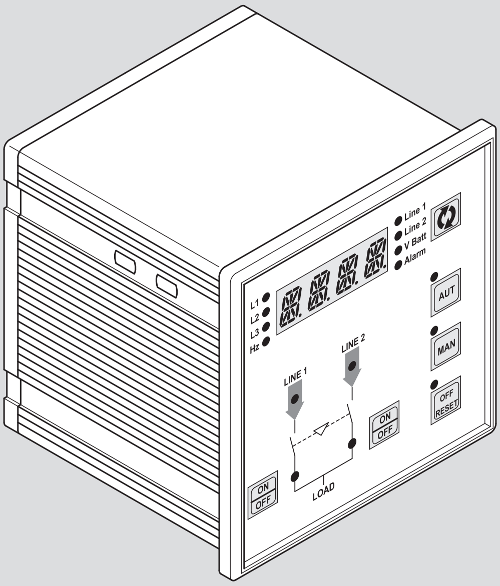
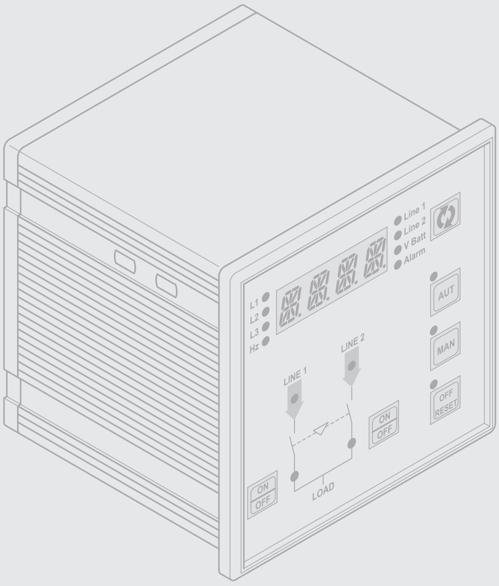
Automatic transfer switch
4 226 80

Modbus communication Protocol



Automatic transfer switch
4 226 80



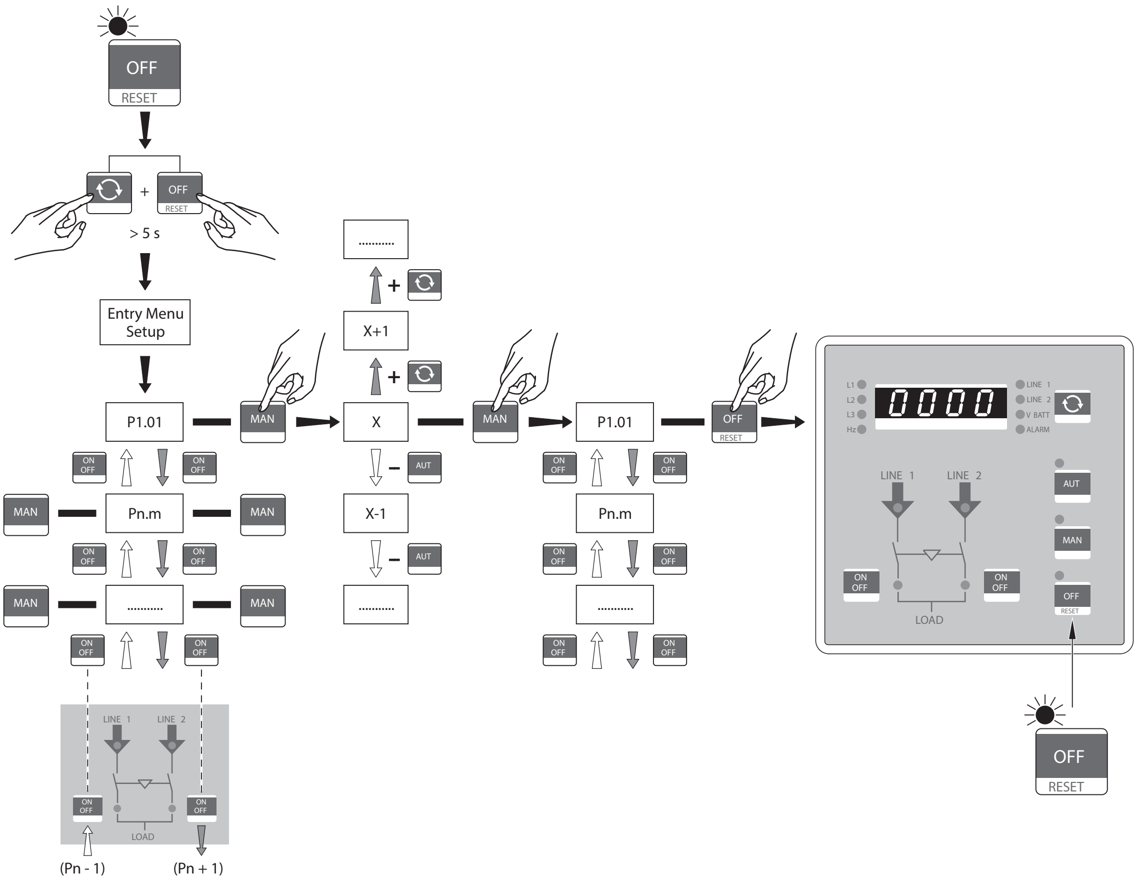
Index

1. Modbus Protocol

The 4 226 80 automatic transfer switch controller support the communication protocols Modbus RTU and Modbus ASCII on RS232 onboard serial port, connected with 4 226 96 cable (contact Legrand for more info).
Using this function it is possible to read the device status and to control the unit through supervision software or through other master devices supporting Modbus, like PLCs.

2. Parameters Setting

To configure the Modbus protocol, enter P7 menu – Serial communication. To access menus and to navigate through them, see the following procedure:



Menu P7 – Serial Communication

PAR	FUNCTION	RANGE	DEFAULT
P7.01	RS-232 Address	1 ..245	1
P7.02	RS-232 Baud Rate	2400 4800 9600 19200 38400	9600 baud
P7.03	RS-232 Protocol	RTU ASC – ASCII Mod – ASCII + modem	RTU
P7.04	RS-232 Parity	Non – None Odd EvE - Even	NON

3. Modbus RTU Protocol

For Modbus RTU protocol, the communication message has the following structure:

T1T2T3	Address (8 bit)	Function (8 bit)	Data (N x 8 bit)	CRC (16 bit	T1T2T3

- The Address field contains the serial address of the slave destination device.
- The Function field contains the code of the function that must be executed by the slave.
- The Data field contains data sent to the slave or data received from the slave in response to a query (maximum length for data field is 60 16-bit registers, so 120 bytes).
- The CRC field allows the master and slave devices to check the message integrity.
  If a message has been corrupted by electrical noise or interference, the CRC field allows the devices to recognize the error and thereby to ignore the message.
- The T1 T2 T3 sequence corresponds to a time in which data must not be exchanged on the communication bus to allow the connected devices to recognize the end of one message and the beginning of another. This time must be at least 3.5 times the time required to send one character.

The ATS measures the time that elapses from the reception of one character and the following. If this time exceeds the time necessary to send 3.5 characters at the selected baud rate, then the next character will be considered as the first of a new message.

4. Modbus Functions

The available functions are:

03 = Read Multiple Holding Registers	It allows to read the ATS internal registers
04 = Read input register	It allows to read the ATS measures
06 = Preset single register	It allows writing parameters
07 = Read exception	It allows to read the device status
10 = Preset multiple register	It allows writing several parameters
17 = Report slave ID	It allows to read information about the device

Example:
To read the value of the battery voltage, which resides at location 30 ($1E_{hex}$) from the ATS with serial address 01 the message to send is the following:

01	04	00	1D	00	02	E1	CD

Where:

01= slave address
04 = Modbus function 'Read input register'
00 1D = Address of the required register (battery voltage) decreased by one
00 02 = Number of registers to be read beginning from address 30
E1 CD = CRC Checksum

The ATS answer will be the following:

01	04	04	00	00	00	7C	FA	65

Where:

01 = ATS address (Slave 01)
04 = Function requested by the master
04 = Number of bytes sent by the ATS
00 00 00 7C = Hex value of battery voltage = 124 = 12.4V
FA 65 = CRC checksum

4.1 Function 04: Read Input Register

The Modbus function 04 allows to read one or more consecutive registers from the slave memory. The address of each measure is given in table 7.1. As for Modbus standard, the address in the query message must be decreased by one from the effective address reported in the table.
If the measure address is not included in the table or the number of requested registers exceeds the acceptable max number, the ATS will return an error code (see error table in chap. 5).

Example:

MASTER QUERY	
Slave address	08_{hex}
Function	04_{hex}
MSB address	00_{hex}
LSB address	$0F_{hex}$
MSB register number	00_{hex}
LSB register number	08_{hex}
MSB CRC	21_{hex}
LSB CRC	57_{hex}

In the above example, slave 08 is requested for 8 consecutive registers beginning with address 10_{hex}. Thus, registers from 10_{hex} to 17_{hex} will be returned. As usual, the message ends with the CRC checksum.

SLAVE RESPONSE	
Slave address	08_{hex}
Function	04_{hex}
Byte number	10_{hex}
MSB register 10_{hex}	00_{hex}
LSB register 10_{hex}	00_{hex}
--	----
MSB register 17_{hex}	00_{hex}
LSB register 17_{hex}	00_{hex}
MSB CRC	$5E_{hex}$
LSB CRC	83_{hex}

The answer is always composed of the slave address, the function code requested by the master and the contents of the requested registers. The answer ends with the CRC.

4.2 Function 06: Preset Single Register

This function allows to write in the registers. It can be used only with registers with address higher than 1000_{hex}. For instance, it is possible to change setup parameters. If the value is not in the correct range, the ATS will answer with an error message. In the same way, if the parameter address is not recognised, the ATS will send an error response.
The address and the valid range for each parameter are indicated in Table 7.3.

Example:

MASTER MESSAGE	
Slave address	08_{hex}
Function	06_{hex}
MSB address	16_{hex}
LSB address	03_{hex}
MSB register number	00_{hex}
LSB register number	$1E_{hex}$
MSB CRC	FD_{hex}
LSB CRC	13_{hex}

Slave response:
The slave response is an echo to the query, that is the slave sends back to the master the address and the new value of the variable.

legrand

4. Modbus Functions

4.3 Function 07: Read Exception Status

This function allows to read the status of the automatic transfer switch.

Example:

MASTER QUERY	
Slave address	08_{hex}
Function	07_{hex}
MSB CRC	47_{hex}
LSB CRC	$B2_{hex}$

The following table gives the meaning of the status byte sent by the ATS as answer:

BIT	MEANING
0	Operative mode OFF / Reset
1	Operative mode MAN
2	Operative mode AUT
3	(not used)
4	On error
5	(not used)
6	DC power supply ok
7	Global alarm on

4.4 Function 16: Preset Multiple Register

This function allows to modify multiple parameters with a single message, or to preset a value longer than one register.

Example:

MASTER MESSAGE	
Slave address	08_{hex}
Function	10_{hex}
MSB register address	20_{hex}
LSB register address	01_{hex}
MSB register number	00_{hex}
LSB register number	02_{hex}
MSB data	01_{hex}
LSB data	$F4_{hex}$
MSB data	06_{hex}
LSB data	83_{hex}
MSB CRC	55_{hex}
LSB CRC	$3A_{hex}$

SLAVE RESPONSE	
Slave address	08_{hex}
Function	10_{hex}
MSB register address	20_{hex}
LSB register address	01_{hex}
MSB byte number	00_{hex}
LSB byte number	04_{hex}
MSB CRC	$9B_{hex}$
LSB CRC	53_{hex}

4.5 Function 17: Report Slave Id

This function allows to identify the device type.

Example:

MASTER QUERY	
Slave address	08_{hex}
Function	11_{hex}
MSB CRC	$C6_{hex}$
LSB CRC	$7C_{hex}$

SLAVE RESPONSE	
Slave address	08_{hex}
Function	11_{hex}
Bytes counter	04_{hex}
Data 01 (Type) ➊	$5F_{hex}$
Data 02 (SW release)	04_{hex}
Data 03 (HW release)	00_{hex}
Data 04 (Parameters release)	01_{hex}
MSB CRC	$..._{hex}$
LSB CRC	$..._{hex}$

➊ 95 – $5F_{hex}$ = 4 226 80

5. Errors

In case the slave receives an incorrect message, it answers with a message composed by the queried OR-ed function with 80_{hex}, followed by an error code byte.
In the following table the error codes sent by the slave to the master:

CODE	ERROR
01	Invalid function
02	Invalid address
03	Parameter out of range
04	Function execution impossible
06	Slave busy, function momentarily not available

6. Modbus ASCII Protocol

The Modbus ASCII protocol is normally used in applications that require to communicate through a couple of modems. The functions and addresses available are the same as for the RTU version, but the transmitted characters are in ASCII and the message end is delimited by Carriage Return CR and Line Feed LF instead of a transmission pause.
If parameter P7.04 is set as Modbus ASCII protocol, the communication message on the correspondent communication port has the following structure:

:	ADress (2 chars)	Function (2 chars)	Data (N chars)	LRC (2 chars)	CR LF

- The Address field contains the serial address of the slave destination device.
- The Function field contains the code of the function that must be executed by the slave.
- The Data field contains data sent to the slave or data received from the slave in response to a query. The maximum allowable length is of 80 consecutive registers.
- The LRC field allows the master and slave devices to check the message integrity. If a message has been corrupted by electrical noise or interference, the LRC field allows the devices to recognize the error and thereby ignore the message.
- The message ends always with CRLF control character (0D 0A).

Example:
To read the value of the phase-to-phase voltage, which resides at location 04 (04_{hex}) from the slave with serial address 08, the message to send is the following:

:	08	04	00	03	00	02	EF	CRLF

Where:

: = ASCII $3A_{hex}$ message start delimiter
08 = slave address
04 = Modbus function “Read input register”
00 03 = Address of the required register (L2 voltage of line 1) decreased by one
00 02 = Number of registers to be read beginning from address 04
EF = LRC Checksum
CRLF = ASCII $0D_{hex}$ $0A_{hex}$ = Message end delimiter

The answer is the following:

:	08	04	04	00	00	01	A0	4F	CRLF

Where:

: = ASCII $3A_{hex}$ message start delimiter
08 = address (Slave 08)
04 = Function requested by the master
04 = Number of bytes sent by the device
00 00 01 A0 = Hex value of the phase-to-phase voltage (= 416V)
4F = LRC checksum
CRLF = ASCII $0D_{hex}$ $0A_{hex}$ = Message end delimiter

7. Tables

7.1 Measures Supplied by Serial Communication Protocol

To be used with functions 03 and 04

ADDRESS	WORDS	MEASURE	UNIT	FORMAT
02_{hex}	2	Voltage of line 1 L1-N	V	Unsigned long
04_{hex}	2	Voltage of line 1 L2-N	V	Unsigned long
06_{hex}	2	Voltage of line 1 L3-N	V	Unsigned long
08_{hex}	2	Voltage of line 1 L1-L2	V	Unsigned long
$0A_{hex}$	2	Voltage of line 1 L2-L3	V	Unsigned long
$0C_{hex}$	2	Voltage of line 1 L3-L1	V	Unsigned long
$0E_{hex}$	2	Voltage of line 2 L1-N	V	Unsigned long
10_{hex}	2	Voltage of line 2 L2-N	V	Unsigned long
12_{hex}	2	Voltage of line 2 L3-N	V	Unsigned long
14_{hex}	2	Voltage of line 2 L1-L2	V	Unsigned long
16_{hex}	2	Voltage of line 2 L2-L3	V	Unsigned long
18_{hex}	2	Voltage of line 2 L3-L1	V	Unsigned long
$1A_{hex}$	2	Frequency of line 1	Hz/10	Unsigned long
$1C_{hex}$	2	Frequency of line 2	Hz/10	Unsigned long
$1E_{hex}$	2	Battery voltage (DC power supply)	VDC / 10	Unsigned long
20_{hex}	2	(not used)	-	-
22_{hex}	2	(not used)	-	-
24_{hex}	2	(not used)	-	-
26_{hex}	2	(not used)	-	-
28_{hex}	2	(not used)	-	-
$2A_{hex}$	2	(not used)	-	-
$2C_{hex}$	2	(not used)	-	-
$2E_{hex}$	2	(not used)	-	-
30_{hex}	2	(not used)	-	-
32_{hex}	2	Number of operations of line 1 breaker in AUT	nr	Unsigned long
34_{hex}	2	Number of operations of line 2 breaker in AUT	nr	Unsigned long
36_{hex}	2	Number of operations of line 1 breaker in MAN	nr	Unsigned long
38_{hex}	2	Number of operations of line 2 breaker in MAN	nr	Unsigned long
$3A_{hex}$	2	Number of switching alarms of breaker 1	nr	Unsigned long
$3C_{hex}$	2	Number of switching alarms of breaker 2	nr	Unsigned long
$3E_{hex}$	2	(not used)	-	-
40_{hex}	2	Alarms (1)	bits	Unsigned long

7. Tables

(1) Reading the words starting at address 40_{hex} will return 32 bits with the following meaning:

BIT	CODE	ALARM
0	A01	Battery voltage too low
1	A02	Battery voltage too high
2	A03	Line 1 circuit breaker timeout
3	A04	Line 2 circuit breaker timeout
4	A05	Line 1 wrong phase sequence
5	A06	Line 2 wrong phase sequence
6	A07	Timeout load not powered
7	A08	Generator not available
8	A09	Emergency
9	A10	Breaker 1 Trip
10	A11	Breaker 2 Trip
11..31	--	(not used)

7.2 Status Bits

To be used with functions 03 and 04

ADDRESS	WORDS	FUNCTION	FORMAT
2070_{hex}	1	Front panel keyboard status ❶	Unsigned integer
2071_{hex}	1	Digital inputs status (by pin) ❷	Unsigned integer
2072_{hex}	1	Digital outputs status (by pin) ❸	Unsigned integer
-	-	-	-
2074_{hex}	1	Line 1 voltage status ❹	Unsigned integer
2075_{hex}	1	Line 1 breaker status ❺	Unsigned integer
2076_{hex}	1	Line 2 voltage status ❹	Unsigned integer
2077_{hex}	1	Line 2 breaker status ❺	Unsigned integer
2078_{hex}	2	Input function status ❻	Unsigned integer
$207A_{hex}$	1	Output function status ❼	Unsigned integer
$207B_{hex}$	1	Display messages status ❽	Unsigned integer

❶ Following table shows meaning of bits of the word at address 2070_{hex}:

BIT	KEY
0	ON-OFF Line 2
1	OFF mode
2	MAN mode
3	AUT mode
4	(not used)
5	ON-OFF Line 1
6	Line 2 measure selection
7	Line 1 measure selection
8…15	(not used)

❷ Following table shows meaning of bits of the word at address 2071_{hex}:

BIT	INPUT
0	Input terminal 2.1 status
1	Input terminal 2.2 status
2	Input terminal 2.3 status
3	Input terminal 2.4 status
4	Input terminal 2.5 status
5	Input terminal 2.6 status
6…15	(not used)

❸ Following table shows meaning of bits of the word at address 2072_{hex}:

BIT	OUTPUT
0	Output terminal 4.1 status
1	Output terminal 4.3 status
2	Output terminal 5.1 status
3	Output terminal 5.3 status
4	Output terminal 3.1 status
5	Output terminal 3.4 status
6…15	(not used)

7. Tables

❹ Following table shows meaning of bits of the word at address 2074_{hex} (Line 1) and 2076_{hex} (Line 2):

BIT	LINE STATUS
0	Line values into limits
1	Line values into limits delayed
2	Voltage into limits
3	Voltage ok
4	Frequency into limits
5	Frequency ok
6	Voltage below min
7	Voltage above max
8	Voltage asymmetry
9	Voltage phase loss
10	Frequency below min
11	Frequency above max
12	Wrong phase sequence
13	All line parameters ok
14-15	(not used)

❺ Following table shows meaning of bits of the word at address 2075_{hex} (Line 1) and 2077_{hex} (Line 2):

BIT	BREAKER STATUS
0	Breaker closed
1	Trip alarm
2	(not used)
3	Command status (1 = close)
4	Close command output
5	Open command output
6...15	(not used)

❻ Following table shows meaning of bits of the word at address 2078_{hex}:

BIT	INPUT FUNCTIONS STATUS
0	Line 1 breaker closed feedback
1	Line 1 breaker trip
2	(not used)
3	Line 2 breaker closed feedback
4	Line 2 breaker trip
5	(not used)
6	Transfer to secondary line
7	Inhibit return to main line
8	Emergency pushbutton
9	Generator start
10	Generator 1 ready
11	(not used)
12	Keyboard locked
13	Lock parameters
14	(not used)
15	(not used)

❼ Following table shows meaning of bits of the word at address $207A_{hex}$:

BIT	OUTPUT FUNCTIONS STATUS
0	Line 1 breaker open
1	Line 1 breaker close
2	Line 2 breaker open
3	Line 2 breaker close
4	Global alarm
5	(not used)
6	Generator 2 start
7	ATS ready
8…15	(not used)

❽ Following table shows meaning of bits of the word at address $207B_{hex}$:

BIT	DISPLAY MESSAGE STATUS
0	(not used)
1	Generator 2 start
2	(not used)
3	Generator 2 cooling
4	Load transfer 2 → 1
5	Load transfer 1 → 2

7. Tables

7.3 Commands

To be used with function 06

ADDRESS	WORDS	STATUS	FORMAT
2F00h	1	Operative mode change ❶	Unsigned integer
2F01h	1	Device reset (warm boot) ❷	Unsigned integer
2F02h	1	Setup parameters back to factory default ❷	Unsigned integer
2F03h	1	Save parameters in memory ❸	Unsigned integer
2F04h	1	Hour counter reset❹	Unsigned integer
2F05h	1	Operation counters reset❹	Unsigned integer
2F06h	1	(not used)	
2F07h	1	(not used)	
2F08h	1	Keyboard lock ON/OFF ❺	Unsigned integer
...	...	...	...
2F0Dh	1	Front panel keystroke simulation ❻	Unsigned integer

❶ The following table shows the values to be written to address $2F00_{hex}$ to achieve the correspondent functions.

VALUE	FUNCTION
0	Switch to OFF mode
1	Switch to MAN mode
2	Switch to AUT mode

❷ Writing value 01_{hex} to the indicated address, the correspondent function will be executed.

❸ Writing value AA_{hex} to the indicated address, the correspondent function will be executed.

❹ Writing value FF_{hex} to the indicated address, the correspondent function will be executed.

❺ The following table shows the values to be written to address $2F08_{hex}$ to achieve the correspondent functions.

VALUE	FUNCTION
0	Keyboard unlock
1	Keyboard lock

❻ The following table shows the bit positions of the value to be written to address $2F0D_{hex}$ to achieve the correspondent functions.

BIT	KEYSTROKE SIMULATION
0	Line 2 manual switching
1	OFF mode
2	MAN mode
3	AUT mode
4	(not used)
5	Line 1 manual switching
6	Line 2 measure selection
7	Line 1 measure selection
8...15	(not used)

7.4 Parameter Settings

Using the Modbus protocol it is possible to access the menu parameters. In the following tables it is reported the numerical range for each parameter. To correctly understand the correspondence between the numeric value and the selected function and/or the unit of measure, please see the ATS operating manual.

To make effective the changes made to setup parameters it is necessary to store the values in memory, using the dedicated command described in table 7.3.

7.4.1 Setup Parameters

To be used with functions 04 and 06.

ADDRESS	WORDS	PARAMETER	RANGE	FORMAT
3000_{hex}	1	P1.01 Nominal voltage	100…690	Unsigned integer
3001_{hex}	1	P1.02 VT ratio	100…999	Unsigned integer
3002_{hex}	1	P1.03 Wiring configuration	0…3 ❶	Unsigned integer
3003_{hex}	1	P1.04 Voltage control mode	0…2 ❶	Unsigned integer
3004_{hex}	1	P1.05 Nominal frequency	0…1 ❶	Unsigned integer
3005_{hex}	1	P1.06 Nominal battery voltage	0…3 ❶	Unsigned integer
3006_{hex}	1	P1.07 Language	0…5 ❶	Unsigned integer
3100_{hex}	1	P2.01 Application type	0…2 ❶	Unsigned integer
3101_{hex}	1	P2.02 Phase sequence control	0…2 ❶	Unsigned integer
3102_{hex}	1	P2.03 Priority line selection	0…1 ❶	Unsigned integer
3103_{hex}	1	P2.04 Line1->Line2 interlock time	1…900	Unsigned integer
3104_{hex}	1	P2.05 Line2->Line1 interlock time	1…900	Unsigned integer
3105_{hex}	1	P2.06 Transfer strategy	0…1 ❶	Unsigned integer
3106_{hex}	1	P2.07 Switch command mode	0…2 ❶	Unsigned integer
3107_{hex}	1	P2.08 Switch open/close timeout	1…900	Unsigned integer
3108_{hex}	1	P2.09 Switch open pulse duration	1…600	Unsigned integer
3109_{hex}	1	P2.10 Switch close pulse duration	1…600	Unsigned integer
$310A_{hex}$	1	P2.11 Load not powered timeout	0…3600 ❷	Unsigned integer
$310B_{hex}$	1	P2.12 Inhibit auto retransfer	0…1 ❶	Unsigned integer
$310C_{hex}$	1	P2.13 Pre-transfer time	0…300 ❷	Unsigned integer
$310D_{hex}$	1	P2.14 Post-transfer time	0…300 ❷	Unsigned integer
$310E_{hex}$	1	P2.15 Generator start delay	0…900	Unsigned integer
$310F_{hex}$	1	P2.16 Generator cooling time	1…3600	Unsigned integer
3110_{hex}	1	P2.17 Generator rotation time	0…14 ❶	Unsigned integer
3111_{hex}	1	P2.18 Generator rotation hour	0…23	Unsigned integer
3112_{hex}	1	P2.19 Generator rotation minutes	0…59	Unsigned integer
3113_{hex}	1	P2.20 Min battery voltage	69…100 ❷	Unsigned integer

❶ The association between the numerical value and the function has to be done in a sequential way, considering the function listed on the operative manual. The first function is obtained by setting 0, while the last function by setting the maximum value allowed by the range.

❷ To select OFF, set the minimum numerical value allowed by range.

❸ To select OFF, set the maximum numerical value allowed by range.

❹ The association between the numerical value and the function has to be done in a sequential way, considering the function listed on the operative manual. The first function is obtained by setting the maximum value allowed by the range, while the last function by setting value 0.

7. Tables

ADDRESS	WORDS	PARAMETER	RANGE	FORMAT
3200_{hex}	1	P3.01 Min voltage drop out	70… 98	Unsigned integer
3201_{hex}	1	P3.02 Min voltage pick up	75… 100	Unsigned integer
3202_{hex}	1	P3.03 Min voltage delay	1 … 9000	Unsigned integer
3203_{hex}	1	P3.04 Max voltage drop out	102…121 ❸	Unsigned integer
3204_{hex}	1	P3.05 Max voltage pick up	100…115	Unsigned integer
3205_{hex}	1	P3.06 Max voltage delay	1…9000	Unsigned integer
3206_{hex}	1	P3.07 Phase loss threshold	59…85 ❷	Unsigned integer
3207_{hex}	1	P3.08 Phase loss delay	1…300	Unsigned integer
3208_{hex}	1	P3.09 Asymmetry threshold	2…21 ❸	Unsigned integer
3209_{hex}	1	P3.10 Asymmetry delay	1…9000	Unsigned integer
$320A_{hex}$	1	P3.11 Min frequency	79…100 ❷	Unsigned integer
$320B_{hex}$	1	P3.12 Min frequency delay	1 …9000	Unsigned integer
$320C_{hex}$	1	P3.13 Max frequency	100…121 ❸	Unsigned integer
$320D_{hex}$	1	P3.14 Max frequency delay	1…9000	Unsigned integer
$320E_{hex}$	1	P3.15 Line 1 ok delay (when line 2 not OK)	1…3600	Unsigned integer
$320F_{hex}$	1	P3.16 Line 1 ok delay (when line 2 OK)	1…3600	Unsigned integer
3300_{hex}	1	P4.01 Min voltage drop out	70… 98	Unsigned integer
3301_{hex}	1	P4.02 Min voltage pick up	75… 100	Unsigned integer
3302_{hex}	1	P4.03 Min voltage delay	1 … 9000	Unsigned integer
3303_{hex}	1	P4.04 Max voltage drop out	102…121 ❸	Unsigned integer
3304_{hex}	1	P4.05 Max voltage pick up	100…115	Unsigned integer
3305_{hex}	1	P4.06 Max voltage delay	1…9000	Unsigned integer
3306_{hex}	1	P4.07 Phase loss threshold	59…85 ❷	Unsigned integer
3307_{hex}	1	P4.08 Phase loss delay	1…300	Unsigned integer
3308_{hex}	1	P4.09 Asymmetry threshold	2…21 ❸	Unsigned integer
3309_{hex}	1	P4.10 Asymmetry delay	1…9000	Unsigned integer
$330A_{hex}$	1	P4.11 Min frequency	79…100 ❷	Unsigned integer
$330B_{hex}$	1	P4.12 Min frequency delay	1 …9000	Unsigned integer
$330C_{hex}$	1	P4.13 Max frequency	100…121 ❸	Unsigned integer
$330D_{hex}$	1	P4.14 Max frequency delay	1…9000	Unsigned integer
$330E_{hex}$	1	P4.15 Line 1 ok delay (when line 1 not OK)	1…3600	Unsigned integer
$330F_{hex}$	1	P4.16 Line 1 ok delay (when line 1 OK)	1…3600	Unsigned integer
3400_{hex}	1	P5.01.01 Programmable Input 1	0…13❶	Unsigned integer
3401_{hex}	1	P5.01.02 Programmable Input 1	0…1 ❶	Unsigned integer
3402_{hex}	1	P5.01.03 Programmable Input 1	0…250 ❶	Unsigned integer
3403_{hex}	1	P5.01.04 Programmable Input 1	0…250 ❶	Unsigned integer
3404_{hex}	1	P5.02.01 Programmable Input 2	0…13❶	Unsigned integer
3405_{hex}	1	P5.02.02 Programmable Input 2	0…1 ❶	Unsigned integer
3406_{hex}	1	P5.02.03 Programmable Input 2	0…250 ❶	Unsigned integer
3407_{hex}	1	P5.02.04 Programmable Input 2	0…250 ❶	Unsigned integer
3408_{hex}	1	P5.03.01 Programmable Input 3	0…13❶	Unsigned integer

3409 hex	1	P5.03.02 Programmable Input 3	0…1 ❶	Unsigned integer
340A hex	1	P5.03.03 Programmable Input 3	0…250 ❶	Unsigned integer
340B hex	1	P5.03.04 Programmable Input 3	0…250 ❶	Unsigned integer
340C hex	1	P5.04.01 Programmable Input 4	0…13❶	Unsigned integer
340D hex	1	P5.04.02 Programmable Input 4	0…1 ❶	Unsigned integer
340E hex	1	P5.04.03 Programmable Input 4	0…250 ❶	Unsigned integer
340F hex	1	P5.04.04 Programmable Input 4	0…250 ❶	Unsigned integer
3410 hex	1	P5.05.01 Programmable Input 5	0…13❶	Unsigned integer
3411 hex	1	P5.05.02 Programmable Input 5	0…1 ❶	Unsigned integer
3412 hex	1	P5.05.03 Programmable Input 5	0…250 ❶	Unsigned integer
3413 hex	1	P5.05.04 Programmable Input 5	0…250 ❶	Unsigned integer
3414 hex	1	P5.06.01 Programmable Input 6	0…13❶	Unsigned integer
3415 hex	1	P5.06.02 Programmable Input 6	0…1 ❶	Unsigned integer
3416 hex	1	P5.06.03 Programmable Input 6	0…250 ❶	Unsigned integer
3417 hex	1	P5.06.04 Programmable Input 6	0…250 ❶	Unsigned integer
3500 hex	1	P6.01.01 Programmable Output 1	0…11 ❶	Unsigned integer
3501 hex	1	P6.01.02 Programmable Output 1	0…1 ❶	Unsigned integer
3502 hex	1	P6.02.01 Programmable Output 2	0…11 ❶	Unsigned integer
3503 hex	1	P6.02.02 Programmable Output 2	0…1 ❶	Unsigned integer
3504 hex	1	P6.03.01 Programmable Output 3	0…11 ❶	Unsigned integer
3505 hex	1	P6.03.02 Programmable Output 3	0…1 ❶	Unsigned integer
3506 hex	1	P6.04.01 Programmable Output 4	0…11 ❶	Unsigned integer
3507 hex	1	P6.04.02 Programmable Output 4	0…1 ❶	Unsigned integer
3508 hex	1	P6.05.01 Programmable Output 5	0…11 ❶	Unsigned integer
3509 hex	1	P6.05.02 Programmable Output 5	0…1 ❶	Unsigned integer
350A hex	1	P6.06.01 Programmable Output 6	0…11 ❶	Unsigned integer
350B hex	1	P6.06.02 Programmable Output 6	❶	Unsigned integer

❶ The association between the numerical value and the function has to be done in a sequential way, considering the function listed on the operative manual. The first function is obtained by setting 0, while the last function by setting the maximum value allowed by the range.

❷ To select OFF, set the minimum numerical value allowed by range.

❸ To select OFF, set the maximum numerical value allowed by range.

ADDRESS	WORDS	PARAMETER	RANGE	FORMAT
3600 hex	1	P7.01 RS232 serial address	1…245	Unsigned integer
3601 hex	1	P7.02 RS232 serial speed	0…4 ❶	Unsigned integer
3602 hex	1	P7.03 RS232 protocol	0…4 ❶	Unsigned integer
3603 hex	1	P7.04 RS232 parity	0…4 ❶	Unsigned integer

❶ The association between the numerical value and the function has to be done in a sequential way, considering the function listed on the operative manual. The first function is obtained by setting 0, while the last function by setting the maximum value allowed by the range.

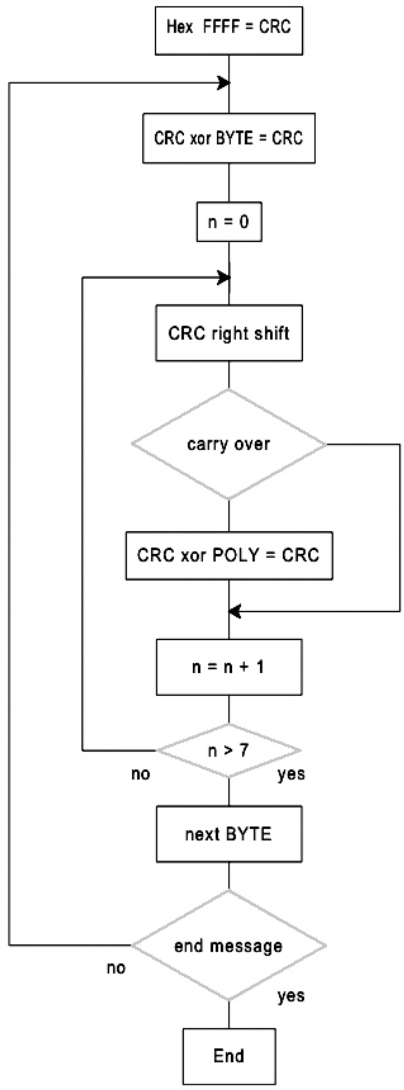
7. Tables

A. CRC Calculation (Checksum for RTU)

CRC calculation algorithm

Example: Frame = 0207_{hex}

CRC initialization	**1111**	**1111**	**1111**	**1111**	
Load the first byte			0000	0010	
Execute xor with the first Byte of the frame	1111	1111	1111	1101	
Execute 1st right shift	0111	1111	1111	1110	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1101	1111	1111	1111	
Execute 2nd right shift	0110	1111	1111	1111	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1100	1111	1111	1110	
Execute 3rd right shift	0110	0111	1111	1111	**0**
Execute 4th right shift	0011	0011	1111	1111	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1001	0011	1111	1110	
Execute 5th right shift	0100	1001	1111	1111	**0**
Execute 6th right shift	0010	0100	1111	1111	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1000	0100	1111	1110	
Execute 7th right shift	0100	0010	0111	1111	**0**
Execute 8th right shift	0010	0001	0011	1111	**1**
Carry=1, load polynomial	1010	0000	0000	0001	
Load the second byte of the frame			0000	0111	
Execute xor with the second byte of the frame	1000	0001	0011	1001	
Execute 1st right shift	0100	0000	1001	1100	**1**
Carry=1, load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1110	0000	1001	1101	
Execute 2nd right shift	0111	0000	0100	1110	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1101	0000	0100	1111	
Execute 3rd right shift	0110	1000	0010	0111	**1**
Carry=1,load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1100	1000	0010	0110	
Execute 4th right shift	0110	0100	0001	0011	**0**
Execute 5th right shift	0010	0100	0000	1001	**1**
Carry=1, load polynomial	1010	0000	0000	0001	
Execute xor with the polynomial	1001	0010	0000	1000	
Execute 6th right shift	0100	1001	0000	0100	**0**
Execute 7th right shift	0010	0100	1000	0010	**0**
Execute 8th right shift	0001	0010	0100	0001	**0**
CRC Result	**0001**	**0010**	**0100**	**0001**	
	12_{hex}		**41_{hex}**		



Note: The byte 41_{hex} is sent first (even if it is the LSB), then 12_{hex} is sent.

B. LRC Calculation (Checksum for ASCII)

Example:

Address	01		00000001
Function	04		00000100
Start address high	00		00000000
Start address low	00		00000000
Number of registers	08		00001000
		Sum	00001101
	1. complement		11110010
		+ 1	00000001
	2. complement		11110101
LRC result			**$F5_{hex}$**

LEGRAND
Pro and Consumer Service
BP 30076 - 87002
LIMOGES CEDEX FRANCE
www.legrand.com

Installer stamp