Data-address first register 2 Bytes
- Quantity of Bytes per register to write = X 2 Bytes
- quantity of data-Bytes which follow = N 1 Byte
- Data-Bytes X*N Byte(s)

parameters (datafield) S -> M

- Data-address first register 2 Bytes
- Number of data-Bytes that are written 2 Bytes

FC6 Preset Single register

parameters (data field) M -> S

- Data-address of the register 2 Bytes
- Data-Bytes 2 Bytes

parameters (datafield) S -> M

- Data-address of the register 2 Bytes
- Data-Bytes 2 Bytes

The possible Exception Codes which are supported in the first version are: 01, 02, 03, 06

NOTES

When configuration settings are set when not in configuration mode a IllegalDataAdress exception is set

When Configuration exceed the limit, an IllegalDataValue exception is generated.

To get in configure mode coil 2 has to be set

To exit configure mode coil 2 has to be cleared and changed settings will be flashed. coil 5 is automatically set when the data is flashed. When no parameters are changed, the step to go to configure mode en go out of it, has to be done, so the softstarter "know" the master noticed a new starter when it was replaced. After this coil 5 is made 1 so the master know if the start command isn't blocked. When writing coil 1 (start motor) when the starting command is not released (coil 5) an IllegalDataAdress will be returned.

When a message is received corrupt or incomplete by the slave no reaction is responded

When a type of error occurred, coil 9 is set. This coil has to be read by the master regularly

When the starter hasn't received anything from the master within Time-out time (register41) in seconds a Modbus time-out will occur. When the content of this register is 0, a time-out will never occur. This register will standard be preprogrammed at the value requested.

When a time-out is occurred a Modbus time-out message will be generated. Because of this the motor will be stopped.

When usb is the controller coil 7 is set and every Modbus write is replied as Illegal function code exception.

When parallel input is the controller coil 8 is set and every Modbus write is replied as Illegal function code exception.

When a register variable uses only the low Byte, the low Byte will only be written. An IllegalDataValue exception will only be written if the low Byte isn't in specified range. When writing a register which uses the low and high Byte and there is a specified range (which isn't the case with the current register-set) an IllegalDataValue exception will be generated when this range is exceeded.

The Modbus specification defines big endian order (The MSB must be send first).

At this point we run out of ram so we could receive and send a maximum of 54 Bytes. When more than 54 Bytes (including CRC, slave-id and function-code) will be received by the slave the message will be ignored and no response is send. When the number of Bytes which is needed to send by the slave is larger than 54 Bytes this message will not be generated and a SlaveDeviceFailure exception is send instead.

the currents which is read and limited on, at the 1 phase softstarter (which is the stator current) will be higher than the total current (stator and rotor current). This is due to reactive power which cancel each other partially.

The 1 phase softstarter will start the motor only when the startcapacitor is out of charge

Errata and Modifications

- 10-03-14 IMotor at register 12 deleted and register 13-14 changed to 12-13
- 19-03-14 coil 3-4 moved to 5-6
- 20-03-14 The phrase "Because running MODBUS is not the only task of the starter program, it is important that a flag must be set when the EUART is ready. " deleted because this is something about the modbus control inside the microcontroller
- 21-03-14 added; "When the motor is stopped once by usb, when returning to Modbus control mode de motor will stop til a start command is sent after returnig to modbus mode or after an usb start command which is not followed by a stop command. "
- 31-03-14 register 5 - and up numbering changed and added register 6 (Master Type configuration number), 4 registers (7 to 11) for future use. polynome in text changed to 0xA001 extended register 4 to 2 registers
- 3-04-14 Changed the startup routine, added coils clear errors, USB is controller, Parallel input is controller,extra to defined later 8 error bits. added registers decrementTime (x 0,1 sec) ,AutoResettimer (writable), Time-out time (sec), added description Acceleration .
- 04-04-14 Added register 0 Starterversion
- 15-04-14 Added ranges, examples about registersand a note about count of Bytes
- 06-05-14 Added 16 bit register 18 for future use (mains voltage)
- 28-08-14 Changed Acceleration table, because it doesn't match the text
- 28-08-14 Added register 42
- 28-08-14 Added coil 4
- 08-09-14 Added coil 16
- 26- 9-14 added coil 17
- 6-3-15 added SST130i-075-M (-ES), SST130i-140-M (-ES) and register 20-29.

SW version changes

SST330i-M (-ES)

- 4.04 added register 22-29 to make the full starter type available over Modbus
- 4.04 changed Register configuration version to 1.01

SST130i-075-M (-ES) SST130i-140-M (-ES) (in adition to SST330i-M-ES SW 4.02)

- 2.00 added register 20, 22-29 to make the full starter type and capacitor voltage available over Modbus
- 2.00 changed Register configuration version to 1.01
- 2.00 added coil 18, to generate an error when the start capacitor fails
- 2.04 improved performance Modbus response rate. 12-01-16