The S-LOG is a complete system that saves on memory cards the data received from serial communication lines, specifically designed for an industrial use and for a direct mounting on automatic machines. The accepted memory cards are the SD (Secure Digital) and the MMC (Multi Media Card), that are the ones actally provided of the best features concerning price, handling and reliability. The S-LOG is a complete peripheral device, capable to receive serial data from an external system with different formats and to write them on the listed memories supports. The memory card is organized with FAT16 standard and it can be used by any personal computer, as an external removable disk. This really simplify the data transfer and use in fact each PC, provided of memory cards interface, can read, write, copy, open the available files, without any limits. The S-LOG is the suitable component in all the applications where data must be saved, even in big quantity up to 2 GigaBytes maximum limit, by using a communication serial line that is normally provided by each automation and control systems, available on the market. Among the typical application we can remind the replacement of printers with serial interface, the logging of data transmitted by measure and analysis systems, the momorization of status, alarms, and other operating conditions, the logging of data exchanged on a serial line for a following inspection and searching, etc.

The data saved on the memory card are exactly the same ones received from external system. When required the S-LOG can perform some autonomous operations on the saved data as: the partition of data on different files upon a time basis, the addition of strings that help the data identifications, the addition of formatters that simplify the following data usage and the addition of current date and time in order to temporally collocate the data.
IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsibility for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware. grifo® reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product. For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

- Attention: Generic danger
- Attention: High voltage
- Attention: ESD sensitive device

Trade Marks

GPC®, grifo®: are trade marks of grifo®. Other Product and Company names listed, are trade marks of their respective companies.
GENERAL INDEX

INTRODUCTION ................................................................................................................... ..... 1

HARDWARE AND FIRMWARE VERSION ............................................................................ 3

BOARD NAME ............................................................................................................................ 3

GENERAL INFORMATION ...................................................................................................... 4

REQUIREMENTS ................................................................................................................ .... 7

TECHNICAL FEATURES .......................................................................................................... 8

GENERAL FEATURES ............................................................................................................ 8

ELECTRIC FEATURES .......................................................................................................... 9

PHYSICAL FEATURES ......................................................................................................... 10

INSTALLATION ................................................................................................................... ..... 11

CONNECTIONS ...................................................................................................................... 12

CN3 - POWER SUPPLY CONNECTOR ........................................................................... 12

CN1 - SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR .......... 14

CN2 - MEMORY CARDS CONNECTOR ........................................................................ 18

JUMPERS ................................................................................................................................ 19

BACK UP .................................................................................................................................. 20

SIGNALLING LED .............................................................................................................. .. 20

ASYNCHRONOUS SERIAL LINE CONFIGURATION ................................................... 21

POWER SUPPLY .................................................................................................................... 22

SOFTWARE DESCRIPTION ................................................................................................... 24

OPERATIVE SELECTION MODE ...................................................................................... 24

WORK PARAMETERS ......................................................................................................... 26

CONFIGURATION MODE ................................................................................................... 27

ELECTRIC CONFIGURATION PROTOCOL ................................................................ 28

PHYSICAL CONFIGURATION PROTOCOL ................................................................ 28

LOGIC CONFIGURATION PROTOCOL ....................................................................... 28

RECEPTION AND SAVING DATA MODE ........................................................................ 31

ELECTRIC PROTOCOL FOR RECEPTION AND SAVING DATA ............................ 32

PHYSICAL PROTOCOL FOR RECEPTION AND SAVING DATA ............................. 32

LOGIC PROTOCOL FOR RECEPTION AND SAVING DATA .................................... 32

DATA SAVING ...................................................................................................................... 34

HANDSHAKE COMMUNICATION .................................................................................... 35

OPERATIONS ON MEMORY CARD .................................................................................. 36

DEMO PROGRAMS .............................................................................................................. 38

ERRORS ...................................................................................................................... ............ 38

NOTE FOR SYNCHRONOUS COMMUNICATION I2C BUS ........................................ 39

TIMEOUT IN COMMUNICATION ..................................................................................... 40
FIGURES INDEX

FIGURE 1: LOCATION OF HARDWARE AND Firmware VERSION ............................................................... 3
FIGURE 2: VIEW WITHOUT BOX ........................................................................................................ 5
FIGURE 3: CONSUMPTION TABLE ....................................................................................................... 9
FIGURE 4: LOCATION OF JUMPERS, CONNECTORS, LED, BATTERY, ETC. ........................................ 11
FIGURE 5: CN3 - Power supply connector .......................................................................................... 12
FIGURE 6: +5 Vdc regulated power supply connection ...................................................................... 13
FIGURE 7: 8+24 VAC alternate power supply connection (with .SW option) ............................... 13
FIGURE 8: 10+38 Vdc direct power supply connection (with .SW option) .................................. 13
FIGURE 9: CN1 Serial communication line and switching connector ............................................. 14
FIGURE 10: Connection example in RS 232 with three wires (without handshake) ...................... 15
FIGURE 11: Connection example in RS 232 with five wires (with handshake) ................................. 15
FIGURE 12: Connection example in TTL with three wires (without handshake) ............................ 15
FIGURE 13: Connection example in TTL with five wires (with handshake) .................................. 15
FIGURE 14: Connection example point to point in I2C BUS without power supply .................... 16
FIGURE 15: Connection example point to point in I2C BUS with power supply ............................ 16
FIGURE 16: Network connection example I2C BUS communication ............................................ 17
FIGURE 17: Components map components side and solder side .................................................... 18
FIGURE 18: Jumpers table ................................................................................................................ 19
FIGURE 19: Signalling LED ............................................................................................................... 20
FIGURE 20: Asynchronous serial RS 232 jumper configuration ...................................................... 21
FIGURE 21: Asynchronous serial TTL jumper configuration .......................................................... 21
FIGURE 22: Power supply EL 12 ...................................................................................................... 23
FIGURE 23: Work parameters definition .......................................................................................... 25
FIGURE 24: Performed operations .................................................................................................... 27
FIGURE 25: Asynchronous communication RS 232, TTL for configuration .................................... 29
FIGURE 26: Synchronous Communication I2C BUS for configuration ........................................ 30
FIGURE 27: Reception and saving data mode .................................................................................... 31
FIGURE 28: Asynchronous communication RS 232, TTL for data saving ..................................... 33
FIGURE 29: Synchronous communication I2C BUS for data saving ............................................. 33
FIGURE 30: Memory card insertion .................................................................................................. 37
FIGURE 31: Network of different I2C BUS devices ........................................................................ 41
FIGURE 32: Real time clock parameters .......................................................................................... 45
FIGURE 33: Available connections diagram .................................................................................. 47
FIGURE 34: Communication line parameter munication ............................................................... 46
FIGURE 35: Baud rate parameter ..................................................................................................... 48
FIGURE 36: Parity parameter ............................................................................................................ 49
FIGURE 37: Stop parameter ............................................................................................................. 49
FIGURE 38: Handshake parameter .................................................................................................. 50
FIGURE 39: Timeout parameter ....................................................................................................... 51
FIGURE 40: Complete view ............................................................................................................. 53
FIGURE 41: Duration parameter ...................................................................................................... 54
FIGURE 42: Action parameter ........................................................................................................ 55
FIGURE 43: Group time parameter ................................................................................................ 56
FIGURE 44: Data saving application with GAB H844 .................................................................... 56
FIGURE 45: Special sequence label parameter ............................................................................ 58
FIGURE 46: SPECIAL SEQUENCE FOR PREFIX PARAMETER ................................................................. 59
FIGURE 47: SPECIAL SEQUENCE OF SUFFIX PARAMETER ............................................................. 60
FIGURE 48: RS 232 CONNECTION WITH PC .................................................................................... 61
FIGURE 49: PC SERIAL SETTING WITH CONF_SLOG ................................................................. 62
FIGURE 50: ACQUIRE VERSION WITH CONF_SLOG ................................................................. 63
FIGURE 51: COMMUNICATION CONFIGURATION WITH CONF_SLOG ........................................... 63
FIGURE 52: SAVING CONFIGURATION WITH CONF_SLOG ........................................................... 64
FIGURE 53: GRAPHIC WITH SAVED DATA BY S-LOG ................................................................. 66
FIGURA A1: CONFIGURATION COMMANDS TABLE (1 OF 2) .................................................. A-1
FIGURE A2: CONFIGURATION COMMANDS TABLE (2 OF 2) .................................................... A-2
FIGURE B1: DEFAULT CONFIGURATION TABLE ........................................................................ B-1
FIGURE B2: DEFAULT JUMPERS CONFIGURATION TABLE ........................................................... B-2
FIGURE B3: OPTIONS TABLE ................................................................................................ B-2
FIGURE B4: CKS.AMP4 CONNECTION ACCESSORY ............................................................... B-2
FIGURE B5: AMP4.CABLE CONNECTION ACCESSORY ............................................................ B-3
FIGURE B6: CKS.AMP8 CONNECTION ACCESSORY ............................................................... B-3
FIGURE B7: AMP8.CABLE CONNECTION ACCESSORY ............................................................ B-4
INTRODUCTION

The use of these devices has turned - IN EXCLUSIVE WAY - to specialized personnel.

This device is not a safe component as defined in directive 98-37/CE.

Pins of module are not provided with any kind of ESD protection. Many pins of the card are directly connected to their respective pins of on board's components and these last are sensitive to electrostatic noises. So personnel who handles the product/s is invited to take all necessary precautions that avoid possible damages caused by electrostatic discharges.

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

The reported data are destined- IN EXCLUSIVE WAY- to specialized users, that can interact with the devices in safety conditions for the persons, for the machine and for the enviroment, impersonating an elementary diagnostic of breakdowns and of malfunction conditions by performing simple functional verify operations, in the height respect of the actual safety and health norms.

The informations for the installation, the assemblage, the dismantlement, the handling, the adjustment, the reparation and the contingent accessories, devices, installation, etc. are destined - and then executable - always and in exclusive way from specialized warned and educated personnel, or directly from the AUTHORIZED TECHNICAL ASSISTANCE, in the height respect of the manufacturer recommendations and the actual safety and health norms.

The devices can't be used outside a box. The user must always insert the cards in a container that respect the actual safety normative. The protection of this container is not threshold to the only atmospheric agents, but specially to mechanic, electric, magnetic, etc. ones.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the AUTHORIZED TECHNICAL ASSISTANCE directly.
To prevent problems during card utilization, it is a good practice to read carefully all the informations of this manual. After this reading, the user can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

grifo® provid this documentation "as is" without warranty of any kind. In no event shall grifo® be liable for indirect, special, incidental or consequential damages of any kind arising from any error in this documentation, including any loss or interruption of business, profits, use, or data. Moreover is not guaranteed the updating of the product for new computers or new operating systems, that will become available in the future.

All trademarks listed in this manual are copyright of the relative manufacturers.
HARDWARE AND FIRMWARE VERSION

This handbook make reference to printed circuit version 110705 and to firmware version 2.1 and following ones. The validity of the information contained in this manual is subordinated to the version numbers on the used panel, and so the user must always verify the correct correspondence between the notations. The version numbers are reported in several places on the electronic part of the product, and following figure shows the most accessible ones.

![Figure 1: Location of Hardware and Firmware Version](image)

Obviously if the version must be checked, then it must be extracted from the metallic container: the firmware version can be checked on bottom of the container too, near the configuration ordered. Moreover both numbers version can be also directly required to S-LOG by using dedicated command.

Normally the S-LOG is always supplied with the latest firmware version that is available but, for specific requirements, the user can receive also a different version; he must carefully specify this particular condition in the order.

In this manual there are information referred to other programs that built S-LOG too: each ones have proper number version that, if needed, is shown in this manual.

In case of necessity of technical assistance is important that user supply version numbers of his products.

BOARD NAME

It's important note and remember the product S-LOG is got through SDI 02 board where run a proper firmware thaperform functions described in next pages of the manual. The not corrispondance of board name is displayed only when the container is open or in some figures of the manual that show the printed board: in all these cases the user has to take SDI 02 indications as S-LOG.
GENERAL INFORMATION

The S-LOG is a complete system that saves on memory cards the data received from serial communication lines, specifically designed for an industrial use and for a direct mounting on automatic machines. The accepted memory cards are the SD (Secure Digital) and the MMC (Multi Media Card), that are the ones actually provided of the best features concerning price, handling and reliability. The S-LOG is a complete peripheral device, capable to receive serial data from an external system with different formats and to write them on the listed memories supports. A list of proper commands allow to configure the S-LOG in order to be linked and used by a large number of installed devices and in order to satisfy many data logger applications.

The memory card is organized with FAT16 standard and it can be used by any personal computer, as an external removable disk. This really simplify the data transfer and use in fact each PC, provided of memory cards interface, can read, write, copy, open the available files, without any limits. Some additional features as the presence of a Power Supply Section that accepts a wide range input, a Plastic Container that speed up the installation and a Real Time Clock that adds a time reference to saved data, complete the product with an optimum price/performance ratio.

The S-LOG is the suitable component in all the applications where data must be saved, even in big quantity up to 2 GigaBytes maximum limit, by using a communication serial line that is normally provided by each automation and control systems, available on the market. Among the typical application we can remind the replacement of printers with serial interface, the logging of data transmitted by measure and analysis systems, the memorization of status, alarms, and other operating conditions, the logging of data exchanged on a serial line for a following inspection and searching, etc. In all the applications the memory card is removed from S-LOG and then used by any other system capable to manage the FAT16 standard: here the data can be processed, displayed and registered in archives either by standard or dedicated programs.

The data saved on the memory card are exactly the same ones received from external system. When required the S-LOG can perform some autonomous operations on the saved data as: the partition of data on different files upon a time basis, the addition of strings that help the data identifications, the addition of formatters that simplify the following data usage and the addition of current date and time in order to temporally collocate the data.

The serial communication with the external system can be established through an Asynchronous Line buffered in RS 232 or TTL or alternatively through an I2C BUS Synchronous Line. The selection of the used serial line and of the relative parameters is comfortably executed with a specific Configuration Modality, directly accessible to the final user.

Hardware features of S-LOG, complete of possible options, are listed below:
- Modular plastic container DIN 50022 modulbox, M3 HC53 model
- Overall dimension: 90x53 mm frontal size; 58 mm height
- Mounting on DIN Omega rails DIN 46277-1 and DIN 46277-3
- Very low weight: only 84 g.
- Connector for memory cards in SD and MMC formats
- Circuit for recognition of memory card Insertion and its Write Protect status
- 1 bicoline LED for visual signalations, driven with different modalities
- Real Time Clock backed by proper Lithium battery
- Asynchronous communication serial line configurable in RS 232 or TTL
- I2C BUS synchronous communication serial line
- 1 digital output in Open Collector, connected to internal connector
- 2 handy connectors for a fast wiring of:
  - Power supply
  - Communication lines
- Power supply section capable to support also small external loads
- Two different power supply types:
  - 5 Vdc regulated
  - Unregulated wide range from 10 to 38 Vdc or from 8 to 24 Vac
- Minimum required power equal to about 0.390 W + memory card consumption
- On board protection against voltage peaks by TransZorb
- Customized firmware and program under specific request of the user
- For specific requirements about functionality and prices, please contact directly grifo®

**FIGURE 2: VIEW WITHOUT BOX**

In order to simplify the use of S-LOG the grifo® has provided a list of functionalities and features that speed up the working setup of the final user. These general features, are summarized below:

- Management of memory cards with capacity up to 2 GBytes.
- Memory cards managed as a Disk with FAT16 format.
- Data on disk are organized in Folders and File, compatible with the most diffused operating systems for PC (MS-DOS, Windows, Mac-OS, etc.).
- File and Folders names are managed with 8.3 format (8 characters for name and 3 for extension): the extended names are converted into this format.
- Management of current Date and Time from RTC that are used as a temporal reference for the saved data.
- Possibility to set and acquire the Real Time Clock (RTC) and all its temporal parameters (hour, minutes, seconds, day, month, year).
- Recognition and management of the memory card Insertion.
- Recognition and management of the memory card Write Protect selector.
- Memory card status visualized by Bicolour LED, that shows up to 5 different conditions.
- Complete management of the possible Errors: 12 general and 9 for the disk.
- The Errors are acquired and cleared through specific commands in order to simplify the solution of possible damages and malfunctions.
- Tens of Commands dedicated to typical configuration requirements (see details in following table).
- All the configuration Commands, and the relative Parameters and Responses, are exchanged through the synchronous Communication Line by following a proper logic, physic and electric protocol. The communication logic protocol for configuration is ASCII in order to simplify and speed up the use.
- The S-LOG acts as a Data logger that saves on memory card all the Data Received, with the modalities preset with configuration.
- Possibility to define the Action performed during the memory card Insertion.
- The name of the file generated by S-LOG can be defined in configuration, complete of possible saving Folder.
- Data saved on a Single file or on Different files associated to a time period.
- The Duration time for different files saving can be Hour, Day, Month or Year and the file name is the start acquisition time, with the yyyymmdd.hh format.
- Possibility to automatically add a Label at the beginning of the generated file, as an header that supply informations on the saved data.
- Recognition and management of a Data Group based on time interval and defined in configuration; the end of a group and the beggingin of the following one coincides with a time interval with no data received from external unit.
- Possibility to automatically add a Prefix at the beginning of a data group.
- Possibility to automatically add a Suffix at the end of a data group.
- Label, prefix and suffix are defined in configuration with different Attributes: Messages, Strings, current Date, current Time, current Milliseconds abd some Formatters. By defining these attributes the user can decide to save additional information on the file and the format of these information.
- Possibility to select the Physic and Electric Protocols, that will be used for the reception of the data to save, through proper configuration commands.
- The provided Electric Protocol are:
  - Asynchronous Serial Line buffered in RS 232 or TTL.
  - I2C BUS Synchronous Serial Line.
- Selectable Physic Protocol for asynchronous serial communication:
  - 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 115200 Bauds;
  - 8 bits for character;
  - No, Even, Odd Parity;
  - 1 or 2 Stop bits.
  - Hardware (RTS, CTS), Software (XON, XOFF) or repeat software, communication Handshaking.
- Selectable Physic Protocol for I2C BUS synchronous serial communication:
  - Bit Rate from 225 to 460799 bps;
  - Slave mode;
  - S-LOG Slave Address from 00H to FEH at step of 2, equal to 128 different values.
- Serial Network connection through I2C BUS synchronous protocol.
- Maximum Speed for receiving and saving data on memory card are:
  - 5000 Bytes/second, with asynchronous serial line at 115200 Baud
  - 6000 Bytes/second, with synchronous I2C BUS serial line at 400 KBits/second
- Supplied with Demo Programs that shows and simplify the configuration. One of these can be executed on PC and allows the management of S-LOG connected through a free RS 232 serial line of computer.
- The interface can be configured even by hand by using any sistem with an asynchronous serial line and a Terminal Emulation program (i.e. a standard PC that executes Hyperterminal).
REQUIREMENTS

Following there is a bill of necessary material to use S-LOG:

a) Purpose documentation, in other words, this user manual.

b) A power supply compatible with needed configuration (please see POWER SUPPLY paragraph for details).

c) An external system able to talk one of three serial interfaces of S-LOG, according with physical and electric protocol selected.

d) A serial communication cable able to communicate with the electric standard selected, between external device described at point c and S-LOG. To realize this connection please see the several examples and figures on this manual.

e) One memory card SD or MMC up to 2 GB capacity, in FAT or FAT16 format.

Moreover isn't needed but is recommended:

f) One personal computer able to execute Conf_SLOG program for S-LOG configuration and make it ready to work in the system to realize. This PC must have following basic features:

   Personal Computer: IBM compatible
   RAM memory: ≥ 64 MBytes
   Hard disk : ≥ 8 MByte liberi
   Video board: ≥ 800x600 pixels, 65536 colours
   Monitor: Colours
   Interfaces: One serial line COM free
   Operating system: Windows 98, ME, 2000, XP

   Conf_SLOG program doesn't have proper drivers but uses those ones already presents in the operating system so every item above described has to be configurated and working according with the operating system.

To realize the final application, there are available more examples programs made for talk with S-LOG. The customer has to find before his interesting components and then to use them how is described in the same programs or in following chapter of this manual.

Some of described elements, can be downloaded from grifo® web sites.
TECHNICAL FEATURES

GENERAL FEATURES

Resources:
- 1 status bicolour LED
- Asynchronous serial line, full duplex in RS 232, or TTL
- Synchronous I2C BUS line slave mode
- RTC backed by lithium battery
- Interface and connector for memory cards SD and MMC
- Recognizing inserting and protection memory card
- Configuration mode selecting
- Wide range power supply section

Turn off time:  70 ms
Timing resolution:  20 ms
LEDs intermittent time:  640 ms
Tempo rip. handshake software:  1000 ms
Communication timeout:  0÷5000 ms or disabled
Data group time:  0÷5000 ms or disabled
Storage on file duration:  Hour, Day, month, Year

Network unity number:  128 with synchronous serial I2C BUS

Communication:
Select between: synchronous (I2C BUS)
Default: asynchronous

Asynchronous communication protocol (RS 232, TTL):
- Baud rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Bit per character: 8
- Parity: off, even, odd
- Stop Bit: 1, 2
- Handshake: off, hardware, software, software repeated
Default: 19200 Baud, 8 Bit, None parity, 1 Stop, None handshake

Synchronous communication protocol (I2C BUS):
- Bit rate: from 225 to 460.799 bits per second
- Mode: Slave
- Slave Address: from 00H to FEH step of 2
Default: Slave Address = 128 = 80H

Reception buffer size:  250 characters
Command buffer commands:  50 characters
Max. file name length:  35 characters
Max. label, prefix, suffix length: 30 characters

Maximum reception speed and data storage (*1): 5.000 Bytes/sec: asynchronous serial at 115.200 Baud; 6.000 Bytes/sec: synchronous serial I2C BUS at 400 KBits/sec

Memory card format: FAT, FAT16

(*1) Speed depend on type and status of used memory card.

ELECTRIC FEATURES

Power voltage: +5 Vdc ± 5%
or 10÷38 Vdc, 8÷24 Vac (option) (*2)

Power consumption: please see next table (*2)

<table>
<thead>
<tr>
<th>Measure conditions</th>
<th>consumption max. +5 Vdc</th>
<th>Consumption max. 10÷38 Vdc 8÷24 Vac</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-LOG without memory card, not connected to other systems</td>
<td>44 mA</td>
<td>0,23 W</td>
</tr>
<tr>
<td>S-LOG with memory card, connected, at work</td>
<td>60 mA</td>
<td>0,39 W</td>
</tr>
</tbody>
</table>

Output power supply voltage: +5.0 Vdc

Current available on +5Vdc output: 400 mA - consumption max. +5 Vdc (*2)

RS 232 extravoltage protection: ±15 KV

I2C BUS pull up resistor: 4.7K Ω

Back up battery: 3 V Lithium; 180 mAh; CR 2032 model

Back up current: 3.2 μA

(*2) The data are referenced to 20 C° environmental work temperature (for further information please refer to chapter POWER SUPPLY).

The previous table lists the S-LOG power consumption referred to typical operating condition; for the wide range power supply are described the required power, in place of the current, already corrected with efficiency factor of the on board power supply section.
PHYSICAL FEATURES

Size: 90 x 53 x 58 mm (L x A x P) with container
      86 x 50 x 30 mm (L x A x P) without container

Container: DIN 50022 modulbox, M3 HC53 model, plastic

Mounting: On Omega rail DIN 46277-1 and DIN 46277-3

Weight: 84 g

Temperature range: From 0 to 50 °C

Relative umidity: 20% up to 90% (without condense)

Connectors: CN1: 8 pins AMP MODU II, vertical, male, pitch 2,54
              CN2: SD, MMC, connector 90 degree
              CN3: quick release screw terminal, 2 pins, male, pitch 3,5
INSTALLATION

In this chapter there are the information for a right installation and correct use of S-LOG. In detail there are the locations and functions of each connector, of the user settable jumpers, of the battery and any other information concerning hardware configuration. All components that are not described in this chapter are for internal use and they can't be used by user.

FIGURE 4: LOCATION OF JUMPERS, CONNECTORS, LED, BATTERY, ETC.
CONNECTIONS

The S-LOG is provided of 3 connectors that are used for all field connections and with other system data boards to realize. Below are reported the pin outs, the meaning of the connected signals (including their directions) and some connection examples, that simplify and speed the installation phase. In addition the figure 4 show the connectors position on the board, to simplify their recognitions.
All the connectors are accessible through a proper breaking in the container that allows comfortable insertion and deinsertion.

CN3 - POWER SUPPLY CONNECTOR

CN3 is a vertical, 2 pins, male, quick release screw terminal connector, with 3.5 mm pitch.
On CN3 must be connected the single power supply voltage for the module when is requested, \textit{SW} option and is present the power supply section, able to accept a wide range voltage.

![Figure 5: CN3 - Power Supply Connector](image)

Signals description:

- \( \text{Vac} \) = I - AC power supply lines connected to on board switching section; these signals must be in the range \( 8\text{÷}24 \text{ Vac} \).
- \( +\text{Vdc pow} \) = I - DC power supply lines connected to on board switching section \( (+10\text{÷}+38 \text{ Vdc}) \) or stabilized \( (+5 \text{ Vdc}) \) voltage connected to on board logic, according to ordered configuration.
- GND = - Ground signal for DC power supply.

\textbf{NOTE} For further information about power supply configurations, please refer to paragraph \textit{POWER SUPPLY}, to next figures and the CN1 -SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR paragraph.
**Figure 6:** +5 Vdc regulated power supply connection

**Figure 7:** 8÷24 Vac alternate power supply connection (with .SW option)

**Figure 8:** 10÷38 Vdc direct power supply connection (with .SW option)
CN1 - SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR

CN1 is a 8 pins, male, vertical, AMP MODU II 4+4 connector with 2.54 mm pitch. On this connector there are the signals for communication in RS 232 or TTL, performed through hardware serial line of S-LOG. Signals placement has been designed to reduce interferences and electrical noises and to simplify the connections with other system; the electric protocols follow the CCITT directives of the used standard.

Moreover on CN1 there is the regulated power voltage +5 Vdc that can be supplied or drawn; it depend on S-LOG configuration.

Female connector for CN1 is directly available between grifo® accessories, and it can be ordered by using the codes CKS.AMP8 or AMP8.Cable, as described in APPENDIX B of the manual.

**FIGURA 9: CN1 SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR**

**Legenda:**

SDA = I/O - Data signal of I2C BUS.
SCL = I/O - Clock signal of I2C BUS.
RTS RS232 = O - Handshake reception line in RS 232.
RX RS232 = I - Reception line in RS 232.
TX RS232 = O - Transmission line in RS 232.
CTS RS232 = O - Handshake transmission line in RS 232.
RTS TTL = O - Handshake reception line in TTL.
RX TTL = I - Reception line in TTL.
TX TTL = O - Transmission line in TTL.
CTS TTL = O - Handshake transmission line in TTL.
+5 Vdc = O - Power supply line +5 Vdc.
GND = - Ground line.
N.C. = - Not connected.

For details on board power supply, please see CN3 - wide range POWER SUPPLY CONNECTOR, POWER SUPPLY paragraphs and see figures 6+8.

For details about asynchronous comunication we recommend to examine the ASYNCHRONOUS SERIAL LINE COMMUNICATION paragraph and to see figures 10+16 that show connection examples.
**Figure 10:** Connection example in RS 232 with three wires (without handshake)

**Figure 11:** Connection example in RS 232 with five wires (with handshake)

**Figure 12:** Connection example in TTL with three wires (without handshake)

**Figure 13:** Connection example in TTL with five wires (with handshake)
For details about I2CBUS we recommend to examine the NOTE FOR SYNCHRONOUS I2CBUS COMMUNICATION paragraph and to see figures 14+16 that show connection examples both in point to point and network.

**FIGURE 14: CONNECTION EXAMPLE POINT TO POINT IN I2C BUS WITHOUT POWER SUPPLY**

**FIGURE 15: CONNECTION EXAMPLE POINT TO POINT IN I2C BUS WITH POWER SUPPLY**

In connecting power supply is important to know that +5 Vdc voltage has to be generated by only one connected system and this one is able to supply powerful for the other system too. The S-LOG can get and supply power voltage on CN1 connector too, as described in POWER SUPPLY paragraph, and for this purpose that J4 jumper can connect pin 8 of the connector.
Figure 16: Network connection example I2C BUS communication

Please remind that in a I2C BUS must be present two pull up resistor at the limits of the line, one next to master unity and one next slave unity.

On S-LOG are always present the resistor (*1) and their value is reported in the ELECTRIC FEATURES paragraph. User has to choose and/or configure I2C BUS devices to connect, keeping in mind this feature. In detail on S-LOG these resistor have to be disconnected from the unity for those line as shown in the previous figure on slaves 1 and 2.

For further information see "THE I2C-BUS SPECIFICATION" document of PHILIPS Semiconductors.
CN2 - MEMORY CARDS CONNECTOR

CN2 is a connector for memory cards like SD and MMC type. In this chapter there are no description of connector pin out, in fact is perfectly compatible with used standard on memory card, but only a short bill of expedient and recommendation for use:

a) The max. capacity of utilizable cards is 2 GigaBytes.
b) Utilizable cards have to be provided of SPI interface.
c) Cards have to be formatted before with FAT or FAT16 standard.
d) In case of problems of insertion in CN2 connector, verify the integrity of the same connector and than make sure of not finding not related pieces by mistake.
e) When the memory card is provided of mechanical adapter, before insertion of the card into the adapter, be sure that the anchorage is between card and adapted is good. A bag connection with the adapter can cause a bad inserting or removing and/or create malfunctions.
f) Avoid that card electric contacts get dirty or in touch with any substances.
g) Before insertion make sure that writing protection selector is disableb, otherwise the S-LOG shows error status and doesn’t save any data.

![Figure 17: Components map components side and solder side]
JUMPERS

On **S-LOG** there are nine jumpers for card configuration and by connecting them, the user can perform some selections that regards the working conditions of the card. Here below there is the jumpers list and relative functions in the possible connection modalities:

<table>
<thead>
<tr>
<th>JUMPER</th>
<th>CONNESSIONE</th>
<th>UTILIZZO</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>posizione 1-2</td>
<td>Seleziona la modalità di configurazione.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Seleziona la modalità di normale lavoro (ricezione e salvataggio dati).</td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td>non connesso</td>
<td>Non collega batteria di bordo BT1 alla circuiteria di back up.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connesso</td>
<td>Collega batteria di bordo BT1 alla circuiteria di back up.</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>posizione 1-2</td>
<td>Non collega tensione di alimentazione +5 Vdc al connettore CN1.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Collega tensione di alimentazione +5 Vdc al connettore CN1.</td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>posizione 1-2</td>
<td>Collega pin 7 di CN1 al segnale RTS RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Collega pin 7 di CN1 al segnale RTS TTL.</td>
<td></td>
</tr>
<tr>
<td>J6</td>
<td>posizione 1-2</td>
<td>Collega pin 1 di CN1 al segnale CTS RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Collega pin 1 di CN1 al segnale CTS TTL.</td>
<td></td>
</tr>
<tr>
<td>J7</td>
<td>posizione 1-2</td>
<td>Collega pin 3 di CN1 al segnale TX RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Collega pin 3 di CN1 al segnale TX TTL.</td>
<td></td>
</tr>
<tr>
<td>J8</td>
<td>posizione 1-2</td>
<td>Collega pin 5 di CN1 al segnale RX RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Collega pin 5 di CN1 al segnale RX TTL.</td>
<td></td>
</tr>
<tr>
<td>J9</td>
<td>posizione 1-2</td>
<td>Collega segnale di ricezione seriale della scheda al driver RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Non collega segnale di ricezione seriale della scheda al driver RS 232.</td>
<td></td>
</tr>
<tr>
<td>J10</td>
<td>posizione 1-2</td>
<td>Collega segnale di handshake seriale CTS della scheda al driver RS 232.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>posizione 2-3</td>
<td>Non collega segnale di handshake seriale CTS della scheda al driver RS 232.</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 18: JUMPERS TABLE**

To recognize the valid connections and locations of these jumpers, please refer to the board printed diagram (serigraph) or to figure 4 of this manual, where the pins numeration is listed.

In previous table the "*" denotes the default connection, or on the other hand the connection set up at the end of testing phase, that is the configuration the user receives.
Further information about purpose of the **S-LOG** jumpers are reported in the following paragraphs, that describe the sections where the same jumpers are used.

**BACK UP**

When **S-LOG** contain a real time clock (**RTC**) feature, it is provided of a lithium battery that keeps hours, minutes, seconds, days, months and years, and the content of SRAM even when power supply is off. This device is managed by user with proper configuration commands so it is possible to set and get current data and time and decide if they must be saved in file on memory card. This information tell to user a temporal indication respect of saved data, so the system that uses data can check elapsed time, can obtain the time of some events, calculate production data in a period, to make graphic on base time, etc... The user can connect or not this battery to back up circuitry, by acting on dedicated jumper J2, as described in figure 18. The card is supplied with the jumper connected to preserve the clock counting and the SRAM content in each operating condition. Whenever the **S-LOG** is not used for a long time, or the application doesn't need the back up circuit, it is suggested to prevent the battery discharge by removing the jumper J2. Obviously if the J2 connection must be changed, then the card must be extracted from the container making a pressure on the hooks. When printed circuit is extracted from container, the J2 location can be easily found by using figure 4.

**SIGNALLING LED**

On **S-LOG** is present the bicolour LED DL1 that signal some operative conditions, as is described in the following table:

<table>
<thead>
<tr>
<th>STATUS</th>
<th>CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch off</td>
<td><strong>S-LOG</strong> not powered or out of order.</td>
</tr>
<tr>
<td>red, green, yellow alternating in about each half second</td>
<td><strong>S-LOG</strong> working but without memory card.</td>
</tr>
<tr>
<td>green</td>
<td><strong>S-LOG</strong> is working with memory card inserted.</td>
</tr>
<tr>
<td>red</td>
<td><strong>S-LOG</strong> is saving received data, on memory card.</td>
</tr>
<tr>
<td>yellow</td>
<td><strong>S-LOG</strong> is working and is performing a configuration command.</td>
</tr>
<tr>
<td>blinking yellow</td>
<td><strong>S-LOG</strong> is working but with errors.</td>
</tr>
</tbody>
</table>

**Figure 19: Signalling LED**

As shown in figure 4 the LED is easily visible in the insertion side if memory card; for more information on signalled conditions, please see SOFTWARE DESCRIPTION.
ASYNCHRONOUS SERIAL LINE CONFIGURATION

Serial communication line of S-LOG can be in three kind: asynchronous RS 232, asynchronous TTL or synchronous I2C BUS. The choice between asynchronous and synchronous line, of physical and logic communication protocol is via software by proper configuration modality (see bearing the same name paragraph).

When for configuration is chosen asynchronous serial, selection of electric protocol (RS 232 or TTL) is via hardware and reach a proper jumper configuration, as described on figure 18 table. Following there are two figure that show jumper configuration involved in this selection, to make realization easy:

**Figure 20: Asynchronous serial RS 232 jumper configuration**

**Figure 21: Asynchronous serial TTL jumper configuration**

Used communication signals in both configurations always are that connect to 1, 3, 5, 7 and 2 pin of CN1, as described in figures 10÷13.

NOTE. Two RS 232 and TTL protocols are electrically not compatible and user, before connect asynchronous serial of S-LOG to external system, has to carefully verify current serial interface on the system and configures jumpers properly. The RS 232 line connection to S-LOG in TTL configured can also cause the S-LOG damaging.
POWER SUPPLY

S-LOG terminal is provided with a power supply section that solves in a efficient and comfortable way the problem to supply the board, in any situation. When S-LOG is ordered in default version the power supply section is partially assembled and accepts only one regulated voltage as +5 Vdc. For alternate current it can ordered with a proper power supply switching that accepts a wide range voltage.

Here follow the voltages required from S-LOG according to card configuration together with the relative right connection:

Default version: Here the J4 jumper has to be in position 2-3. This configuration is not provided of any power supply section, so a +5 Vdc ± 5% stabilized supply voltage must be provided by an external source, at pin 2 (GND) and pin 8 (+5 Vdc) of CN1 connector. Polarity must be respected also in this case. This allows to provide energy to the terminal through laboratory power supply, other cards, etc. This condition is comfortable because with only one cable it can connect the power supply and the communication line, always on CN1.

.SW Version This configuration includes a switching power supply that requires 10÷38 Vdc or 8÷24 Vac provided through CN1 (polarity must be respected in case of DC supply). This allows to supply the terminal using standard industrial and commercial power sources like transformers, batteries, solar cells, etc. A comfortable and cheap solution for default version power supply can be the EL12 product that can be directly connected to the terminal starting from mains.

Please remind that on board switching section is provided with single diode rectifier, so in case of DC supply, all ground signals of the terminal (GND) are at the same potential. When a single AC source is used to supply different units (both some S-LOG or other cards provided of supply section with single diode rectifier), please ensure that the two phases of AC voltage must be connected at the same input pins of power supply connector. Whenever this rule is not satisfied dangerous malfunctions or damages can rise up on all the connected devices. For example, if we call Phase1 and Phase2 the two signals of the AC voltage, then Phase1 must be always connected to positive inputs (Vac, +Vdc pow) and Phase2 must be connected to negative input (Vac, GND). Complete information and details can be found on paragraph CN1 - POWER SUPPLY CONNECTOR.

With J4 jumper uses can decides to connect the regulated voltage to+5 Vdc, generated from switching, to CN1 connector: if J4 is in position 2-3 that voltage is connected and can used for supply external components; if J4 is in 1-2 the pin 8 of CN1 is not connected and generated voltage is used only by S-LOG.

The complete version of power supply section described has to be properly ordered using the S-LOG.SW indication.

Selection of power supply section must be performed during the order phase, in fact it involves a different hardware configuration that must be made by grifo® technicians.
The S-LOG is always provided with a TransZorb™ protection circuit to avoid damages from incorrect voltages and/or break down of power supply section. It is also provided with a distributed filtering circuitry that saves the terminal from disturbs or noise from the field, improving the overall system performances.

For further information please refer to paragraph ELECTRIC FEATURES.

**Figure 22: Power supply EL 12**
SOFTWARE DESCRIPTION

The S-LOG module, as we already told, is a complete data logger which saves on a memory card everything which has been received from one of its serial communication lines. In order to get usable this things to each external system, and in order to be able to satisfy the huge number of applications requests, it is also available a configuration modality which allows to set the communication protocol and some S-LOG autonomous functions. In general the autonomous functions add data to the memory card in order to make more complete those which has been received by the external system; those functions are very interesting since they increase the S-LOG sectors use.

In this chapter are described the main S-LOG functional features while in the next one you can find the detailed documentation of the available configuration commands: these last ones allow the users to satisfy every market request.

OPERATIVE SELECTION MODE

The S-LOG has two operative modes which are totally independent and separated, selected by the proper jumper J1.

J1 in 1-2 position -> selects the configuration mode.
J1 in 2-3 position -> selects the reception mode and data saving.

Both modes are described by the same following paragraphs, while the information related to their selection can be found below.

a) The J1 jumper is always acquired by the S-LOG and for this reason the passage between the two modes is always allowed. The only exception is during an I2C BUS synchronous communication in which the file is acquired again only when the same communication is brought to an end, that is the STOP sequence.

b) Because of a passage from the reception mode and the saving data to the configuration one, the communication line and its physical protocol are those which have been used before. In this way you can use the configuration commands related to the status and errors, in order to be able to check some malfunctionings, without changing anything on the external system.

The same system which transmits the data to save can communicate with the S-LOG during the configuration mode to verify the status.

c) The communication line and the default physical protocol are selected if the configuration mode is already selected when the S-LOG is on. This system makes you sure to check and/or reset the module, even when its setting is not well known. The default selections are those which are related in figure B1, that is the 19200 Baud asynchronous communication, 8 bit, none parity, 1 Stop and none handshake.

d) Both the modes can be managed from the external system, linked through all the S-LOG serial communication lines. The used line and its physical protocol are established by the conditions where the modality has been chosen like you can read in the previous points.

e) Because of a passage from the configuration mode to the reception and saving data ones, the parameters which are potentially configurated again can’t be used. In other words the work parameters are set with those which are configured only through an ignition, in reception and saving data mode ( for further details see figure 23 ).

F) In order to be able to operate on the J1, you have to get the board out from its container by making some pressure on the closing hooks under the container.

This choice is justified since the reception and saving data mode is normally selected.
while the configuration mode is selected only occasionally (before the installation or in first try-out). The user who makes the entire system will have to open the container just one time, select the configuration mode to prepare and check the S-LOG with, reput the jumper in 2-3 position and reclose the container for good. By setting J1, the accidental wrong selection (which would cause data losing) are avoid.

**Figure 23: Work Parameters Definition**

- **Power on**
  - Gets configuration parameter previously set and store in S-LOG

- **Yes**
  - Valid configuration parameters

- **No**
  - Sets work parameters for storage, with default parameters:
    - Saving file = SLOG.DAT
    - Duration file = None
    - Insertion action = None
    - Data group time = Disabled
    - Start file label = None
    - Data group prefix = None
    - Data group suffix = None

- **Jumper J1**
  - Position 1-2 = configuration
  - Position 2-3 = reception and storage

- **Sets work parameters for storage, with configuration parameters:**
  - Saving file = Path parameter
  - Duration file = Duration parameter
  - Insertion action = Action parameter
  - Data group time = Time parameter
  - Start file label = Label parameter
  - Data group prefix = Prefix parameter
  - Data group suffix = Suffix parameter

- **Gets work parameters for communication, with configuration parameters:**
  - Communication line = Com.line parameter
  - Baud rate = Baud parameter
  - Bit per character = 8
  - Parity = Parity parameter
  - Stop bit = Stop parameter
  - Handshake = Handshake parameter
  - Slave address = Sla.add parameter
  - Timeout com. = Tout parameter

- **Sets work parameters for communication, with default parameters:**
  - Communication line = asynchronous (RS 232, TTL)
  - Baud rate = 19200
  - Bit per character = 8
  - Parity = none
  - Stop bit = 1
  - Handshake = none
  - Slave address = 128 (not used)
  - Timeout com. = 250 * 20 msec=5 seconds

- **Initialize synchronous serial line with physical protocol (Baud rate, 8 Bit per chr, Parity, Stop bit, Handshake) defined at power on in work parameter**

- **Initialize synchronous I2C BUS line in slave mode with slave address defined at power on in work parameter**

- **S-LOG menagement**
WORK PARAMETERS

The S-LOG working in any its operative modality depends on a series of work parameters that define both communication aspects and data storage on memory card.

Current parameters are:

**Work parameters for communication:**
- **com.line**: Serial line used for communication with external system.
- **baud**: Serial Baud rate used for asynchronous serial communication in RS 232, TTL with external system.
- **parity**: Parity check used for asynchronous serial communication in RS 232, TTL with external system.
- **stop**: Stop bit used for asynchronous communication serial in RS 232, TTL with external system.
- **handshake**: Handshake used for asynchronous serial communication in RS 232, TTL with external system.
- **sla.add**: Recognizing S-LOG address used for synchronous serial I2C BUS communication with external system.
- **tout**: Max. waiting time during communication with external system.

**Work parameters for storage:**
- **path**: File name for received data, with eventual folders.
- **duration**: Storage duration of several generated files from S-LOG on memory card.
- **action**: Performed operation of S-LOG after memory card insertion.
- **time**: Recognizing time of a group of received data from S-LOG.
- **label**: Label saved by S-LOG automatically at the begin of file on memory card.
- **prefix**: Prefix saved by S-LOG automatically at the begin of a data group on memory card.
- **suffix**: Suffix saved by S-LOG automatically at the end of a data group on memory card.

A better description of work parameters is available in following paragraph and figures and in COMMANDS chapter; In that chapter are shew all available values as parameter and their default value.

Work parameters are very important and for their storage it was choosen a EEPROM to be sure of good performance and saved data preservation, naturally without power supply too. When saved values in EEPROM are not valid, work parameters will be set as default values as figure 23 and B1.
**CONFIGURATION MODE**

Configuration modality is selected by position of J1 in 1-2 as described in OPERATIVE MODE SELECTION paragraph and user can draw and set all work condition of **S-LOG**. Configuration modality is composed by a bidirectional serial between **S-LOG** and a external system. This one, with a proper electric, physical and logical protocol, can provides a command series and...
it can:
- get current S-LOG status, with eventual errors;
- reset active errors;
- get and set date and time of RTC;
- get and set work parameters for storage;
- get and set work parameters for communication.

Configuration commands are described in detail in the following chapter or summarized in tables in 'APPENDIX A.

In this paragraph are described 3 communication protocols (electric, physical, logic) used in configuration modality.

**ELECTRIC CONFIGURATION PROTOCOL**

Serial communication with S-LOG in configuration mode, can be done with all electric protocols available: asynchronous TTL, asynchronous RS 232 and synchronous I2C BUS, each ones available on CN1.

Used communication line depends on several factors as described in figures 23 and 24. When asynchronous serial line is chosen, the selection between RS 232 and TTL is via hardware as described in ASYNCHRONOUS SERIAL LINE CONFIGURATION paragraph.

**PHYSICAL CONFIGURATION PROTOCOL**

Serial communication with S-LOG in configuration mode, can be done with all physical protocols, that are correspondent with possible combination of work parameters involved (baud, parity, stop, handshake, sla.add). Available values for physical protocol are listed in GENERAL FEATURES paragraph and in command paragraph.

The definition of physical protocol used by S-LOG depends on work parameters configuration and on other factors described in figure 23.

**LOGIC CONFIGURATION PROTOCOL**

Serial communication with S-LOG in configuration mode, can be done with a proper logic protocol in order to simplify and increase speed the preparation and check out of the module by user. Main features of this protocol are summarized in the follow points, while flow charts in figures 25 and 26 show graphically operations performed by two involved communication configuration unit.

a) Logic configuration protocol is based on proper commands that external system transmits to S-LOG; than S-LOG executes them always transmits a response.

b) Both in the command and in the response can be present some added data that meet configuration parameters.

c) All command and response data are always coded in ASCII: moreover in case of numeric values it uses the decimal code, only with more significant numbers. This thing increase the number of communicated characters but in the same time provides an important benefit for S-LOG that can be configured by hand by any user. It is sufficient a simple program of terminal emulation with the selected electric and physical protocols and no other devices, ecc.

d) If the command or the response contain more parameters, these ones are separated by a space.

e) In logic configuration protocol a method for recognize the end of data of communication is used. This method meet with an end character adding (CR in command and CRLF> in
response) when is used the asynchronous communication or with transmission of the numbers of response bytewhen is used the synchronous communication. This is because a lot of communicated data have a variable length (see path and file name, strings to save on file, suffix and prefix, etc.) and that length is not known by reception unit.

f) During configuration are menaged potential communication handshakes and timeout, as described in paragraphs bearing the same name.

g) During configuration the LED of S-LOG is yellow.

**Figure 25: Asynchronous Communication RS 232, TTL for Configuration**
h) The first character of response of configuration commands always meet the executed command flag. This is a single character and can assumes values:

0 (ASCII code 48=30H) -> command not executed
1 (ASCII code 49=31H) -> command executed correctly

In the first case the 'S-LOG' don’t give back any other response data and external system can recognize communication and/or configuration problems, through comando STATUS REQUEST command. Naturally in the second case all data response are sent to S-LOG.
RECEPTION AND SAVING DATA MODE

Reception and saving data mode is selected by J1 in 2-3 position as described in OPERATIVE SELECTION MODE paragraph and allows to user to save on memory card all have been received from selected serial line. Reception and saving data mode consist of a monodirectional serial communication between an external system and S-LOG. The first, with a proper electric, physical and logic protocol, can provide data to save and can actives some saving functions previously configured. This modality is normally set and allow to use S-LOG for data saving system development both in civil sector and industrial sector, like in figure below.

Data are saved on memory card with format set from external system and can be easily copied, registered, worked, printed, etc. by a normal PC, provided of an interface of memory cards. To simplify memory cards management, the S-LOG can be set for save some added information to received data, or divide these data in different files with configurable length, as described DATA SAVING paragraph.

In following three paragraphs are described communication protocols (electric, physical, logic) used by this mode.

**Figure 27: Reception and saving data mode**
ELECTRIC PROTOCOL FOR RECEPTION AND SAVING DATA

Serial communication with S-LOG in reception and saving data mode, can be done with all electric protocols available: asynchronous TTL, asynchronous RS 232 and synchronous I2C BUS. Used communication line depends on several factors as described in figures 23 and 24.

In details:

Asynchronous serial line RS 232, TTL:
Asynchronous communication uses signals on CN1 and they can be:
- GND, RX xxxx when external system don't use handshake;
- GND, RX xxxx, RTS xxxx when external system use hardware handshake;
- GND, RX xxxx, TX xxxx when external system use software handshake (XON, XOFF);
- Not listed signals are not managed and can be connected or not.
This communication is good for point to point connections in RS 232 or TTL, as described in ASYNCHRONOUS SERIAL LINE CONFIGURATION paragraph.

Synchronous serial line I2C BUS:
Synchronous communication uses signals on CN1 and they can be:
- GND, SCL, SDA when system not provides or non needs power supply;
- GND, SCL, SDA, +5 Vdc when external system provides or nedds power supply.
This communication is good both for point to point connections and as network.

PHYSICAL PROTOCOL FOR RECEPTION AND SAVING DATA

Serial communication with S-LOG in reception and saving data mode, can be done with all physical protocols, that are correspondent with possible combination of work parameters involved (baud, parity, stop, handshake, sla.add). Available values for physical protocol are listed in GENERAL FEATURES paragraph and in command paragraph.
The definition of physical protocol used by S-LOG depends on work parameters configuration and on other factors described in figure 23.

LOGIC PROTOCOL FOR RECEPTION AND SAVING DATA

Serial communication with S-LOG in reception and saving data mode, can't be done with any logic protocol in order to semplify and increase speed the use of module by user. The absence of logic protocol mean that each received data is saved directly on memory card so is described in SAVING DATA paragraph.
Durante communication are managend only potential handshake and timeout, as described in paragraph bearing the same name.

The Flow charts in figure 28 and 29 show grafically operations performed by two unit involved in communication for saving data.
**EXTERNAL SYSTEM**

Start

Initialize serial line with physical protocol (Baud rate, 8 Bit per chr, Parity, Stop bit, Handshake) set on S-LOG

External unity process management

Input handshake enabled and inactive

YES

NO

Transmit data to store to S-LOG

**S-LOG**

Data reption management with asynchronous serial (RS 232, TTL)

Received Bytes by external unity

YES

NO

Save received bytes in the reception buffer

Reception buffer full

YES

NO

Active output handshake, if enabled

**FIGURE 28: ASYNCHRONOUS COMMUNICATION RS 232, TTL FOR DATA SAVING**

**EXTERNAL SYSTEM**

Start

Initialize I2C BUS line in master mode with physical protocol (Bit rate) from 250 to 460000 Bit per second

External unity process management

**S-LOG**

Reception data management with synchronous serial (I2C BUS)

Build a communication as master transmit:
1) send START sequence;
2) send slave address set on S-LOG writing (with R/W=0);
3) verify ACK reception;
4) send the first byte of the command to execute;
5) verify ACK reception;
6) send the second byte of the command to execute;
7) verify ACK reception;
8) send the last byte of the command to execute;
n+1) verify ACK reception;
n+2) send STOP sequence.

Build a communication as slave receive:
1) recognize START sequence;
2) receive and compare the slave address set on S-LOG writing (with R/W=0);
3) if they meet go on otherwise remove all I2C BUS data until the next STOP sequence.
4) Check the kind of R/W operation, if in writing (R/W=0) goes on to receive otherwise sets sequence error and invalid communication;
5) send ACK;
6) receive and save on reception buffer the first byte to store;
7) send ACK;
8) receive and save on reception buffer the second byte to store;
9) send ACK;
n) receive and save on reception buffer the last byte to store;
n+1) send ACK;
n+2) recognize STOP sequence.

**FIGURE 29: SYNCHRONOUS COMMUNICATION I2C BUS FOR DATA SAVING**
DATA SAVING

As we have told in the previous paragraphs, all the received data in the data reception mode are saved on memory card without any changing: so the saved data format is not established by the S-LOG but by the transmitting system. In other words it is the external system to decide if the saved data are in binary, ASCII, codified, lined up format etc…

Always in this mode beside the serial communication and the received data saving, the S-LOG can also manage some autonomous functions which add some info to the memory card in order to make them more complete and easier to manage. Among these added information we should remember:

- Current time indication, complete of date, time and milliseconds;
- Format and ranging characters ( CR, LF, TAB, SPACE );
- Data identification strings.

The added information can be saved at the beginning or at the end of a group of data by taking advantage of the proper configuration commands which work on the prefix and/or the suffix. As we have illustrated in the paragraphs related to these commands, the suffix and the prefix are strings which show to the S-LOG which information need to be added on the memory card, the order and the format.

The data group coincides instead with a series of characters which are continuously received or without any temporal interruption with a higher duration than any other programmed values. The S-LOG measures the elapsed time from the last received and saved data, and when this time goes over the TIME working parameter, the data group is considered ended.

At the same time if the suffix says to save some adding information, these last ones are written on the memory card. In the same way, when the next data group begins ( when the S-LOG receives the first data after the described interval ), if the prefix indicates to save the adding information, those are written on the memory card, before the received data. With this method ( also illustrated in the figure 24 diagram ) the prefix information are saved before the data group and the suffix ones are saved after.

All saved data by S-LOG just in one file or in more files that are written on memory card the defined features from work parameter path and duration. In detail:

- If is not set the duration the S-LOG write all data just in one saving file, that has the name and the folder defined by path.

For example when the S-LOG is configured with the following two work parameters:

path = OVEN\OUTPUT\PRNOVEN.TXT
duration = N (none)

Than in file OVEN\OUTPUT\PRNOVEN.TXT all data will be written.

- If is set the duration (≠N) the S-LOG write its data in different saving files, that have a name with yyyyymmdd.hh format, in the folder defined from path. The elapsed time for working file coincides with the value of duration parameter and can be like one hour, one day, one month or one year. The name of used file meet the beginning time of choosen duration so as follow:

<table>
<thead>
<tr>
<th>duration</th>
<th>file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>hour</td>
<td>yyyyymmdd.hh</td>
</tr>
<tr>
<td>day</td>
<td>yyyyymmdd.00</td>
</tr>
<tr>
<td>month</td>
<td>yyyyymm01.00</td>
</tr>
<tr>
<td>year</td>
<td>yyyy0101.00</td>
</tr>
</tbody>
</table>

The selected file name format allows to makes order when they are managed by PC; listing the files in alphabetical order, they will be ordered in temporal order too.

When in the path parameter is specified the file name too, this one will be discard and replaced by the described name.
If for example the **S-LOG** is configured with this two following work parameters:

- **path** = OVEN1\OUTPUT\PRNOVEN.TXT
- **duration** = D (day)

and its clock is 21/07/2008 time 14:30, then:

- in file OVEN1\OUTPUT\20080721.00 will write the data up to the end of the day,
- in file OVEN1\OUTPUT\20080722.00 will write the data of next day,
- in file OVEN1\OUTPUT\20080723.00 will write the data of the third day,

etc.

By taking advantage of the proper configuration commands, the user can establish the name of the file and its potential path (folders) where the file is found, as it is described in the SET FILE and GET FILE paragraphs. This data file is first prepared after the switching on or the memory card inserting and then extended every time there is data to save. Every new data is always added at the end of the file which gets bigger. During the writing of every data the **S-LOG** also verifies the right developing of the operation and communicates every potential error with the yellow LED blinking.

When file duration elapses and data saving pass to next file, if the reception buffer contains data, these are saven on the new file, not the old one.

It is important to remember that the data saving can be inhibited by the protection selector during the writing of the inserted memory card: in fact, if this last one is active the **S-LOG** can’t perform any writing on the card. Since the main function of the **S-LOG** is the data saving, if the card protection is on, an error is reported (yellow LED blinking) in correspondence to the first attempt of data saving. During writing operation of data the **S-LOG** activates its LED red to indicates the operation in progress. It is important that the user does not remove the memory card when the LED is red, because he can lose data and can damage the memory card removed. Please see the following parameter OPERATION ON MEMORY CARD for more information.

**HANDSHAKE COMMUNICATION**

Whether in configuration and data reception the **S-LOG** makes some characterized by a variable execution time. Some of these operations, like the work parameters saving, physical writing on received data files, the formatting of the memory card, and many others, take a long execution time with the possible loss of transmitted data from the external system.

In order to avoid this loss, the **S-LOG** foresees several kind of handshake communication. The handshakes manage only the asynchronous communication, where the two systems don’t have any temporal correlation, they can have the execution speed notably different and have some variable work amounts.

The foreseen technicals are:

- **Handshake HW**: it is based on a couple of hardware signals which join the communication ones, always available on the CN1 connector.

  - **RTSxxxx** **S-LOG** -> **CTS** external system
    - When the **S-LOG** is ready to get data, it activates its RTS and its external system subordinates the transmission to the status of its active CTS.
  - **RTS** external system <- **CTS xxxx** **S-LOG**
    - When the external system is ready to get data, its activates its RTS and the **S-LOG** subordinates the transmission to the status of its active CTS. This handshake is not managed in data saving reception mode, since in this case the **S-LOG** never transmits data to the external system.
- Handshake SW: it is based on the characters XON (code 17=11H) and XOFF (code 19=13H) which
  are added to the communication.

  **S-LOG** -> external system

  When the **S-LOG** can’t get any data it transmits XOFF, followed by XON when it is ready
  again. The external system subordinates its transmission to the last XON, XOFF received
  status.

  External system -> **S-LOG**

  When the external system can’t get any data, it transmits XOFF followed by XON when it
  is ready again. The external system subordinated its transmission to the last XON, XOFF
  received status. This handshake is not managed in data saving reception mode, since in this
  case the **S-LOG** never transmits data to the external system.

- SW Repeated Handshakes: the condition is similar to the software handshakes that we have
  previously described, but it is transmitted every second by the **S-LOG**.

The **S-LOG** can’t get any data when its reception buffer is 90% full, but it can get data again when
the same percentage is lower than the 20%.

It is important to remember that the described handshakes are managed only if one of the three
techniques has been set through the proper configuration command; otherwise other handshakes are
not foreseen.

**OPERATIONS ON MEMORY CARD**

The **S-LOG** can write inserted memory card only if this one is ready for data saving; the preparation
is performed by **S-LOG** at every switch on and at every card insertion.

Both insertion **action** and **label** are two work parameters that can be configured by user through
proper management commands. In particular with insertion **action** user can decides to keep data
saved before in memory card at the insertion increasing the current file or delete them with creation
of a new data file.

In this case **S-LOG** can write a label too, at the begin of new file; that label meet added information
automatically saved, that can includes real time information, formatting character and data identification
strings, etc.

Note that data file that is verified and than open or created meet the one described in SAVING DATA
paragraph. In case of a length configuration, the file in object is the actual one, with name set by
current date and time.

At the end of operations that follow the memory card insertion, if all is right, green LED is active,
and viceversa in case of errors yellow LED is blinking.

Memory card can be inserted everytimes you want and in any operative conditions. Viceversa card
with saved data has to be removed only when there are no operations in progress, in fact when DL1
is red, the card can’t be removed. If this rule is not respect in saving data will be definitely lost and
memory card can be damaged. Once the card is removed, if there are no errors, LED is blinking by
turns with red, green and yellow colours. If there are errors LED is yallow blinking.

If the memory card is valid (see CN2 paragraph -MEMORY CARD CONNECTOR) are performed
operations described in following flow chart:
Both insertion action and label are two work parameters that can be configured by the user through proper management commands. In particular, with insertion action, the user can decide to keep data saved before in memory card at the insertion increasing the current file or delete them with creation of a new data file.

In this case S-LOG can write a label too, at the begin of new file; that label meet added information automatically saved, that can includes real-time information, formatting character and data identification strings, etc.

Note that data file that is verified and than open or created meet the one described in SAVING DATA paragraph. In case of a length configuration, the file in object is the actual one, with name set by current date and time.

At the end of operations that follow the memory card insertion, if all is right, green LED is active, and vice versa in case of errors yellow LED is blinking.

Memory card can be inserted everytimes you want and in any operative conditions. Vice versa, card with saved data has to be removed only when there are no operations in progress, in fact when DL1 is red, the card can't be removed. If this rule is not respect in saving data will be definitely lost and memory card can be damaged. Once the card is removed, if there are no errors, LED is blinking by turns with red, green and yellow colours. If there are errors LED is yellow blinking.
DEMO PROGRAMS

For **S-LOG** module are available demo programs both in source level and executable level; these programs can be used without any changes for a first check out of the product and then modified, or used in pieces, to satisfy user's requirements. These programs are available for more high level programming languages (C, PASCAL, BASIC, etc.) both for PC and microprocessor **GPC** or **GMM** board of **grifo®** family. User can examine their comments and decide himself if test them or not.

All demo programs can be directly used or modified by user with no any authorization or added costs. In case of particular requirements or use combination you can requires specific demo programs in according with **grifo®**.

Moreover is available a PC program, called **Conf_SLOG** that allow to manage configuration modality of the board, with a normal PC connected to asynchronous serial line RS 232. The purpose of this program is easy and intuitive, but for who need help, we recommend to see HOW TO START chapter, where there are an example use.

ERRORS

The **S-LOG** check a several of condition during his work, both in configuration mode and in reception and saving data. Several error condition are included, they meet defects, malfunctions and/or not foreseen status.

The complete list of recognized errors is shown in GET STATUS paragraph that describes the command able to get the total status of **S-LOG**.

When an error is recognized the LED of **S-LOG** become yellow blinking, in order to signalling the wrong condition; this blinking length continues as long as errors are reset with proper command.

In detail to correctly manages errors, user has to perform these following steps:

a) Use the **S-LOG** in reception and saving data mode.

b) When an error occurs, the LED becomes blinking yellow.

c) Modify the external unit working in order to able manage the configuration mode, or disconnect and replace it with another unit with the same functionality (i.e. a PC).

d) Shift the J1 jumper in 1-2 in order to select configuration mode on **S-LOG**.

e) With external unit provide the GET STATUS command and examine active errors in his response.

f) Once seen errors, try to fix them (for example unlock the write protection on memory card, replace the full card, increase communication timeout, etc.).

g) Provide RESET ERRORS command that disables all error status and stop the blinking of the LED and become fixed in red if memory card is inserted or red,green, yellow by turns if the card is not inserted.
h) Shift J1 jumper in 2-3 in order to select the reception and saving data mode on S-LOG.

i) Modify the external unit work in order to restart to transmit data to save; if, at point c the unit was replaced, re-connect the original one.

Keep in mind that error are rare phenomena when application is check out and installed. The operations listed above normally have to be performed only at beginning phase preparation of S-LOG.

**NOTE FOR SYNCHRONOUS COMMUNICATION I2C BUS**

When is using synchronous communication I2C BUS in both operative modes of S-LOG the rules to follow are:

a) The systems that communicates with S-LOG in this modality must operate as master, either in transmit and receive mode, following the rules of I2C BUS standard protocol detailed described in the document "THE I2C-BUS SPECIFICATIONS", from PHILIPS semiconductors.

b) The S-LOG always works as Slave, in particular as Slave receive when reception and saving data mode is selected and as Slave receive+transmit when configuration mode is selected. The external unit has to works as Master transmit to send data da save or configuration commands and as Master receive to receive the configuration commands response.

c) Local network communicartion is possible both with unit of the same type and different type as shown in figures 16 and 31.

d) I2C BUS communication has a synchronism between the two communication systems, as shown in figures 26 and 29. So is not necessary none handshake used only in asynchronous communication.

e) Each communication interest only the used slave address of S-LOG that meet the express one in work parameter sla.add. In case of a I2C BUS communication network, each S-LOG, have to be set with a different slave, different address and different from other ones of I2C BUS device potential present in the same network.

f) In order to semplicity the management, the first data give back from S-LOG, always meet the number of character of response, that is the number of data that the external system has to receive. The last one has to finish communication with a STOP sequence, after all data reception.

g) In single communication between external unit and S-LOG can be trasferred several data to save, taking care to doen't fill upthe reception buffer.

h) The slave address defined in local setup is 7 bit but is set to 8 bit with less significant bit (R/W) set to 0: it is possible to use the 128 different values even, in range 00H+FEH.
i) In case of I2C BUS network connection perform configuration described in paragraph CN1 - SERIAL COMUNICATION LINE AND SWITCHING CONNECTOR making sure that the line is correctly terminated (see figure 16).

j) The S-LOG doesn't support I2C BUS protocol extension (addressing 10 bits, fast communication, etc.) and reserved slave address; these thing has not to be used by user in the external unit.

TIMEOUT IN COMMUNICATION

To avoid long waiting during the communication there is the \texttt{tout} work parameter that ensures that S-LOG is always operative. The \texttt{tout} value can be define and acquire by proper configuration command and it is express in twenties of milliseconds, as described in referred paragraph.

In correspondence to any configuration command in communication, the S-LOG activates a counter of elapsed time and during the waiting it verify if the counter got the timeout parameter, if enabled. In case it is disabled it continue to waiting, viceversa step to following operation and set the expired timeout error. The timeout is check for each communication event, so if S-LOG has to wait several events, it will be operative again after a complessive time equal to the product of two ones.

When the S-LOG communication is managed manually by an operator we recommend to disable timeout or to set it to a very high time.
Figure 31: Network of different I2C BUS devices

- GPC® xxx, GMM xxx = MASTER system with I2C BUS interface
- SAA 1064 = I2C BUS driver for 4 displays with 7 segments
- PCF 8574 = I2C BUS interface for 8 TTL digital I/O
- RELAYS
- TRANS.
- OPTO
- GMB HRxxx = programmable controllers with I2C BUS interface
- DS 1621 = Digital thermometer and thermostat (-55° to +125° C) with I2C BUS interface
- QTP 03, QTP 12, QTP 16Big = terminal panels with I2C BUS interface
- Any devices with I2C BUS interfaces, in SLAVE mode
- PFC 8591 = D/A converter (1 ch, 8 Bits, 0±5V) and A/D converter (4 chs, 8 Bits, 0±5V) with I2C BUS interface
- S-LOG = data logger on SD and MMC memory cards with I2C BUS interface
- MCC = I2C BUS interfaces for PC
CONFIGURATION MODE COMMANDS

In this chapter are described all available commands in configuration mode of S-LOG with relative input and output parameters. Commands are divided in subgroups in according with their function and for each one is made a triple description: mnemonic one, through ASCII characters and the numeric one express in decimal and exadecimal format.

All commands and associated parameters are coded with rules already described in LOGIC CONFIGURATION PROTOCOL paragraph and summarized below:

a) The external system transmits the command and the S-LOG give back a response.

b) Potential command and response parameters always are coded in ASCII.

c) If the parameter is numeric it can be used the decimal code, with only significant numbers.

d) If the command or the response include more parameters, these are separated by space.

e) In the follow description is not give back the performed command flag that meet the first data response, with the following code:

0 (48=30H) -> not performed command: the rest of response don't give back;  
1 (49=31H) -> command performed correctly: the rest of response give back.

f) In the following description there aren't data to recognize the end of communication, that is:

CR (13=0DH) -> at the end of second command with asynchronous communication;

CRLF> -> at the end of response with asynchronous communication. This thing do the prompt too in case of manually provided commands from a terminal emulator;

<bytes number -> at the begin of response with synchronous communication, before of response> the performed command flag.

g) In the following descriptions there aren't handshaking data of communication (XON,XOFF) that can exchanged when asynchronous communication is selected and software handshake are active.

For a quickly search of available commands we recommend to see summarizing tables in APPENDIX A. In figure B1 there are default settings of all configuration commands described in this chapter.

Keep in mind that command are performed only when configuration mode is selected; if they are transmitted when is selected the reception and saving data mode, the effect is that data will be saved directly on memory card if present. More information on operative mode selection and its communication features, are shown in the paragraph of previous chapter.

During the management of a command (reception, execution and response transmission) the S-LOG keeps the LED fixed in yellow.
GENERAL COMMANDS

Following we are, relative commands to current status of S-LOG, its potential errors, real time clock and version.

GET VERSIONS

Command:  v  
Response:  hw.ver  fw.ver
Command Dec:  118  
Response Dec:  hw.ver  32  fw.ver
Command Hex:  76  
Response Hex:  hw.ver  20  fw.ver

The command give back hardware and firmware versions of S-LOG like ASCII strings. The two strings have the following constant format:

- hw.ver -> 6 numeric characters;
- fw.ver -> 3 numeric characters in unit.tens format.

For examples as shown in HARDWARE, FIRMWARE VERSION beginning chapter are give back the following characters:

110907 4.2
49 49 48 57 48 55 32 46 50
31 31 30 39 30 37 20 34 2E 32
Hex

RESET ERRORS

Command:  Z  
Response:
Command Dec:  90  
Response Dec:
Command Hex:  5A  
Response Hex:

This command reset all error managed from S-LOG. The reset can be verified with ACQUIRE STATUS command, there is the complete list of errors too.

The command don't give back none response parameter.

GET STATUS

Command:  z  
Response:  general.status disck.status
Command Dec:  122  
Response Dec:  general.status 32 disck.status
Command Hex:  7A  
Response Hex:  general.status 20 disck.status

This command give back the S-LOG status as two ASCII strings. These strings has numeric format with variable length of status and current errors with the following relation to bit:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>dec. weight</th>
<th>Hex Weight</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>general status.00</td>
<td>00001</td>
<td>0001</td>
<td>Communication overflow error.</td>
</tr>
<tr>
<td>general status.01</td>
<td>00002</td>
<td>0002</td>
<td>Parity error in asynchronous line.</td>
</tr>
<tr>
<td>general status.02</td>
<td>00004</td>
<td>0004</td>
<td>Status error in synchronous line.</td>
</tr>
<tr>
<td>general status.03</td>
<td>00008</td>
<td>0008</td>
<td>Invalid communication sequence error.</td>
</tr>
<tr>
<td>general status.04</td>
<td>00016</td>
<td>0010</td>
<td>RTC access error.</td>
</tr>
<tr>
<td>general status.05</td>
<td>00032</td>
<td>0020</td>
<td>Invalid configuration error (set work parameter with default values).</td>
</tr>
<tr>
<td>general status.06</td>
<td>00064</td>
<td>0040</td>
<td>Not recognized command error.</td>
</tr>
<tr>
<td>general status.07</td>
<td>00128</td>
<td>0080</td>
<td>Command parameters error.</td>
</tr>
<tr>
<td>general status.08</td>
<td>00256</td>
<td>0100</td>
<td>Memory card present.</td>
</tr>
<tr>
<td>general status.09</td>
<td>00512</td>
<td>0200</td>
<td>Memory card write protect (WP selector enabled).</td>
</tr>
<tr>
<td>general status.10</td>
<td>01024</td>
<td>0400</td>
<td>Configuration mode active (J1 in 1-2).</td>
</tr>
<tr>
<td>general status.11</td>
<td>02048</td>
<td>0800</td>
<td>Elapsed Timeout of communication.</td>
</tr>
<tr>
<td>general status.12</td>
<td>04096</td>
<td>1000</td>
<td>Not used.</td>
</tr>
<tr>
<td>general status.13</td>
<td>08192</td>
<td>2000</td>
<td>Not used.</td>
</tr>
<tr>
<td>general status.14</td>
<td>16384</td>
<td>4000</td>
<td>Not used.</td>
</tr>
<tr>
<td>general status.15</td>
<td>32768</td>
<td>8000</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.00</td>
<td>00001</td>
<td>0001</td>
<td>Disk initialization error.</td>
</tr>
<tr>
<td>disk status.01</td>
<td>00002</td>
<td>0002</td>
<td>Disk formatting error.</td>
</tr>
<tr>
<td>disk status.02</td>
<td>00004</td>
<td>0004</td>
<td>Creating folder error in data path file.</td>
</tr>
<tr>
<td>disk status.03</td>
<td>00008</td>
<td>0008</td>
<td>Change folder error of data path file.</td>
</tr>
<tr>
<td>disk status.04</td>
<td>00016</td>
<td>0010</td>
<td>Opening data file error.</td>
</tr>
<tr>
<td>disk status.05</td>
<td>00032</td>
<td>0020</td>
<td>Writing data file error.</td>
</tr>
<tr>
<td>disk status.06</td>
<td>00064</td>
<td>0040</td>
<td>Flushing buffer error on data file.</td>
</tr>
<tr>
<td>disk status.07</td>
<td>00128</td>
<td>0080</td>
<td>Writing on protect disk error.</td>
</tr>
<tr>
<td>disk status.08</td>
<td>00256</td>
<td>0100</td>
<td>Reception data and following saving error with not present disk.</td>
</tr>
<tr>
<td>disk status.09</td>
<td>00512</td>
<td>0200</td>
<td>Closing data file error.</td>
</tr>
<tr>
<td>disk status.10</td>
<td>01024</td>
<td>0400</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.11</td>
<td>02048</td>
<td>0800</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.12</td>
<td>04196</td>
<td>1000</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.13</td>
<td>08192</td>
<td>2000</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.14</td>
<td>16384</td>
<td>4000</td>
<td>Not used.</td>
</tr>
<tr>
<td>disk status.15</td>
<td>32768</td>
<td>8000</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

Keep in mind that error bit are set by **S-LOG**, during its work and reset only by its **RESET ERRORS** proper command; In this mode every error is memorized and kept as long as the user reset it, after get it, as described in **ERRORS** paragraph. In previous description was used the simply disk term to indicate used memory card, for PC analogies that manages it like a disk.

If the **S-LOG** is in configuration mode, with a memory card protected inserted with already data received but non saved because of the protection, when receives this command gives back the following response:

```
1792 128
49 55 57 50 32 49 50 56
31 37 39 32 20 31 32 38
```

or

```
Hex
```
SET REAL TIME CLOCK

Command: \texttt{T \textit{gg/mm/aaaa 00:mm:ss}}

Command Dec: 84 32 dd 47 mm 47 yyyy 32 hh 58 mm 58 ss

Response Dec: dd 47 mm 47 yyyy hh 58 mm 58 ss

Response Hex: 54 20 dd 2F mm 2F yyyy 20 hh 3A mm 3A ss

The command set the real time clock of \texttt{S-LOG}, with contained data in previous parameters.

Following are reported in detail the means of 6 values, divided in two parameters, with the relative range of validity:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd</td>
<td>1+31</td>
<td>-&gt; Day</td>
</tr>
<tr>
<td>mm</td>
<td>1+12</td>
<td>-&gt; Month</td>
</tr>
<tr>
<td>yyyy</td>
<td>0+9999</td>
<td>-&gt; Year</td>
</tr>
<tr>
<td>hh</td>
<td>0+23</td>
<td>-&gt; Hours</td>
</tr>
<tr>
<td>mm</td>
<td>0+59</td>
<td>-&gt; Minutes</td>
</tr>
<tr>
<td>ss</td>
<td>0+59</td>
<td>-&gt; Seconds</td>
</tr>
</tbody>
</table>

\textbf{FIGURE 32: REAL TIME CLOCK PARAMETERS}

The command is always performed, if a parameter has a value out of range too: the external system have guarantee the validity of values.

In order to guarantee the update of clock, without power supply too, please see BACK UP REAL TIME CLOCK paragraph.

The clock function is to save some adding data on memory card, that meet current date and time. Date and time are used to define the creation and modify time of file data too.

\textbf{Note.} Because of the year with four numbers managed by clock has to be always valid, \textit{it is necessary to provide the set clock command, once every 4 years.}

GET REAL TIME CLOCK

Command: \texttt{t}

Response: \texttt{dd/mm/yyyy hh:mm:ss}

Command Dec: 116

Response Dec: dd 47 mm 47 yyyy hh 58 mm 58 ss

Command Hex: 74

Response Hex: dd 2F mm 2F yyyy hh 3A mm 3A ss

Are reported 6 data, grouped in two parameters, that meet temporal parameters with current date and time, get from the clock of \texttt{S-LOG}.

The means of these parameters are shown in previous figure.
COMMUNICATION COMMANDS

Following there are commands referred to serial communication configuration of S-LOG.

SET COMMUNICATION LINE

Command: \texttt{C com.line} \\
Command Dec: \texttt{67 32 com.line} \\
Command Hex: \texttt{43 20 com.line}

Response: \\
Response Dec: \\
Response Hex:

The command set the serial line used by S-LOG for communication with external system, with passed parameter.

Following there is the description of available values of \texttt{com.line} parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>R (82=52H)</td>
<td>asynchronous communication line(RS 232,TTL)</td>
</tr>
<tr>
<td>I (73=49H)</td>
<td>synchronous communication line (I2C BUS)</td>
</tr>
</tbody>
</table>

\textbf{Figure 34: Communication line parameter municazione}

The command is always performed, if the parameter is has a out fo range value not valid too: the external system has to guarantee the validity.

GET COMMUNICATION LINE

Command: \texttt{c} \\
Command Dec: \texttt{99} \\
Command Hex: \texttt{63}

Response: \texttt{com.line} \\
Response Dec: \texttt{com.line} \\
Response Hex: \texttt{com.line}

A parameter is returned relavive to serial line used by S-LOG for communication with external system.

The description of that parameter is the same of previous figure.
Figure 33: Available Connections Diagram
SET BAUD RATE

Command: \textit{R baud} \quad \textit{Response: baud}

Command Dec: 82 32 baud \quad Response Dec: 82 32 baud

Command Hex: 52 20 baud \quad Response Hex: 52 20 baud

The commands set the speed for serial asynchronous line RS 232, TTL used by \textbf{S-LOG} for the communication with the external system, with the passed parameter.

Following there are the description of available values for \textit{baud} parameter:

\begin{tabular}{|c|c|}
  \hline
  Value & Means  \\
  \hline
  300 & 300 Baud \\
  600 & 600 Baud \\
  1200 & 1200 Baud \\
  2400 & 2400 Baud \\
  4800 & 4800 Baud \\
  9600 & 9600 Baud \\
  19200 & 19200 Baud \\
  38400 & 38400 Baud \\
  57600 & 57600 Baud \\
  115200 & 115200 Baud \\
  \hline
\end{tabular}

\textbf{FIGURE 35: BAUD RATE PARAMETER}

The command is always performed, if the parameter is has a out of range value not valid too: the external system has to guarantee the validity.

GET BAUD RATE

Command: \textit{r} \quad \textit{Response: baud}

Command Dec: 114 \quad Response Dec: 114 baud

Command Hex: 72 \quad Response Hex: 72 baud

A parameter is returned relative to serial line used by \textbf{S-LOG} for communication with external system.

The description of that parameter is the same of previous figure.
SET PARITY

**Command:** P parity  
**Command Dec:** 80 32 parity  
**Command Hex:** 50 20 parity  

The command sets the parity for serial asynchronous line RS 232, TTL used by S-LOG for the communication with the external system, with the passed parameter. Following there are the description of available values for the parity parameter:

<table>
<thead>
<tr>
<th>Values</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (78=4EH)</td>
<td>None parity</td>
</tr>
<tr>
<td>E (69=45H)</td>
<td>Parity even</td>
</tr>
<tr>
<td>O (79=4FH)</td>
<td>Parity odd</td>
</tr>
</tbody>
</table>

**Figure 36: Parity parameter**

The command is always performed, if the parameter is has a out of range value, not valid too: the external system has to guarantee the validity.

GET PARITY

**Command:** p  
**Command Dec:** 112  
**Command Hex:** 70  

A parameter is returned a relative parameter to parity check for asynchronous serial RS 232, TTL used by S-LOG for the communication with external system. The means of that parameter is the same shown in previous figure.

SET STOP BIT

**Command:** S stop  
**Command Dec:** 83 32 stop  
**Command Hex:** 53 20 stop  

The command is always performed, if the parameter is has a out of range value, not valid too: the external system has to guarantee the validity. Following there are the description of available values for the stop parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (49=31H)</td>
<td>1 stop bit</td>
</tr>
<tr>
<td>2 (50=32H)</td>
<td>2 stop bit</td>
</tr>
</tbody>
</table>

**Figure 37: Stop parameter**

The command is always performed, if the parameter is has a out of range value, not valid too: the external system has to guarantee the validity.
GET STOP BIT

Command:  s
Command Dec:  115
Command Hex:  73
Response:  stop
Response Dec:  stop
Response Hex:  stop

A parameter is returned a relative parameter to number of stop bit for asynchronous serial RS 232, TTL used by S-LOG for the communication with external system. The means of that parameter is the same shown in previous figure.

SET HANDSHAKE

Command:  H handshake
Command Dec:  72 32 handshake
Command Hex:  48 20 handshake
Response:  handshake
Response Dec:  handshake
Response Hex:  handshake

The command set the handshake for serial asynchronous line RS 232, TTL used by S-LOG for the communication with the external system, with the passed parameter. Following there are the description of available valueds for handshake parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (78=4EH) -&gt; None handshake</td>
<td></td>
</tr>
<tr>
<td>H (72=48H) -&gt; Handshake hardware (RTS, CTS)</td>
<td></td>
</tr>
<tr>
<td>S (83=53H) -&gt; Handshake software (XON, XOFF)</td>
<td></td>
</tr>
<tr>
<td>R (82=52H) -&gt; Handshake software repeated</td>
<td></td>
</tr>
</tbody>
</table>

Figure 38: Handshake parameter

The command is always performed, if the parameter is has a out fo range value not valid too: the external system has to guarantee the validity. Further information about this command are available in COMMUNICATION HANDSHAKE paragraph.

GET HANDSHAKE

Command:  h
Command Dec:  104
Command Hex:  68
Response:  handshake
Response Dec:  handshake
Response Hex:  handshake

A parameter is returned a relative parameter to handshake for asynchronous serial RS 232, TTL used by S-LOG for the communication with external system. The means of that parameter is the same shown in previous figure.
SET SLAVE ADDRESS

Command: A sla.add
Command Dec: 65 32 sla.add
Command Hex: 41 20 sla.add
Response:
Response Dec:
Response Hex:

The command sets the slave address for serial asynchronous line RS 232, TTL used by S-LOG for the communication with the external system, with the passed parameter. Available values to sla.add parameter are all even numeric values from 0 to 254 (00H to FEH) as described in previous parameters referred to I2C BUS communication. The command is always performed, if the parameter is has a out of range value not valid too: the external system has to guarantee the validity.

GET SLAVE ADDRESS

Command: a
Command Dec: 97
Command Hex: 61
Response: sla.add
Response Dec: sla.add
Response Hex: sla.add

A parameter is returned a relative parameter to slave address for synchronous serial I2C BUS used by S-LOG for the communication with external system. The means of that parameter is the same shown in previous figure.

SET COMMUNICATION TIMEOUT

Command: O tout
Command Dec: 79 32 tout
Command Hex: 4F 20 tout
Response:
Response Dec:
Response Hex:

The command sets the max. time to wait from S-LOG for communication event from the external system, with the passed parameter. Following there are the description of available values for tout parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (00H)</td>
<td>-&gt; No waiting</td>
</tr>
<tr>
<td>1+250 (01+FAH)</td>
<td>-&gt; Waiting 20+5000 milliseconds</td>
</tr>
<tr>
<td>255 (FFH)</td>
<td>-&gt; infinite waiting</td>
</tr>
</tbody>
</table>

**Figure 39: Timeout parameter**

The command is always performed, if the parameter is has a out of range value not valid too: the external system has to guarantee the validity.
The parameter writes the timeout in twenties of milliseconds, with two exceptions: with value 0 (00H) the S-LOG don’t wait the communication event while with 255 value (FFH) wait it in infinite time; in the last case the external system has to guarantee that the event occur to avoid the inactivity of the S-LOG.
Further information about this command are available in COMMUNICATION TIMEOUT paragraph.
GET COMMUNICATION TIMEOUT

Command:  o  
Command Dec:  111
Command Hex:  6F
Response:  tout
Response Dec:  tout
Response Hex:  tout

A parameter is returned a relative parameter to the max. waiting time by S-LOG for a communication event from external system. The means of that parameter is the same shown in previous figure.

SAVING COMMANDS

Follow there are commands relative to saving configuration that the S-LOG can perform on memory card.

SET FILE

Command:  F path  
Command Dec:  70 32 path
Command Hex:  46 20 path
Response:  
Response Dec:  
Response Hex:  

The command set the file name of saving data, with pathname, with passed parameter. This file can be created on memory card of S-LOG, as described in OPERATIONS ON MEMORY CARD paragraph, in used for writing all data to save, that are those ones received by external system, and those one set before, as described in SAVING DATA paragraph.

The path parameter meet a string with its potential folder and file are delimited by ",", and have a maximum length of 8 character and add 3 for the extension for a complete length of 35 characters max. Whenever the passed path don't respect the rules 8.3, the format changes like this; if the path is passed through a string "FOLDER_DATA\FILE_DATA.TXT" then is converted in S-LOG in "FOLDER~1\FILE_D~1.TXT".

When is set a file duration the defined path through this command is used but without file, that is only to define the folder where save files.

Related to file for data saving, there a re a group of errors listed in ACQUIRE STATUS paragraph. The command is always performed, if the parameter is has a out fo range value not valid too: the external system has to guarantee the validity.

GET FILE

Command:  f  
Command Dec:  102
Command Hex:  66
Response:  path
Response Dec:  path
Response Hex:  path

It return a parameter relative to the file name of data saving, with path name, using by S-LOG. The means of the parameter is the same reported in previous command and is always as transmitted by SET FILE command, without transformation into 8.3 format.
FIGURE 40: COMPLETE VIEW
SET DURATION FILE

Command:   D duration  
Command Dec:  68 32 duration  
Command Hex:  44 20 duration

The command set the saving data file duration wrote on S-LOG, with the passed parameter. Following there are the description of available values for duration parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (78=4EH) -&gt;</td>
<td>None duration</td>
</tr>
<tr>
<td>H (72=48H) -&gt;</td>
<td>One hour duration file</td>
</tr>
<tr>
<td>D (68=44H) -&gt;</td>
<td>One day duration file</td>
</tr>
<tr>
<td>M (77=4DH) -&gt;</td>
<td>One month duration file</td>
</tr>
<tr>
<td>Y (89=59H) -&gt;</td>
<td>One year duration file</td>
</tr>
</tbody>
</table>

**Figure 41: Duration parameter**

The command is always performed, if the parameter is has a out of range value not valid too: the external system has to guarantee the validity.

With this command user can define if data saving have to be on a single file or more files, as described in detail in SAVING DATA paragraph. Indirectly the command define used file/s name by S-LOG that in fact can be define as described in SET FILE or those ones with yyyyymmdd.hh format(yyyy=year, mm=month, dd=day, hh=hour) associated to current time of saving.

The main features offered from this command is that to temporally organize received data by S-LOG and saved on memory card; the saving file will be contain all data that start from the same time to the name described in duration in that programmation.

For exemple if duration = D, in file 20080721.00 will be memorized all data of the day 21/07/2008; if duration = H, in file 20080822.12 will be memorized all data from 12:00:00 to 12:59:59 of 22/08/2008 day.

GET DURATION FILE

Command:   d  
Command Dec:  100  
Command Hex:  64

A parameter is returned a relative to saving file duration wrote by S-LOG.

The means of that parameter is the same shown in previous figure.

SET INSERTION ACTION

Command:   I action  
Command Dec:  73 32 action  
Command Hex:  49 20 action

**Figure 41: Duration parameter**
The command set the performed action from S-LOG in relation to insertion of a memory card, with the passed parameter. Following is reported the value description available to action parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (78=4EH) -&gt; None action</td>
<td></td>
</tr>
<tr>
<td>V (86=56H) -&gt; Verify if the card is formatted and format it</td>
<td></td>
</tr>
<tr>
<td>F (70=46H) -&gt; Always format inserted memory card.</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 42: Action parameter**

The command is always performed, if the parameter is has a out of range value not valid too: the external system has to guarantee the validity. The S-LOG always adds data to save at the end of proper file, indirectly with this command it define if in relation of card replacement, potential data already present will be kept or deleted. Infact with the first two settings, data are kept, while with the last one, data already saved are deleted. In this case is the user that has to save data present on memory card, before the insertion on S-LOG. To avoid problems during insertion, keep in mind that the used card has to be initially formatted with FAT or FAT16 standard, through a PC. For further information about this command are available OPERATIONS ON MEMORY CARD paragraph.

**GET INSERTION ACTION**

<table>
<thead>
<tr>
<th>Command:</th>
<th>i</th>
<th>Response:</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>105</td>
<td>Response Dec:</td>
<td>action</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>69</td>
<td>Response Hex:</td>
<td>action</td>
</tr>
</tbody>
</table>

It returns a parameter that meet the performed action by S-LOG in relation to insertion of a memory card.
The means of that parameter is shown in previous figure.

**SET GROUP TIME**

<table>
<thead>
<tr>
<th>Command:</th>
<th>G time</th>
<th>Response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>71 32 time</td>
<td>Response Dec:</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>47 20 time</td>
<td>Response Hex:</td>
</tr>
</tbody>
</table>

The command the recognizing time of data group, used by S-LOG to add a prefix and/or a suffix on data file, with the passed parameter.
Following is reported the value description available to **time** parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (00H)</td>
<td>Recognizing group time empty, that is each received data meet the group.</td>
</tr>
<tr>
<td>1÷250 (01÷FAH)</td>
<td>Recognizing group time from 20 to 5000 milliseconds</td>
</tr>
<tr>
<td>255 (FFH)</td>
<td>Recognizing group time disabled.</td>
</tr>
</tbody>
</table>

**Figure 43: Group time parameter**

**Figure 44: Data saving application with GAB H844**
The command is always performed, if the parameter is has a out fo range value not valid too: the external system has to guarantee the validity.

The group is a whole of consecutive data received by external unit without interruptions. When between the last data received and the previous is elapsed the time set, the **S-LOG** perform the following operation:

- manage the end of previous group;
- save on data file the potential suffix of complete group;
- manages the start of a new group;
- save on data file the potential prefix of new group;
- save the last received data.

The main function of data group is to organize the received information from external system on data file; for example it can adds current reception date and time of the group, makes a table of data for each group, adds one string that describes the groups of data, etc. Some examples for the group time are:

- Storage of date and time to transmitted data to a serial printer replaced by **S-LOG** (the recognizing time group has to be less than the minimum time between two printing);
- The adding of CR,LF to transmitted data every second to a measurator (in this case the recognizing time group has to be set to less of a second, value as 45 like a 900 milliseconds);
- Storage of the reception time of any character are in milliseconds;
- etc.

The parameter of command write the group time in twenties of milliseconds, with one eception: with the 255 value (FFH) the check is disable and the **S-LOG** will not recognize a data group and than will not save neither prefix or suffix.

For further information about the command are available in DATA SAVING paragraph and in those ones related to suffix and prefix.

### GET GROUP TIME

<table>
<thead>
<tr>
<th>Command:</th>
<th>g</th>
<th>Response:</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>103</td>
<td>Response Dec:</td>
<td>time</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>67</td>
<td>Response Hex:</td>
<td>time</td>
</tr>
</tbody>
</table>

A parameter is returns a relative parameter relative to recognizing time of a data group used by **S-LOG** for add a prefix and/or a suffix on data file.

The means of that parameter is the same shown in previous figure.

### SET LABEL FILE

<table>
<thead>
<tr>
<th>Command:</th>
<th>L label</th>
<th>Response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>76 32 label</td>
<td>Response Dec:</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>4C 20 label</td>
<td>Response Hex:</td>
</tr>
</tbody>
</table>

The command set the label from **S-LOG** after the creation of data saving file, with the passed parameter. This label is saved at the begin of created file, as descibed in OPERATION ON MEMORY CARD paragraph, and is used to identify data that will write in the same file.
The label parameter meet a string with max. length 30 characters che will be saved on file as configured on S-LOG with this command, with an exption: some special sequence that allow to format and add attributes and other data. In detail:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>\b (92 98=5C 61H)</td>
<td>Add SPACE code (one space) saving</td>
</tr>
<tr>
<td>\n (92 110=5C 6EH)</td>
<td>Add LF code (new line) saving</td>
</tr>
<tr>
<td>\r (92 114=5C 72H)</td>
<td>Add CR code (back line) saving</td>
</tr>
<tr>
<td>\t (92 116=5C 74H)</td>
<td>Add TAB code (horizontal ranging) saving</td>
</tr>
<tr>
<td>\h (92 104=5C 68H)</td>
<td>Add current time in hh:mm:ss format saving</td>
</tr>
<tr>
<td>\d (92 100=5C 64H)</td>
<td>Add current data in dd/mm/yyyy format saving</td>
</tr>
<tr>
<td>\e (92 101=5C 65H)</td>
<td>Add current date in mm/dd/yyyy format saving</td>
</tr>
<tr>
<td>\f (92 102=5C 66H)</td>
<td>Add current date in yyyy/mm/dd format saving</td>
</tr>
<tr>
<td>\m (92 109=5C 6DH)</td>
<td>Add current milliseconds in mmm format saving</td>
</tr>
</tbody>
</table>

**Figure 45: Special sequence label parameter**

Keep in mind that the space character is used logic protocol of configuration to divide parameters and than there is a special sequence able to save on data file this character.

If for example the S-LOG has to save trasmitted data by a measurator of temperature in output of a oven, with getting start time, the command can provide the following configuration parameter:

```
label = T.\boutput\boven\bd\bh\b\r\n
```

The command is always performed, if the parameter is has a out fo range value not valid too: the external system has to guarantee the validity.

**GET FILE LABEL**

<table>
<thead>
<tr>
<th>Command:</th>
<th>l</th>
<th>Response:</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>108</td>
<td>Response Dec:</td>
<td>label</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>6C</td>
<td>Response Hex:</td>
<td>label</td>
</tr>
</tbody>
</table>

As response there are a parameter that meet the saved label by S-LOG after the creation of data file storag.

The means of that parameter is the same reported in previous command, based on same special sequences for storage attributes.

**SET GROUP PREFIX**

<table>
<thead>
<tr>
<th>Commando:</th>
<th>B prefix</th>
<th>Response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>66 32 prefix</td>
<td>Response Dec:</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>42 20 prefix</td>
<td>Response Hex:</td>
</tr>
</tbody>
</table>

The command set the saved prefix by S-LOG right at the beginning of a group of data, with passed parameter. This prefix is saved before of the group data, as described in SAVING DATA and SET GROUP TIME and is used to identify the group of data that will be write in the same file.

The prefix parameter meet a string of max. 30 characters length that is exactly saved on file as is configured on S-LOG with this command, except some special sequences that allow to format and add attributes and other data. In details:
\textbf{FIGURE 46: SPECIAL SEQUENCE FOR PREFIX PARAMETER}

Keep in mind that space character is used by logic configuration protocol to isolate parameters and than there are a proper special sequence to save on data file these characters. If for example the \textit{S-LOG} has to save data for a serial printer adding current time and date in european format, than the command can provide the follow configuration parameter:

\texttt{prefix = Print\bdel\bore\b:}

The command is always executed, even if the parameter has a invalid value: it is the external system that has to guarantee the goodness.

\textbf{GET GROUP PREFIX}

\begin{tabular}{ll}
\textbf{Command:} & \texttt{b} \\
\textbf{Command Dec:} & \texttt{98} \\
\textbf{Command Hex:} & \texttt{62} \\
\end{tabular}

\begin{tabular}{ll}
\textbf{Response:} & \texttt{prefix} \\
\textbf{Response Dec:} & \texttt{prefix} \\
\textbf{Response Hex:} & \texttt{prefix} \\
\end{tabular}

As response there are a parameter that meet the saved prefix by \textit{S-LOG} right at the beginning of a group of data.

The means of that parameter is the same reported in previous command, based on same special sequences for storage attributes.

\textbf{SET GROUP SUFFIX}

\begin{tabular}{ll}
\textbf{Command:} & \texttt{E suffix} \\
\textbf{Command Dec:} & \texttt{69 32 suffix} \\
\textbf{Command Hex:} & \texttt{45 20 suffix} \\
\end{tabular}

\begin{tabular}{ll}
\textbf{Response:} & \texttt{} \\
\textbf{Response Dec:} & \texttt{} \\
\textbf{Response Hex:} & \texttt{} \\
\end{tabular}

The command set the suffix saved by \textit{S-LOG} right at the end of a group of data, with the passed parameter. This suffix is saved after the data of the group, as described in SAVING DATA and SET GROUP TIME paragraph and is used characterize or format the group of data written in the same file.

The \texttt{suffix} parameter meet the string of 30 max. characters length that is exactly saved on file as is configured on \textit{S-LOG} with this command, except some special sequences that allow to format and add attributes and other data. In detail:
Keep in mind that the space character is used logic protocol of configuration to divide parameters and than there is a special sequence able to save on data file this character. For example, if the S-LOG has to save data periodically transmitted from a temperature tester adding the unit of measure and ranging the same data, then the command can provide the configuration parameter:

```
  suffix = \t°C
```

The command is always executed, even if the parameter has a invalid value: is the external system that has to guarantee the goodness.

### GET GROUP SUFFIX

<table>
<thead>
<tr>
<th>Command:</th>
<th>e</th>
<th>Response:</th>
<th>suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Dec:</td>
<td>101</td>
<td>Response Dec:</td>
<td>suffix</td>
</tr>
<tr>
<td>Command Hex:</td>
<td>65</td>
<td>Response Hex:</td>
<td>suffix</td>
</tr>
</tbody>
</table>

As response there are a parameter that meet the saved prefix by S-LOG right at the beginning of a group of data.

The means of that parameter is the same reported in previous command, based on same special sequences for storage attributes.
HOW TO START

In this chapter are described the operation to perform for a first use of S-LOG in linear and quickly mode, without none beginning problem. In particular is reported the right operations sequence that user has to perform for configure and for utilize the product. To semplify the beginning step, in this chapter we can suppose to replace a panel serial printer with a S-LOG and to use a minimal system available from a customer, in other words a PC with a serial line in RS 232.

PRELIMINARY OPERATIONS

A1) Read all the documentation received.

A2) Arrange the S-LOG for work, find a proper power supply source, verify that jumper configuration is for serial line in RS 232 and the back up battery is connected (see figures 18 and 20) and remove the potentially memory card. So is restored the default hardware configuration.

A3) Perform the serial connection described in follow figure so connect the two communication signals (TX RS232, RX RS232), reference ground (GND) and handshake signals (CTS RS232, RTS RS232) to a communication port COMx of the PC.

A4) Switch on the Personal Computer.

A5) In case of the PC serial line arrives from (for example USB <-> RS 232 interfaces), perform the proper operations described into the documentation and verify that all works fine.
CONFIGURATION

B1) Install on PC the Conf_SLOG program, that is the utility realized by grifo® to quickly configure the S-LOG. When the installation program asks to substitute some system files, please reply to maintain the original ones (Keep).

B2) Launch the Conf_SLOG that at the first switch on is set for english language.

B3) Open the S-LOG box, select the configuration mode through the jumper J1 in 1-2, and provide power supply (A2 point): LED has to blink in red, green and yellow.

B4) Configure the communication line of PC for default physical protocol of S-LOG, through Program | Serial option. In the window displayed select the serial line connected to A3 point, the reported parameters in follow figure, press the Confirm button and check that the operation is going fine.

![Figure 49: PC Serial Setting with Conf_SLOG](image)

B5) Verify if communication between PC and S-LOG works fine, selecting the General | Versions option. With this option is displayed the follow info window with hardware and firmware versione of S-LOG in use and has to closed through Exit button.
B6) Set the kind of serial communication that S-LOG has to use to receive data to save, through Configue | Communication option.

In the previous example we supposed that printer to replace communicates with RS 232 serial line at 9600 Baud, none parity, 1 stop bit and without handshake.

In window displayed we have to press before the Preleva configurazioni button, than select the proper settings as described in the following figure and at the end press the Set configuration button. At this point we have to wait the saving of all configuration commands that is finished after tens of seconds, verify that no errors are found (parameters on green background) and close the windows with the Exit button.
B6) Set data referred to saving that the S-LOG has to perform on memory card, through Configure | Saving option. In the application example supposed if between a printing and the following the time elapsed is about 1 second, only one file is created with printing data, the file is called PRNOVEN.TXT, at the beginning there is the identification “OUTPUT TEMPERATURE OVEN” and the messagees of each print are composed by current date and time too. In displayed window we have to set the parameters as described in the follow figure and at the end to press the Set configuration button. Now we have to wait the saving of all configuration command after tens of seconds, verify that there are no errors are found and close the window with the Exit button.

![Conf_SLOG 4.4: S-LOG saving configuration](image)

**Figure 52: Saving configuration with Conf_SLOG**

B7) Check or set current date and time of the clock of S-LOG through General | Clock option.

B8) Who wants to try all features of Conf_SLOG program can also use the other options linked to other menu. Note that these proofs are not necessary to configuration and use of S-LOG. Keep in mind that the utility program Conf_SLOG provides the described command in the previous chapter only. In its main window are also reported the communicated data with difference between those ones directed from PC to S-LOG (commands), and those ones from S-LOG to PC (response). Through this windows the user can check all the aspects of communication.
B9) After configuration and verify, quit from **Conf_SLOG** and come back to Windows operating system, through *Program | Exit* proper option.

B10) Switch off the **S-LOG**, select the data reception and saving mode with the jumper J1 in 2-3, and close the card into the container.

B11) At this point the **S-LOG** is completely configured and ready to work in the application.

**USE**

C1) Connect the communication line of **S-LOG** (see at B4 point), to the external system transmit data to save.
   In the example of application that we have supposed is sufficient to disconnect RS 232 signals from printer and connect them to CN1 connector of **S-LOG**. In the configuration phase the handshakes wasn't activated, in this examples we can connect only two wires: the reception one (RX RS232) and the reference ground (GND).

C2) Supply a voltage to **S-LOG** and verify that the LED is blincking red, green and yellow.

C3) Format one memory card with FAT16 or FAT format, through a PC, insert it into the **S-LOG** and verify that the LED become green.

C4) From this moment the **S-LOG** write on memory card all received data, that are all information printed before, with saving modes defined at B6 point. The user has to check the LED become red during described saving and viceversa keep green in other conditions.

C5) When needed the memory card can be removed from **S-LOG**, making sure that the LED is not red, that show the presence of a saving in progress.

C6) The memory card removed can be inserted in any PC and can be found the **PRNOVEN.TXT** file, that contain all the information printed before, with date and time. This information can be examined opening the file through any editor, like NotePad in Windows system.
In the application supposed the same file can be opened with a data program like EXCEL, so the data that were periodically printed, will be put on a table with acquisition date and time. With the several possibilities of EXCEL the same data can printed, getting the original status, than work out, built in graphic, pressed, registered, etc. based on the final user's requirements.
C7) At this point the memory card can be used for another acquisition: re-inserting it into the S-LOG will be formatted for delete potential data already picked up make it ready for new saving, as defined in configuration at B6 point.
APPENDIX A: COMMANDS SUMMARY TABLES

The following tables list of all the commands recognized by **S-LOG**. In these tables the command sequences and response are reported in mnemonic format and if the user need the decimal or hexadecimal one can see the paragraphs of the CONFIGURATION COMMANDS chapter. Please remind that the commands are always identified from the first character and that the following data meet the parameters of the same commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comunication from main system to S-LOG</th>
<th>Comunication from S-LOG to main system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear errors</td>
<td>Z</td>
<td>-</td>
</tr>
<tr>
<td>Get status</td>
<td>z</td>
<td>general status disk status</td>
</tr>
<tr>
<td>Get versions</td>
<td>v</td>
<td>hw ver. fw ver.</td>
</tr>
<tr>
<td>Set Real Time Clock</td>
<td>T dd/mm/yyyy hh:mm:ss</td>
<td>-</td>
</tr>
<tr>
<td>Get Real Time Clock</td>
<td>t</td>
<td>gg/mm/aaaa oo:mm:ss</td>
</tr>
<tr>
<td>Set Communication line</td>
<td>C com.line</td>
<td>-</td>
</tr>
<tr>
<td>Get Communication line</td>
<td>c</td>
<td>com.line</td>
</tr>
<tr>
<td>Set Baud Rate</td>
<td>R baud</td>
<td>-</td>
</tr>
<tr>
<td>Get Baud Rate</td>
<td>r</td>
<td>baud</td>
</tr>
<tr>
<td>Set Parity</td>
<td>P parity</td>
<td>-</td>
</tr>
<tr>
<td>Get Parity</td>
<td>p</td>
<td>parity</td>
</tr>
<tr>
<td>Set Stop Bit</td>
<td>S stop</td>
<td>-</td>
</tr>
<tr>
<td>Get Stop Bit</td>
<td>s</td>
<td>stop</td>
</tr>
<tr>
<td>Set Handshake</td>
<td>H handshake</td>
<td>-</td>
</tr>
<tr>
<td>Get Handshake</td>
<td>h</td>
<td>handshake</td>
</tr>
<tr>
<td>Set Slave Address</td>
<td>A sla.add</td>
<td>-</td>
</tr>
<tr>
<td>Get Slave Address</td>
<td>a</td>
<td>sla.add</td>
</tr>
<tr>
<td>Set Communication Timeout</td>
<td>O tout</td>
<td>-</td>
</tr>
<tr>
<td>Get Comunication Timeout</td>
<td>o</td>
<td>tout</td>
</tr>
</tbody>
</table>

**FIGURE A1: CONFIGURATION COMMANDS TABLE (1 OF 2)**
<table>
<thead>
<tr>
<th>Command</th>
<th>Communication from main system to S-LOG</th>
<th>Communication from S-LOG to main system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set File</td>
<td>F pathname</td>
<td></td>
</tr>
<tr>
<td>Get File</td>
<td>f</td>
<td>pathname</td>
</tr>
<tr>
<td>Set file duration</td>
<td>D duration</td>
<td></td>
</tr>
<tr>
<td>Get file duration</td>
<td>d</td>
<td>duration</td>
</tr>
<tr>
<td>Set Insertion action</td>
<td>I action</td>
<td></td>
</tr>
<tr>
<td>Get insertion action</td>
<td>i</td>
<td>action</td>
</tr>
<tr>
<td>Set Group Time</td>
<td>G time</td>
<td></td>
</tr>
<tr>
<td>Get Group Time</td>
<td>g</td>
<td>tempo</td>
</tr>
<tr>
<td>Set File Label</td>
<td>L label</td>
<td></td>
</tr>
<tr>
<td>Get File Label</td>
<td>l</td>
<td>label</td>
</tr>
<tr>
<td>Set Group Prefix</td>
<td>B prefix</td>
<td></td>
</tr>
<tr>
<td>Group Group Prefix</td>
<td>b</td>
<td>prefix</td>
</tr>
<tr>
<td>Set Group Suffix</td>
<td>E suffix</td>
<td></td>
</tr>
<tr>
<td>Get Group Suffix</td>
<td>e</td>
<td>suffix</td>
</tr>
</tbody>
</table>

**Figure A2: Configuration commands table (2 of 2)**
APPENDIX B: DEFAULT CONFIG., OPTIONS, ACCESSORIES

In correspondence of the first purchase, or after a reparation, the S-LOG is supplied in its base configuration. The features of this configuration has been described many times in the manual (by using also the name default configuration) and in this appendix they are summarized, opportunely divided in the following tables.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>ASSOCIATED COMMANDS</th>
<th>DEFAULT SETTING</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.line</td>
<td>C c</td>
<td>R</td>
<td>Used communication line = asynchronous serial in RS 232 or TTL</td>
</tr>
<tr>
<td>baud</td>
<td>R r</td>
<td>19200</td>
<td>Baud rate for asynchronous serial communication RS 232, TTL</td>
</tr>
<tr>
<td>parity</td>
<td>P p</td>
<td>N</td>
<td>Parità check for asynchronous serial communication RS 232, TTL = none</td>
</tr>
<tr>
<td>stop</td>
<td>S s</td>
<td>1</td>
<td>Stop bit number asynchronous serial communication RS 232, TTL</td>
</tr>
<tr>
<td>handshake</td>
<td>H h</td>
<td>N</td>
<td>Handshake mode management for asynchronous serial communication RS 232, TTL = none</td>
</tr>
<tr>
<td>sla.add</td>
<td>A a</td>
<td>128</td>
<td>S-LOG address for synchronous serial communication I2C BUS</td>
</tr>
<tr>
<td>tout</td>
<td>O o</td>
<td>250</td>
<td>Timeout waiting data = 250 * 20 ms = 5000 ms = 5 seconds</td>
</tr>
<tr>
<td>path</td>
<td>F f</td>
<td>SLOG.DAT</td>
<td>File name on memory card to use to store received data</td>
</tr>
<tr>
<td>duration</td>
<td>D d</td>
<td>N</td>
<td>Duration of the file for data store = none. Save on a sole file defined by path.</td>
</tr>
<tr>
<td>action</td>
<td>I i</td>
<td>N</td>
<td>Action performed on memory card inserted in S-LOG = none</td>
</tr>
<tr>
<td>time</td>
<td>G g</td>
<td>255</td>
<td>Time to recognize of a data group = disabled</td>
</tr>
<tr>
<td>label</td>
<td>L l</td>
<td></td>
<td>Saved label at the beginning of file on memory card = none</td>
</tr>
<tr>
<td>prefix</td>
<td>B b</td>
<td></td>
<td>Saved prefix at the beginning of a data group on memory card = none</td>
</tr>
<tr>
<td>suffix</td>
<td>E e</td>
<td></td>
<td>Saved suffix at the end of a data group on memory card = none</td>
</tr>
</tbody>
</table>

**Figure B1: Default configuration table**

The values listed in previous table can be modified through the local setup modality, as described with details in the homonimous paragraph.
Please remind that the jumpers default configuration proposed is the one relative to base version of terminal, that is without any options.

During the order phase the user can add to S-LOG, the following features:

**FIGURE B2: DEFAULT JUMPERS CONFIGURATION TABLE**

<table>
<thead>
<tr>
<th>JUMPER</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>position 2-3</td>
</tr>
<tr>
<td>J2</td>
<td>connected</td>
</tr>
<tr>
<td>J4</td>
<td>position 1-2</td>
</tr>
<tr>
<td>J5÷J10</td>
<td>position 1-2</td>
</tr>
</tbody>
</table>

**FIGURE B3: OPTIONS TABLE**

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>.SW</td>
<td>Power supply switching section with wide range input</td>
</tr>
</tbody>
</table>

All these options are described in the paragraphs of the manual that illustrate the functionalities and the use of the same additional features. It is suggested to use the final alphabetical index, to found these paragraphs in a short time.

In addition there are a list of accessories that simplify and speed up the use of the module. Among these ones we remind the following available products:

- **CKS.AMP4** kit composed by female AMP Mod II 4 pins, plus 4 contacts to crimp;

**FIGURE B4: CKS.AMP4 CONNECTION ACCESSORY**

These components can be acquired directly from AMP dealers by using P/N 280359 and P/N 182206-2.
- **AMP4.Cable**  complete connector with 4 coloured wires, 1 metre length;

![AMP4.Cable Connection Accessory](Figure B5)

**Figure B5: AMP4.Cable Connection Accessory**

- **CKS.AMP8**  kit composed by female AMP Mod II 8 pins, plus 8 contacts to crimp;

![CKS.AMP8 Connection Accessory](Figure B6)

**Figure B6: CKS.AMP8 Connection Accessory**

These components can be acquired directly from AMP dealers by using P/N 280365 and P/N 182206-2.
- AMP8.Cable  complete connector with 8 coloured wires, 1 metre length;

**FIGURE B7: AMP8.CABLE CONNECTION ACCESSORY**

- EL 12  power supply for direct connection to mains voltage at 230 Vac, 50 Hz, that generates an output voltage of 12 Vdc, compatible for S-LOG. The photo of this accessories is already available in previous pages of manual, on figure 22.