

EnerSolis

ES 7200HC

Modbus/JBUS

Protocol

Introduction.....	3
Introduction JBUS Function	4
PV Protocol for J-BUS.....	6
GENERAL TABLE DATA AREA DEFINITION	7
JBUS Table	8
1. Alarms	8
GENERAL VECTOR INDEX.....	8
Alarms Data Sequence	8
Alarms Data Area.....	8
2. Errors.....	10
GENERAL VECTOR INDEX.....	10
Errors Data Sequence.....	10
Errors Data Area	10
3. Measurements	12
GENERAL VECTOR INDEX.....	12
Measurements Data Area	12

Introduction

This document describes the Helios Inverter protocol, adopted to communicate with all communication products, like Supervisor, Network communication, etc...

This protocol will be implemented in the PV equipment, in order to use the same driver for all products.

COMMUNICATION LAYERS

APPLICATIONS
<i>PV MONITOR</i>
DATA TABLE
<i>FIXED</i>
ADDRESS SPECIFICATION
<i>JBUS P</i>
JBUS TRANSPORT PROTOCOL
HARDWARE
<i>RS232 / RS485 / USB / TCP/IP</i>

GENERAL MESSAGE FORMAT

SLAVE NUMBER (1 byte)	Specified the destination node
FUNCTION CODE (1 byte)	Specified a READ or WRITE data command
DATA FIELD	Information to read or write data (Address, value, number of data...)
CONTROL WORD (CRC16) (2 bytes, 1 word)	Algorithm calculation of each data

JBUS FUNCTION

READ WORD: **code function 3**

Introduction JBUS Function

FUNCTION 0x3

Ex. Request to slave number1, the data (10 words) beginning at 0xC000 (Address)

Request

Slave Number	Function READ	Address High	Address Low	0	Nb of word to read	CRC Low	CRC High
1	0x03	0xC0	0x00	0	10		

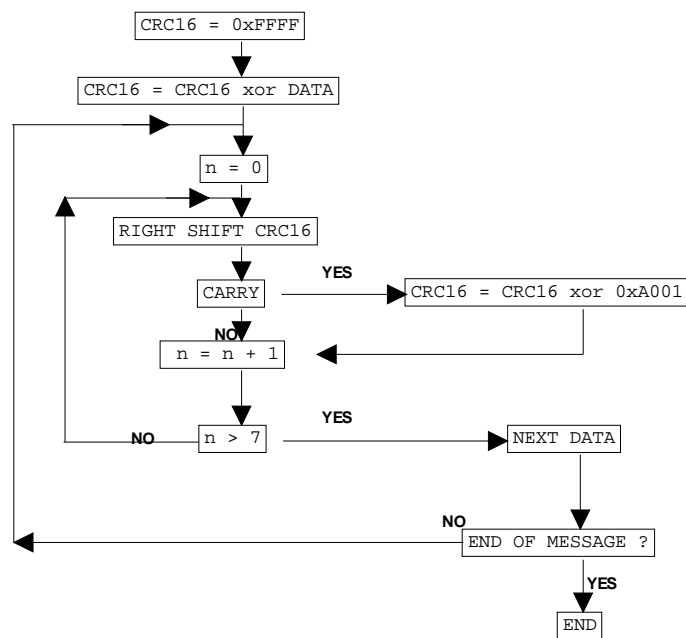
Slave message

Response

Slave Number	Function READ	Nb of byte	First data hi byte	First data low byte	Next data	CRC Low	CRC High
1	0x03	20	0x20	0x02		

Example: the first data is $(0x20 * 0x100) + 0x02 = 0x2002$

CRC 16 CALCULATION



Example of CRC calculation

```
unsigned int CALCUL_CRC(unsigned int *Msg, unsigned int lenght)
{
    unsigned int Crc;
    int i,n;
    Crc = 0xFFFF;
    for ( i = 1 ; i <= lenght ; i++ )
    {
        Crc ^=  Msg[i];
        for ( n = 1 ; n <= 8 ; n++)
        {
            /* if CRC is even */
            if ((Crc % 2) == 0)
                /* to right decrement */
                Crc >>= 1;
            else
            {
                Crc >>= 1;
                Crc ^= 0xA001;
            }
        }
    }
    return( Crc );
}
```

PV Protocol for J-BUS

DATA BASE	INFORMATION CODING
Alarms	ALxx for Alarms
Errors	Erxx for Errors
Measurements	Mxx for Measurements

GENERAL TABLE DATA AREA DEFINITION

DATA	Length in word	TYPE	Information	Jbus Function	Start Address	End Address
Alarms	2	bit	32 Alarms	3 (r)	0xC000	0xC001
Error	3	bit	48 Error	3 (r)	0xC010	0xC012
Measurements	96	word	96 Measurements	3 (r)	0xC020	0xC07F

r : read

JBUS Table

1. Alarms

Ex. Request to slave number 1(Alarms)

Request

Slave Number	Function READ	Address High	Address Low	0	Nb of word to read	CRC Low	CRC High
1	0x03	0xC0	0x00	0	0x03		

Response

Slave Number	Function READ	Nb of byte	First data high byte	First data low byte	Next data	CRC Low	CRC High
1	0x03	0x06	AL15~AL08	AL07~AL00		

GENERAL VECTOR INDEX

Address High	Address Low	End Address	DATA AREA	LENGTH (IN WORDS)
0xC0	0x00	0xC002	Alarms	3

Alarms Data Sequence

Word 0		Word 1	
High	Low	High	Low
AL15.....AL08	AL07.....AL00	AL31.....AL24	AL23.....AL16
Word 2			
High	Low		
AL47.....AL40	AL39.....AL32		

Alarms Data Area

CODE Type(bit)	Description	Necessary
AL00	Utility Voltage Over Rang	
AL01	Utility Voltage Under Rang	
AL02	Utility Frequency Over Rang	
AL03	Utility Frequency Under Rang	
AL04	Boost:1-Input Voltage Over Rang	

AL05	Boost:1-Input Voltage Under Rang	
AL06~AL07	Reserve	
AL08	Anti-islanding general alarm	
AL09	Reserve	
AL10	Ground current fault general alarm	
AL11	Ground impedance fault general alarm	
AL12	Reserve	
AL13	Utility Phase Fault	
AL14~AL20	Reserve	
AL21	Calculate Fail	
AL22~AL24	Reserve	
AL25	Over Temperature Derating	
AL26~AL30	Reserve	
AL31	DC Varistor Fault	
AL32	AC Varistor Fault	
AL33~AL47	Reserve	

2. Errors

Ex. Request to slave number 1(Errors)

Request

Slave Number	Function READ	Address High	Address Low	0	Nb of word to read	CRC Low	CRC High
1	0x03	0xC0	0x10	0	0x03		

Response

Slave Number	Function READ	Nb of byte	First data high byte	First data low byte	Next data	CRC Low	CRC High
1	0x03	0x06	Er15~Er08	Er07~Er00		

GENERAL VECTOR INDEX

Address High	Address Low	End Address	DATA AREA	LENGTH (IN WORDS)
0xC0	0x10	0xC012	Errors	3

Errors Data Sequence

Word 0		Word 1	
High	Low	High	Low
Er15.....Er08	Er07.....Er00	Er31.....Er24	Er23.....Er16
Word 2			
High	Low		
Er47.....Er40	Er39.....Er32		

Errors Data Area

CODE Type(bit)	Description	Necessary
Er00	DC BUS Charge Fault	
Er01	Reserve	
Er02	Slave CPU Fault	
Er03~Er05	Reserve	
Er06	EPO (Emergency Power Off Mode)	
Er07	DC BUS Voltage Over-Rang	
Er08	Reserve	
Er09	Inverter output current Over-Rang	

Er10	Reserve	
Er11	Inverter output power Over-Rang	
Er12	Charger Fault	
Er13	Inverter output Short-Circuit	
Er14	PLL(Phase-Locked Loop) Fault	
Er15~Er16	Reserve	
Er17	EEPROM Data Error ,Use Default Value	
Er18	Heatsink temperature Over-Rang	
Er19~Er21	Reserve	
Er22	Inverter Relay Fault	
Er23	Reserve	
Er24	Inverter Current sense Fault	
Er25	Booster _1 - Input current Over-Rang	
Er26~Er28	Reserve	
Er29	Inverter Output Current Balance Over-Rang	
Er30~Er36	Reserve	
Er37	Fan out of order	
Er38~Er42	Reserve	
Er43	Output Balance current Sense Fault	
Er44~Er47	Reserve	

3. Measurements

Ex. Request to slave number 1(Measurements)

Request

Slave Number	Function READ	Address High	Address Low	0	Nb of word to read	CRC Low	CRC High
1	0x03	0xC0	0x20	0	10		

Response

Slave Number	Function READ	Nb of byte	First data high byte	First data low byte	Next data	CRC Low	CRC High
1	0x03	20	0x20	0x02		

Example: the first data is $(0x20 * 0x100) + 0x02 = 0x2002$

GENERAL VECTOR INDEX

Address High	Address Low	End Address	DATA AREA	LENGTH (IN WORDS)
0xC0	0x20	0xC07F	Measurements	96

Measurements Data Area

ADDRESS INDEX	CODE	Description	Unit	Necessar y
0xC020	M00	Output power	KW*100(1)	✓
0xC021	M01	AC voltage phase L1	V	✓
0xC022	M02	Reserve		
0xC023	M03	Reserve		
0xC024	M04	AC output current L1	A*10(2)	✓
0xC025	M05	Reserve		
0xC026	M06	AC frequency (L1)	Hz*10(2)	✓
0xC027	M07	DC-Bus Positive-voltage	V	✓
0xC028	M08	DC-Bus Negative-voltage	V	
0xC029	M09	Inverter internal temperature	°C	✓
0xC02A	M10	Inverter Heat sink temperature	°C	✓
0xC02B	M11	DC1 input voltage	V	✓
0xC02C	M12	Reserve	V	✓
0xC02D	M13	DC1 input current	A*10(2)	✓
0xC02E	M14	Reserve	A*10(2)	✓

0xC02F	M15	Input Power A	KW*100(1)	√
0xC030	M16	Reserve	KW*100(1)	√
0xC031~ 0xC032	M17, M18	Total Output Power	KW-H(3)	√
0xC033~ 0xC041	M19~M33	Reserve		
0xC042	M34	Event Code 1~2	Code1 (4) Code2	
0xC043	M35	Event Code 3~4	Code3 (4) Code4	
0xC044	M36	Event Code 5~6	Code5 (4) Code6	
	M45~M96	Reserve		

P.S: (1) The number must be in unit*100 format.

Example: M04 = 1234 mean 12.34 KW

(2) The number must be in unit*10 format.

Example: M04 = 1234 mean 123.4 A

(3) The data is $(0xC031 * 65536) + 0xC032$.

Example: $0xC031 = 1234$, $0xC032 = 5678$, Total Power = $1234 * 65536 + 5678$.

(4)Event Code:

Code1/Code2/Code3/Code4/Code5/Code6			
Alarm/Error	high byte	low byte	CODE
Error	1000	0000	Er00
	1000	0001	Er01

	1010	1110	Er46
	1010	1111	Er47
Alarm	1100	0000	AL00
	1100	0001	AL01

	1101	1111	AL31
	1110	0000	AL32