Modbus V4 for GCAU V2

BN 00 1731/01/01
20.02.12

OPERATION MANUAL

AEG
POWER SOLUTIONS

CONTENTS

1 COMMUNICATION AND PROTOCOLS

The Protect RCS series has a communication feature allowing the customer to dialog with one or more systems. These connections can be point to point, as in EIA232, or multidrop configuration as in EIA485.

● Glossary:

ASCII: American Standard Code for Information Interchange
CRC: Cyclic Redundancy Check
CR: Carriage Return (ASCII 13)
EIA: Electronics Industries Association – recommendation body
GCAU: Generic Control and Alarm Unit
I/O: Inputs / outputs
LF: Line Feed (ASCII 10)
LRC: Longitudinal Redundancy Check
LSB: Least Significant Byte
MSB: Most Significant Byte
NRC: Network Report Controller
RTU: Remote Terminal Unit protocol, a compact binary message format used in Modbus
RS: Recommended Standard. Not used and replaced by EIA since mid nineties
SCADA: Supervisory Control And Data Acquisition
SPG: SPS Protocol for the GCAU i.e. AEG Protocol for the GCAU NRC
TBA: to be advised.

Numeric quantities are expressed in decimal. When expression is needed in a binary form a small b is appended (e.g. 1011 0111b for decimal 183) and 0x precedes an hexadecimal quantity (e.g. 0x1fff for decimal 8,191).

Note: Modbus, simply referred to as simply Modbus in this document, was a registered trademark of 'Schneider Automation'. Since April 2004, it is a trademark of the Modbus-IDA organization (http://www.modbus-ida.org). This organization made this standard publicly available ever since.

● Reference Document:

Modicon *Modbus Protocol Reference Guide PI-MBUS-300 Rev. J – June 1996*
Availabe as a PDF here: http://www.modbus-ida.org/docs/PI_MBUS_300.pdf
also here: http:// www.modbustools.com/PI_MBUS_300.pdf

● Physical Layers:

. Serial EIA232 or EIA485 Customer port.

● Supported Protocols:

. Modbus™ RTU and ASCII.

. AEG Maintenance Protocol also referred to as SPG.

When your system is equipped with the serial communication option, you can integrate your systems in a SCADA system or you can use our advanced remote control software (Winsparc, WinSite).

The option comes as a piggy-back board which is connected to the extension bus of the control card.

Note:

Do not attempt to connect or remove this card when the system is powered up. This can cause electrical shock. It can damage the control card and result in losing your system performance.

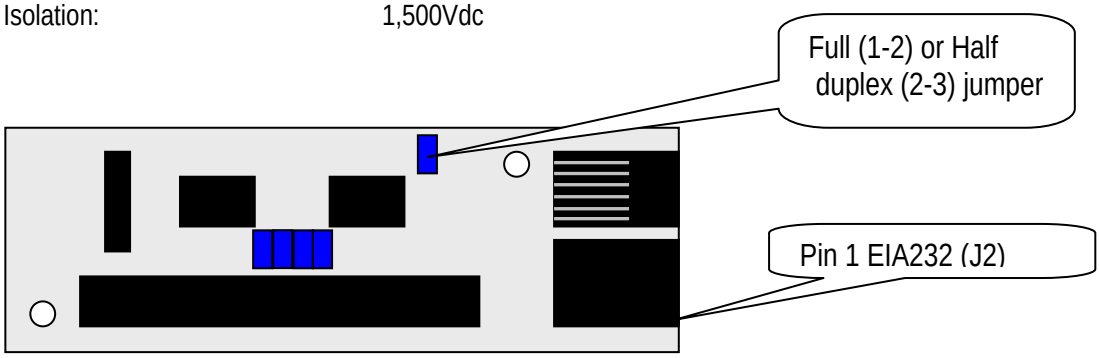
If an upgrade is necessary this must be carried out by qualified persons experienced with the system.

The communication ports are fully isolated from the chargers output and/or internal voltages.

Through the programming mode of the system, the user can change the communication parameters such as baudrate, slave number etc. if required.

2 SPECIFICATIONS

Hardware:	EIA232 or EIA485
Speed:	1,200 – 38,400 Baud
Protocol :	Modbus ASCII or RTU
	SPG (AEG Maintenance protocol)
Communication mode:	Half (2-wire) or full duplex (4-wire)
Isolation:	1,500Vdc



J3 EIA485 Connector – type: RJ6/6	
Pin	Function
1	RX+
2	RX-
3	TX-
4	TX+
5	Bias – (1K to Gnd)
6	Bias + (1K to +5V)

J2 EIA232 Connector – type: RJ8/8	
Pin	Function
1	NC
2	NC
3	NC
4	Signal Ground
5	RxD
6	TxD
7	CTS
8	RTS

Note: J2 pinout is compatible with EIA-561 specification and also with standard RJ-DE9 cords used with other AEG Power Systems controllers such as the ACM1000, ACM1D, NRC30, and NRC50.

3 CUSTOMER COMMUNICATON PORT

The Customer communication port is an add-on card with EIA232 and EIA485 hardware. The ports are isolated from the chargers output.

Two protocols are supported on this port: Modbus and our Maintenance protocol (SPG).

The selection of the protocol is configurable. A dynamic protocol switch can be selected. Then, the stream of incoming data will be analyzed. Whether it is an SPG or ASCII Modbus message, it will be answered accordingly.

3.1 Modbus

This port is used to interface the charger with the customer SCADA system.

The Customer serial port will be a software controlled isolated EIA232 or EIA485.
The standard communication protocol used will be MODBUS.

The following data and commands can be retrieved using Modbus:

- ✓ All analogue measurements
- ✓ Charge status
- ✓ Remaining charge time
- ✓ Active Alarms
- ✓ Alarm list
- ✓ History list
- ✓ Date/time
- ✓ Reset alarms
- ✓ Commissioning charge
- ✓ Highrate charge
- ✓ Floating charge
- ✓ Battery test on/off
- ✓ Clear History
- ✓ Stop/Start

Mapping of above data is given later on in this document.

3.2 MODBUS Specification

3.2.1 Modbus Format (RTU or ASCII)

The Modbus protocol can operate in RTU or ASCII mode. The following example shows both formats. The example shows the Read Holding Registers 40,108 – 40,110 addressed to slave equipment #06. The messages for RTU or ASCII will look as follows:

Query Message	RTU	ASCII	
Header	(none)	: (colon)	
Slave address	0000 0110b	0	6
Function	0000 0011b	0	3
Starting Register	0000 0000b	0	0
	0110 1011b	6	B
Quantity of Registers	0000 0000b	0	0
	0000 0011b	0	3
Error Check	0111 0101b 1010 0000b	8	9
Trailer	(none)	CR	LF
	8 Bytes Total	17 Bytes Total	

Notes: in this document we only deal with holding registers which are read/write 16-bit entities. Modbus defines other I/O's such as input registers, output registers, coils, etc. It is not necessary to explain the differences in further details here. Holding registers are the most common I/O's dealt with and are read and written with the functions numbers seen below. Holding registers are numbered from 40,001 to 49,999. Their addresses in the messages start from 0. In the example, number 40,108 is at address 6B hexadecimal (107 which is 40,108 less 40,001). Note also that quantities in messages are encoded MSB first.

Response Message		RTU	ASCII	
Header		(none)	: (colon)	
Slave address		0000 0110b	0	6
Function		0000 0011b	0	3
Byte Count		0000 0110b	0	6
Data (3 x 16-bit registers)	MSB	0000 0010b	0	2
	LSB	0010 1011b	2	B
	MSB	0000 0000b	0	0
	LSB	0000 0100b	0	4
	MSB	0000 0000b	0	0
	LSB	0110 0011b	6	3
Error Check		0010 0011b	5	D
		0100 1001b		
Trailer		(none)	CR	LF
		11 Bytes Total	23 Bytes Total	

In all cases the ASCII message will be about twice the length in bytes compared to the RTU message.

The error check is either a CRC-16 in case of RTU (see brief introduction below) or an LRC in case of ASCII. The LRC is the modulo 256 arithmetical opposite of the sum of all bytes after the column sign and before the LRC itself. In the example above: 0x06+0x03+0x02+0x2b+0x00+0x04+0x00+0x63=0xa3 and the opposite byte of 0xa3 is 0x5d.

3.2.2 Slave address

Slave number: 1 to 247. Broadcast is not supported.

3.2.3 Modbus functions

The GCAU only supports the two following functions:
- 03 Read holding registers
- 16 Preset multiple holding registers

This limitation is compliant to conformance class 0 as per Open Modbus Specification from Modicon.

● *Function 03, Read Holding Registers*

This function is used for the following purpose :

- Read the discrete statuses
- Read the analog inputs
- Read the events history list
- Read the software version information

If any combination of starting address and requested number of register exceeds these ranges, the reply will contain an exception code 02 (Illegal data address).

Notes:

The alarm and events history list is not entirely accessible in the register mapping in a single read. Only one history list item can be read at a time from dedicated GCAU registers. Since the GCAU may contain a completed list of alarms and events in its memory, and the GCAU can only hold 1 entry at a time, you must first request a history item by writing in a special register holding the current history item number (starting from number one) then you can read that history item.

- ***Function 16, preset multiple registers***

This function is used for the following purposes :

- Request a history item
- Execute a command

Again if any combination of starting address and requested number of register exceeds valid ranges, the reply will contain an exception code 02 (Illegal data address).

Notes :

1. See previous note in section for the use of requesting an history item.

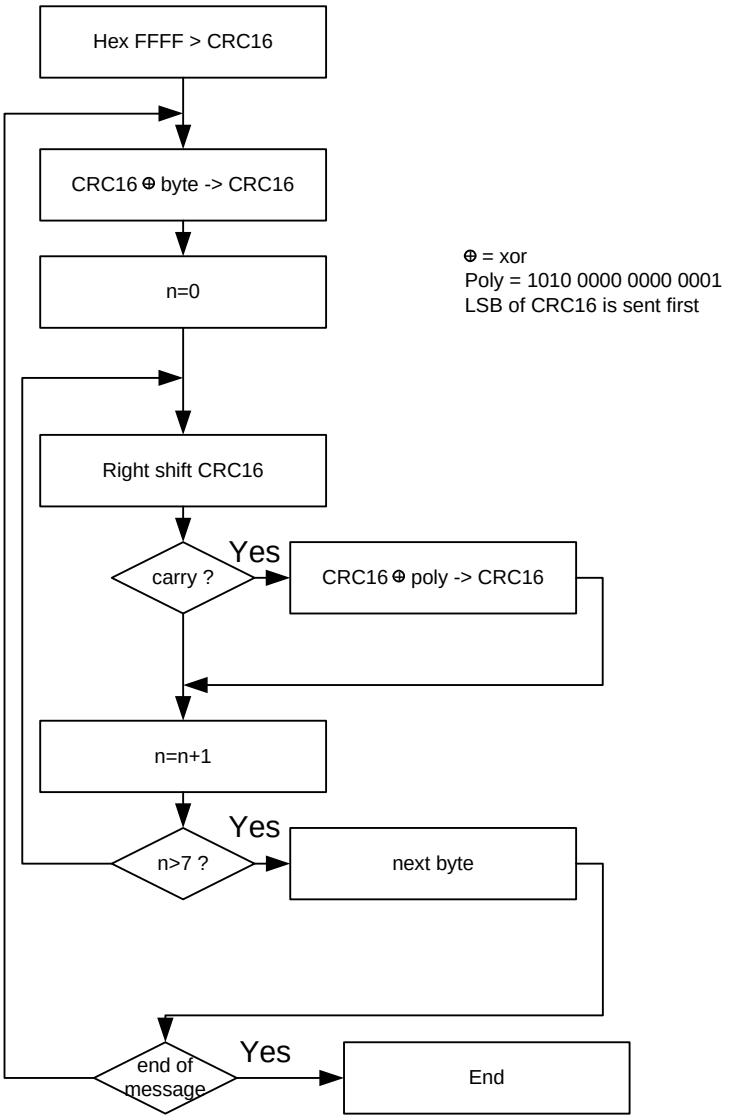
2. It is not allowed to write more than one register in the range at the same time in the range for executing commands (by using function 16, Multiple Preset). Doing so results in an exception code 02 (Illegal Data Address). In addition, if the data or registers do not match, exception code 03 (Illegal Data Value) is returned.

- ***Illegal function codes***

When a function code request is received that is not supported by the GCAU, it will be handled as an exception and it will follow the rules for exception responses within the Modbus protocol. To indicate that the response is a notification of an error, the high order bit of the function code is set to 1. Subsequently, an exception response code will follow, indicating the type of exception. In the case of an unsupported function, exception code 01 (Illegal Function) will be sent. Error codes 1, 2, 3 indicate message format is bad (permanent error) and error code 4 indicates the operation cannot be completed (temporary error).

- ***CRC-16 calculation***

CRC-16 used in Modbus is extensively explained in the standards and else where. Here is a flowchart that exposes the algorithm. Actually, a faster method exists using a pair of lookup tables.



3.2.4 Register locations

- ***Modbus mapping (version 4)***

Note about numbering and addressing registers:

The Modbus numbering scheme for holding registers starts from 40,001. In addition to Modbus holding register number (nMbHR), two other schemes of interest should be defined the Jbus register number (nJbR) and the actual register address that is encoded in Modbus message over the network (aRN). Here are the equations related these three schemes:

nMbHR = 40,001 + aRN,

nJbR = aRN.

Also, the JBUS usage imposes nJbR > 0.

- ***Register tables***

Alarm and event table:

Modbus Register (nMbHR)	Jbus Register (nJbR)	aRN (hex)	Variable	Range
40,002	1	0x1	High Mains Voltage	0 = No alarm, 1 = Alarm
40,003	2	0x2	Low Mains Voltage	
40,004	3	0x3	Charger Failure	
40,005	4	0x4	High Battery Voltage	
40,006	5	0x5	Low Battery Voltage	
40,007	6	0x6	High DC Voltage	
40,008	7	0x7	Low DC Voltage	
40,009	8	0x8	Ground Fault +	
40,010	9	0x9	Ground Fault -	
40,011	10	0xa	Spare 1	
40,012	11	0xb	Spare 2	
40,013	12	0xc	Spare 3	
40,014	13	0xd	Spare 4	
40,015	14	0xe	Spare 5	
40,016	15	0xf	Spare 6	
40,017	16	0x10	Spare 7	
40,018	17	0x11	Spare 8	
40,019	18	0x12	Charger Current Limit Indicator	
40,020	19	0x13	Battery Current Limit Indicator	
40,021	20	0x14	High Charger Current	
40,022	21	0x15	High Battery Current	
40,023	22	0x16	High Temperature	
40,024	23	0x17	Temperature Sensor Error	
40,025	24	0x18	Internal Communication Error	
40,026	25	0x19	Battery Test Fail	
40,027	26	0x1a	Battery Test Aborted	
40,028	27	0x1b	High Battery Temperature	
40,029	28	0x1c	High Float Current	
40,030	29	0x1d	Long Charge Time	
40,031	30	0x1e	No Power Supply Voltage	
40,032	31	0x1f	Battery in Operation	
40,033	32	0x20	Battery Symmetry Fault	
40,034	33	0x21	Equation 1	
40,035	34	0x22	Equation 2	
40,036	35	0x23	Equation 3	
40,037	36	0x24	Equation 4	
40,038	37	0x25	Equation 5	
40,039	38	0x26	Equation 6	
40,040	39	0x27	Equation 7	
40,041	40	0x28	Equation 8	
40,042	41	0x29	Equation 9	
40,043	42	0x2a	Equation 10	
40,044	43	0x2b	Equation 11	
40,045	44	0x2c	Equation 12	
40,046	45	0x2d	Equation 13	
40,047	46	0x2e	Equation 14	
40,048	47	0x2f	Equation 15	
40,049	48	0x30	Equation 16	

All holding registers in this table have a read-only access.

Measurements and status:

Modbus Register (nMbHR)	Modbus register (nJbR)	aRN (hex)	Variable	Range
40,100	99	0x63	Mains Voltage	VAC / 10
40,101	100	0x64	Battery Voltage	VDC / 10
40,102	101	0x65	Load Voltage	VDC / 10
40,103	102	0x66	Charger Current	Positive Amps / 10
40,104	103	0x67	Battery Current	Signed Amps / 10
40,105	104	0x68	Ambient Temperature	Signed Celsius
40,106	105	0x69	Battery Temperature	Signed Celsius
40,107	106	0x6a	Battery Symmetry Voltage	VDC/10
40,108	107	0x6b	Common Alarm Relay	0: negated, 1: asserted
40,109	108	0x6c	Earth Fault Impedance	Signed earth fault in k-Ohm/10
40,110	109	0x6d	Charge Status	0: Float 1: Highrate 2: Commissioning 3: Battery test 4: Charger Off
40,111	110	0x6e	Remaining Charge Time	Minutes
40,112	111	0x6f	AhMeter	%
40,113	112	0x70	Equation 1	16-bit signed integer value of equation result this result is multiplied by a factor defined by EQ configuration parameter 'ModbusMultiplier ' , rounded to nearest integer value and clamped to [-32,768, 32,767]
40,114	113	0x71	Equation 2	
40,115	114	0x72	Equation 3	
40,116	115	0x73	Equation 4	
40,117	116	0x74	Equation 5	
40,118	117	0x75	Equation 6	
40,119	118	0x76	Equation 7	
40,120	119	0x77	Equation 8	
40,121	120	0x78	Equation 9	
40,122	121	0x79	Equation 10	
40,123	122	0x7a	Equation 11	
40,124	123	0x7b	Equation 12	
40,125	124	0x7c	Equation 13	
40,126	125	0x7d	Equation 14	
40,127	126	0x7e	Equation 15	
40,128	127	0x7f	Equation 16	

All holding registers in this table have a read-only access.

History access:

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Range
40,150	149	0x95	Request/confirm history entry	1: most recent
40,151	150	0x96	Number of items in history list	Number of events
40,152	151	0x97	Year - Month	MSB: Actual Year-2000 LSB: Month (1-12)
40,153	152	0x98	Day - Hour	MSB: Day (1-31) LSB: Hours (0-23)
40,154	153	0x99	Minute - Second	MSB: Minutes (0-59) LSB: Seconds (0-59)
40,155	154	0x9A	Code of alarm/event	
40,156	155	0x9B	Setpoint or data value	

All holding registers except first one in this table have a read-only access.
First register in this table must be preset for the other to be updated.

Commands:

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Value definition
40,200	199	0xC7	Charge highrate/float	2 = highrate, 1 = float 0 = ineffective
40,201	200	0xC8	Alarm acknowledgment	1 = proceed 0; 2 = ineffective
40,202	201	0xC9	Rectifier shut down/start up	1 = shut down 2 = start up 0 = ineffective
40,203	202	0xCA	Clear history list	1 = proceed 0; 2 = ineffective
40,204	203	0xCB	Start/Stop battery test	1 = Stop 2 = Start 0 = ineffective
40,205	204	0xCC	Set Ah meter to 100%	1 = Set to 100% 0; 2 = ineffective
40,206	205	0xCD	Commissioning/Float	1 = float 2 = Commissioning 0 = ineffective

All holding registers in this table are write only.
Writing to more than one holding register in this table is not allowed and will result in an explicit error.
For commands to be executed, a protection mechanism exists based on a special configuration parameter. CmdCookie (defined in configuration object COMMUN2).
The command will be taken into account if the value in the Modbus frame less the Modbus CmdCookie equals one of possible definition values for the command at stake. For example, if we want to initiate a highrate charge (value 1 for this) and if Modbus CmdCookie equals 1234, then we must set the proper holding register to 1235.

Modbus is often a middleware to other industrial protocols such as Profibus-DP, etc…
In this case, outputs are virtualized values that are constantly updated in refresh cycles.
Then we must ensure commands, constantly written to in these circumstances, do not constantly trigger actions.

Some precautions have been implemented to allow pass commands through such automations:

- There always exists an "ineffective" value for each command that enables the client to constantly assert they don't want to start any command,
- As network and process latencies always exist it is difficult to make sure that no two successive commands with an active value (for example: start battery test) will trigger two equivalent actions (and then fill the history list with worthless events, etc…). In order to avoid this, a time filter (Configuration parameter COMMUN2.ModbusCmdTimeFilter), ensures that a given command appearing with the same value within this time frame will be accepted and will have no effect on the system. It is then reasonable to set COMMUN2.ModbusCmdTimeFilter beyond the known refresh delay.

Note: setting COMMUN2.ModbusCmdTimeFilter to zero completely disables this time filter.

Software versions:

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Value definition
40,250-40,256	249-255	0xf9-0xff	Software version of the control processor	16 characters, zero terminated
40,257	256	0x100	Software version of the regulation co-coprocessor	MSB: major version LSB: minor version
40,248	257	0x101	Modbus table version	

All holding registers in this table have a read-only access.

Setup time and date:

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Value definition
40,259	258	0x102	Preload command	In order to set time and date, this register must be set to value of COMMUN2.ModbusCmdCookie When read it always returns zero
40,260-40,261	259-260	0x103-0x104	Date and time in local time given in seconds since 2000/1/1 at 00:00:00	This 32- bit value is coded as Most Significant Word at first address and Less Significant Value at second address

Input variables for equations:

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Value definition
40, 262	261	0x105	First input register for equations	0 to 65,535
40, 263	262	0x106	Second input register for equations	0 to 65,535

- These registers can be read by equations as MB1, MB2 variables.
- They are not stored in flash by the GCAU: after a reset, equations will read them as zero prior to being written through Modbus.
- All holding registers in this table have read-write access.

Reserved registers:

The following registers are reserved. They are accessible but still unused. The reason for them is to implement Modbus gateways (such as Profibus) that map these registers in addition to the registers above. This allows those gateways to be compatible with future mappings.

Modbus Register (nMbHR)	Jbus register (nJbR)	aRN (hex)	Variable	Value definition
40,050-40,065	49-64	0x31-0x40	Reserved	Read only – returns 0xffff
40,129-40,149	128-148	0x80-0x94		
40,157-40,158	156-157	0x9c-0x9d		
40,207-40,209	206-208	0xce-0xd0		Read/write Accepts all Returns 0xffff
40,264-40,275	263-274	0x107-0x112		
40,300-40,331	299-330	0x12b - 0x14a		Read only – returns 0xffff

Note: Modbus registers 40,207-40,209 can only be written to, one by one.

4 PROGRAMMING THE COMMUNICATION PARAMETERS

If you need some communication settings, refer to this section for details.

4.1 The "Communication" Setup Group

(See section 2.2 for details about how to enter the programming mode).

Your system can be equipped with serial communication options which allow you to connect your system to a SCADA system or network. In this case, your system has an additional setup menu to program the communication parameters.
The communication card for the GCAU card is a plug-in device which is installed on the GCAU card as a piggy-back card.

Note
Removing or adding this card after the system left the factory must be carried out by qualified persons.

The communication setups incorporate five parameters e.g.

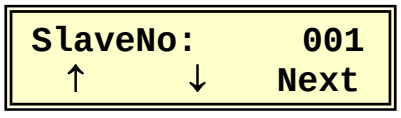
- ✓ *Slave Number*
- ✓ *Protocol*
- ✓ *Baudrate*
- ✓ *Hardware Configuration*
- ✓ *Format*

By default the communication setup values are the following:

- ✓ *Slave Number* *: 1*
- ✓ *Protocol* *: SPG*
- ✓ *Baudrate* *: 38400*
- ✓ *Hardware Configuration* *:* RS232
- ✓ *Format* *:* 8b-NP-1s

4.1.1 The "Slave Number" Configuration Option (Communication)

In case of multi drop configurations your system can be given a unique Slave number.
Only if the broadcasted slave number matches the programmed slave number the GCAU will respond. This allows sequential interrogation of multiple systems on one communication line.
The slave number can be configured between 1 and 247.

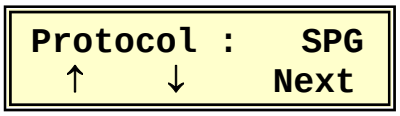


Use the up and down arrow keys to change the slave number setting.

4.1.2 The "Protocol Configuration" Option (Communication)

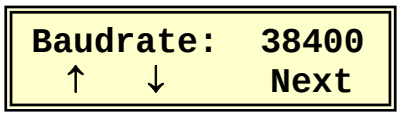
The protocol configuration option can be set for:

- ✓ Mixed: SPG+ASCII Modbus
- ✓ RTU Modbus
- ✓ ASCII Modbus
- ✓ SPG



4.1.3 The "Baudrate" Configuration Option (Communication)

You can configure the communication speed of your system with the Baudrate option.
The Baudrate defines the number of bits per second transferred over the communication bus.
You can set the communication rate at 1200, 2400, 4800, 9600, 19200 or 38400 Baud.
The GCAU system is fixed with 1 stopbit and no parity.



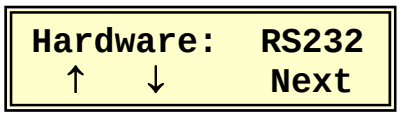
Use the up and down arrow keys to change the Baudrate.

4.1.4 The "Hardware Configuration" Option (Communication)

The hardware configuration option allows you to select the physical layer of the communication option. You can set it for:

- ✓ RS232 point to point with a maximum distance of 30 feet*
- ✓ RS485 Multi drop line up to 255 units maximum. Distance 1km*.

* *Depends on environment, cable type and communication speed.*

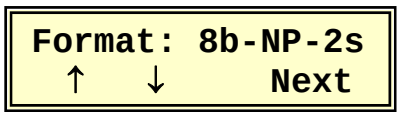


Use the up and down arrow keys to change the selection.

4.1.5 The "Format" Configuration Setting (Communication)

The Format setting allows you to select twelve values:

- ✓ 8 bits-no parity-2 stop. : 8b-NP-2s
- ✓ 8 bits-no parity-1 stop. : 8b-NP-1s
- ✓ 7 bits-even parity-2 stop. : 7b-EP- 2s
- ✓ 7 bits-even parity-1 stop. : 7b-EP-1s
- ✓ 7 bits-odd parity-2 stop. : 7b-OP-2s
- ✓ 7 bits-odd parity-1 stop. : 7b-OP-1s
- ✓ 8 bits- even parity-2 stop. : 8b-EP-2s
- ✓ 8 bits- even parity-1 stop. : 8b-EP-1s
- ✓ 8 bits- odd parity-2 stop. : 8b-OP-2s
- ✓ 8 bits- odd parity-1 stop. : 8b-OP-1s
- ✓ 7 bits-no parity-2 stop. : 7b-NP-2s
- ✓ 7 bits-no parity-1 stop. : 7b-NP-1s

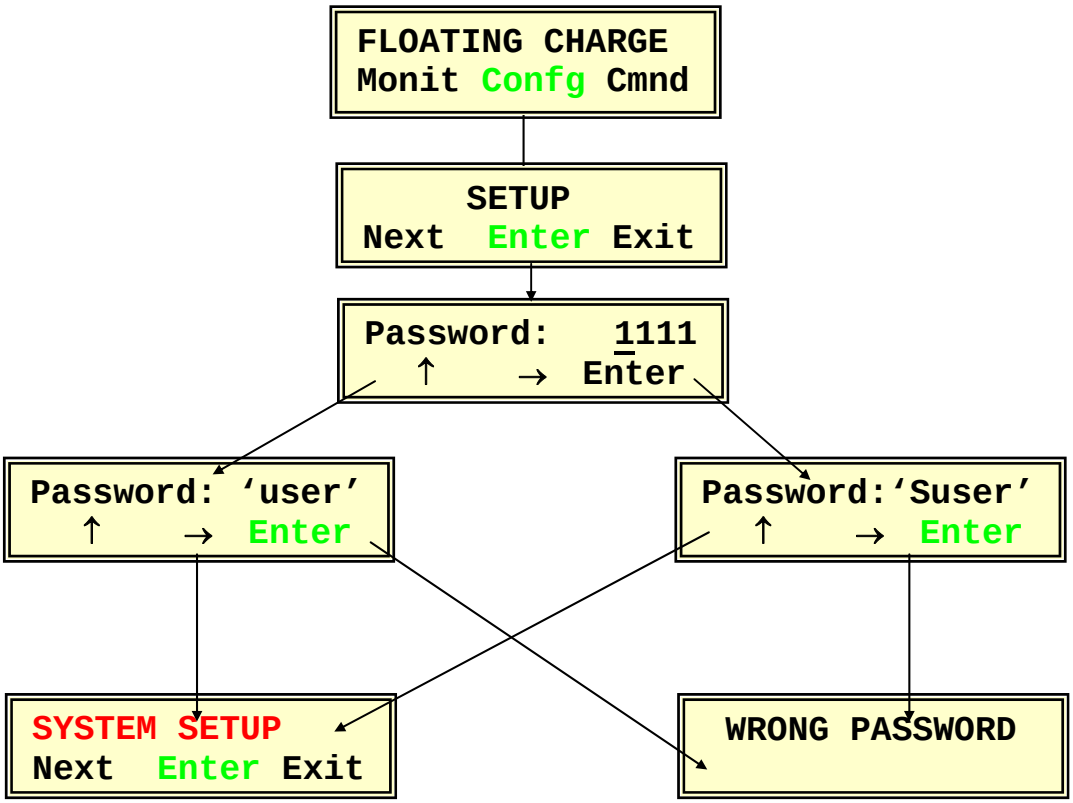


Use the up and down arrow keys to change the FORMAT setting.

4.2 Entering the program mode

To enter the program mode follow the instruction below. The defaults 1111 and 1211 are used for respectively the user and supervisor levels. Select the System Setup and search for the communication options.

LCD access, example.



Data in this document is subject to change without notice and becomes contractual only after confirmation. Photos are not contractual.

AEG POWER SOLUTIONS
10 Rue Jean Perrin
ZI N°1
37170 Chambray lès Tours
France
Tel.: +33 (0)2 47808860
Fax: +33 (0)2 47280719
www.aegps.com