S-LOG

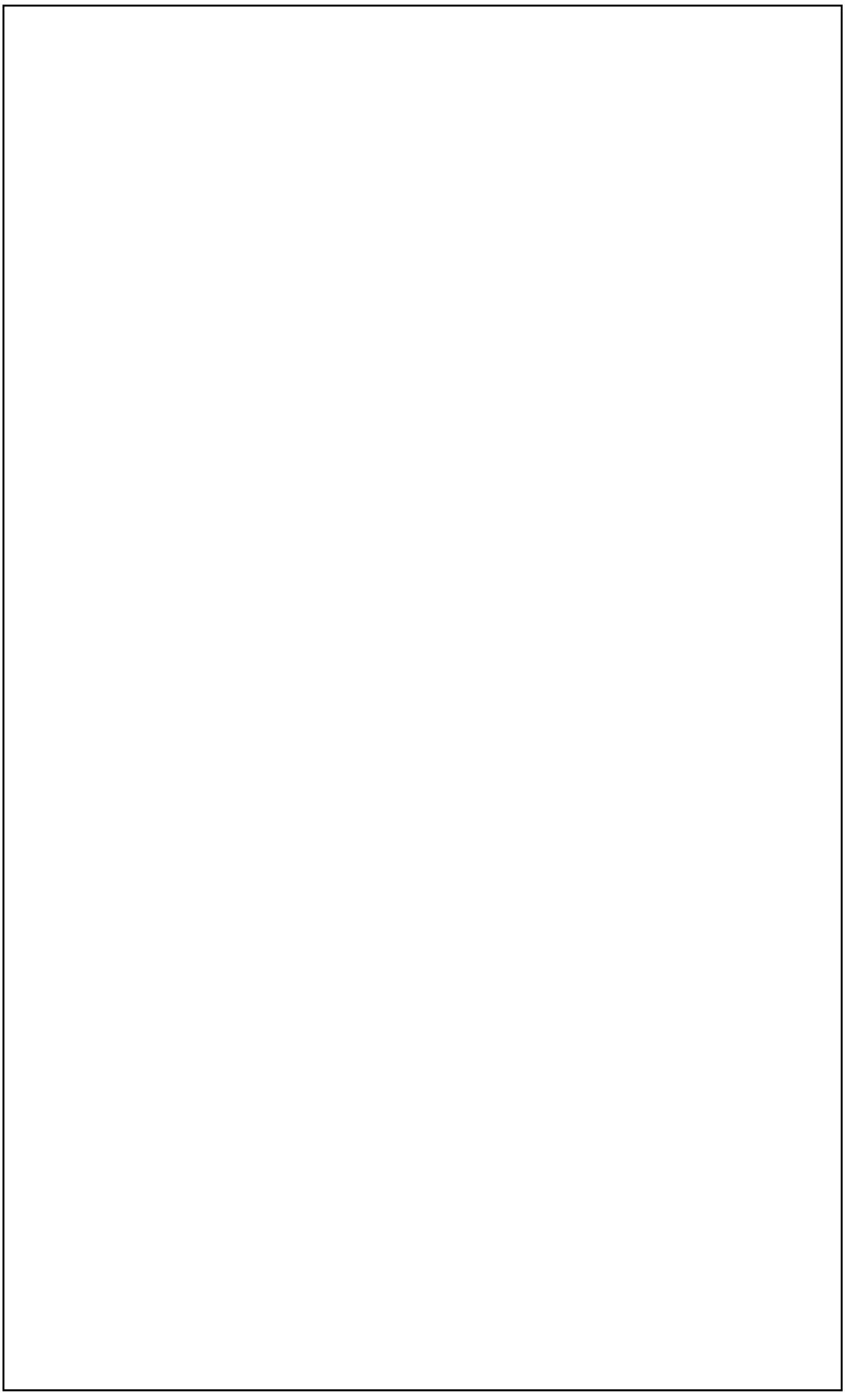
Serial-LOGger for Secure Digital cards

USER MANUAL

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

S-LOG Rel. 5.25 *Edition 29 August2011*



S-LOG

Serial-LOGger for Secure Digital cards

USER MANUAL

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.

grifo®
ITALIAN TECHNOLOGY
Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

S-LOG Rel. 5.25 *Edition 29 August2011*

IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsability 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

GENERAL INDEX

FIGURES INDEX

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 rispect 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.

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 beetween 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.

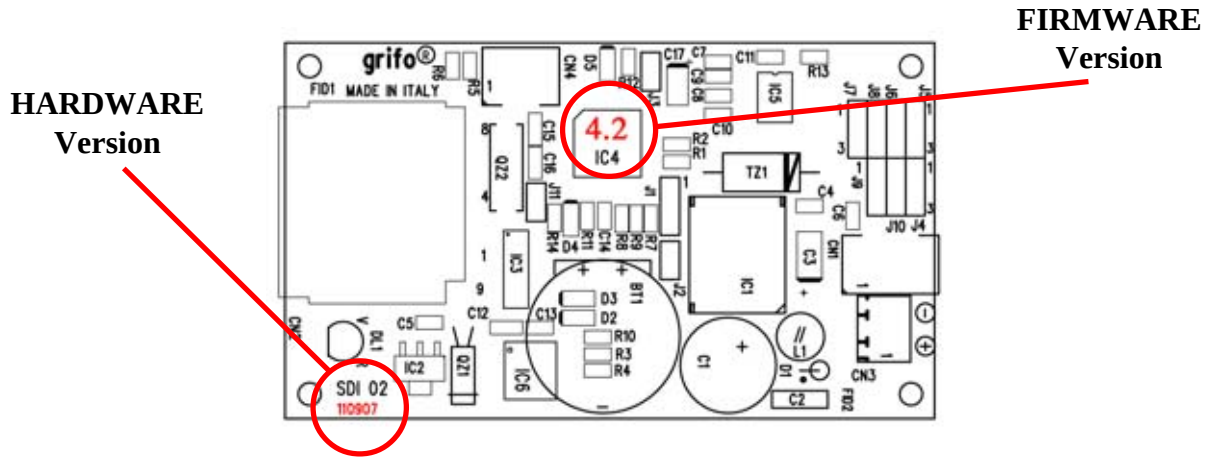


FIGURE 1: LOCATION OF HARDWARE AND FIRMWARE VERSION

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 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. 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 acceppts 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 momorization 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, diplayed 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 tme 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 **bicolour 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: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 begginning 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 receveing 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 seePOWER 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 comunicate 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 recomended:

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 accordig 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 BUSline 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	
Storege on file duration:	Hour, Day, month, Year	
Network unity number:	128	with synchronous serial I2C BUS
Communication:	Select between: (I2C BUS) *Default:*	asynchronous (RS 232,TTL) and synchronous *asynchronous*
Asynchronous communication protocol (RS 232, TTL):	Baud rate: Bit per character: Parity: Stop Bit: Handshake: *Default:*	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 8 off, even, odd 1, 2 off, hardware, software, software repeated *19200 Baud, 8 Bit, None parity, 1 Stop, None handshake*
Synchronous communication protocol (I2C BUS):	Bit rate: Mode: Slave Address: *Default:*	from 225 to 460.799 bits per second Slave from 00H to FEH step of 2 *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**)

Measure conditions	*consumption max. +5 Vdc*	*Consumpion max. 10÷38 Vdc 8÷24 Vac*
S-LOG without memory card, not connected to other systems	44 mA	0,23 W
S-LOG with memory card, connected, at work	60 mA	0,39 W

FIGURE 3: CONSUMPION 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 totipical 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 used by user.

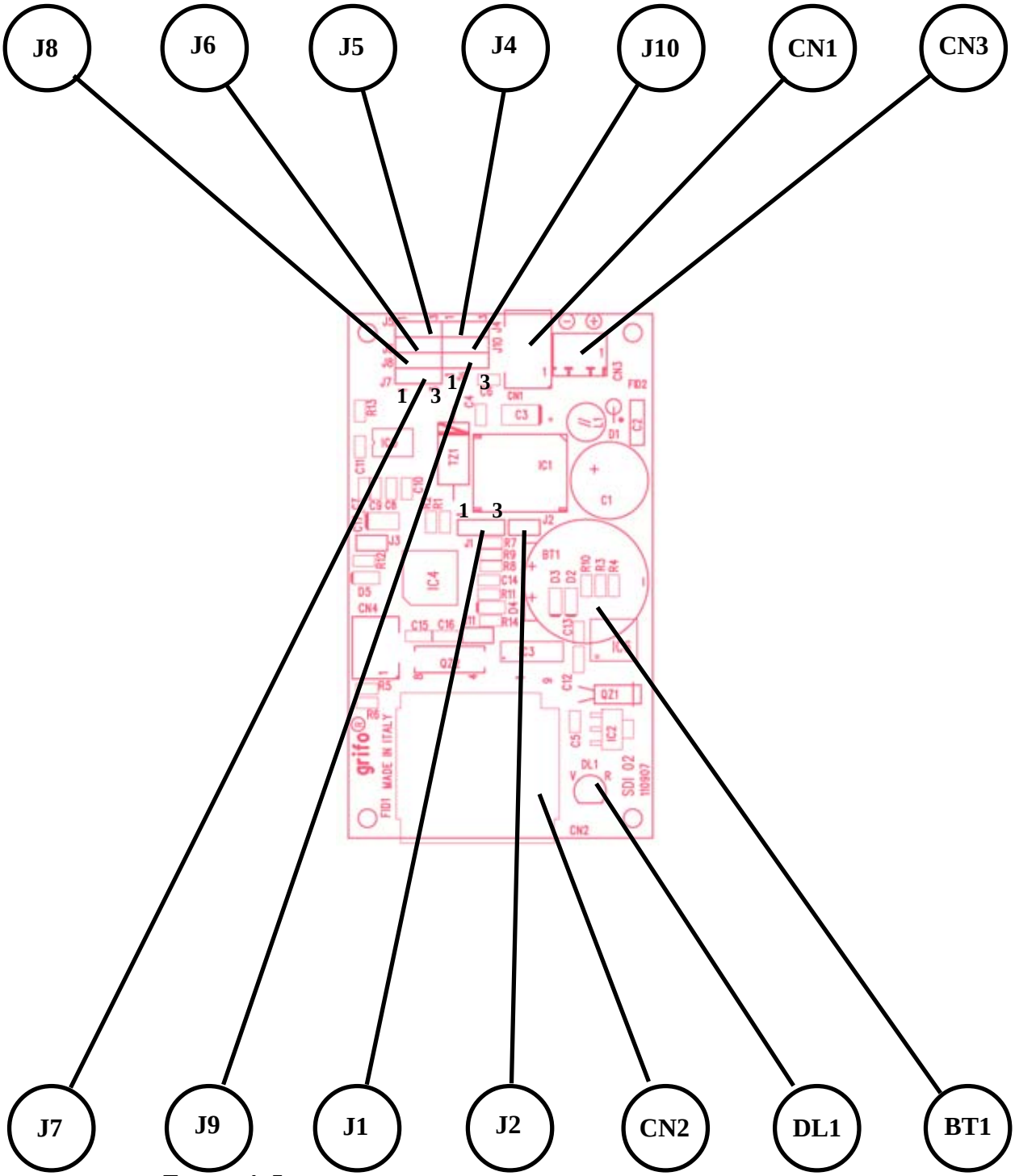


FIGURE 4: LOCATION OF JUMPERS, CONNECTORS, LED, BATTERY, ETC.

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 **.SW** optionand is present the power supply section, able to accept a wide range voltage.

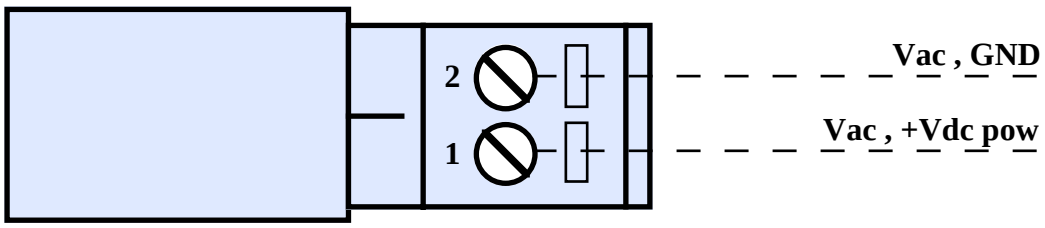


FIGURE 5: CN3 - POWER SUPPLY CONNECTOR

Signals description:

Vac = I - AC power supply lines connected to on board switching section; these signals must be in the range **8÷24 Vac**.

+Vdc pow = I - DC power supply lines connected to on board switching section (**+10÷+38 Vdc**) or stabilized (**+5 Vdc**) voltage connected to on board logic, according to ordered configuration.

GND = - Ground signal for DC power supply.

NOTE For further information about power supply configurations, please refer to paragraph POWER SUPPLY, to next figures and the CN1 -SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR paragraph.

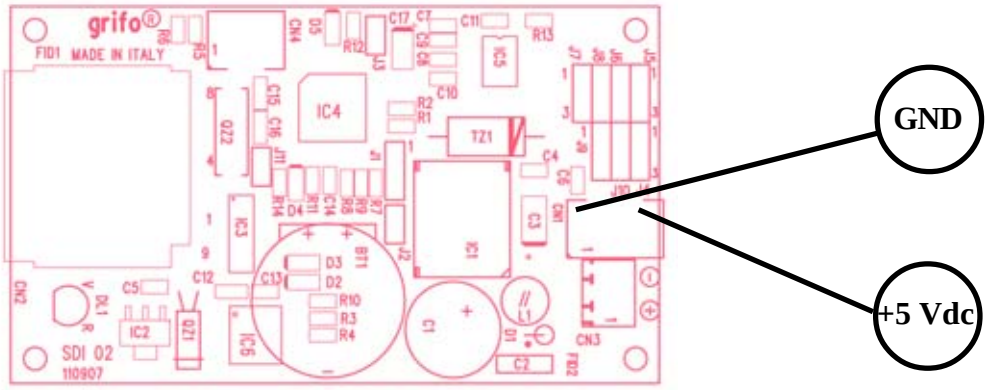


FIGURE 6: +5 VDC REGULATED POWER SUPPLY CONNECTION

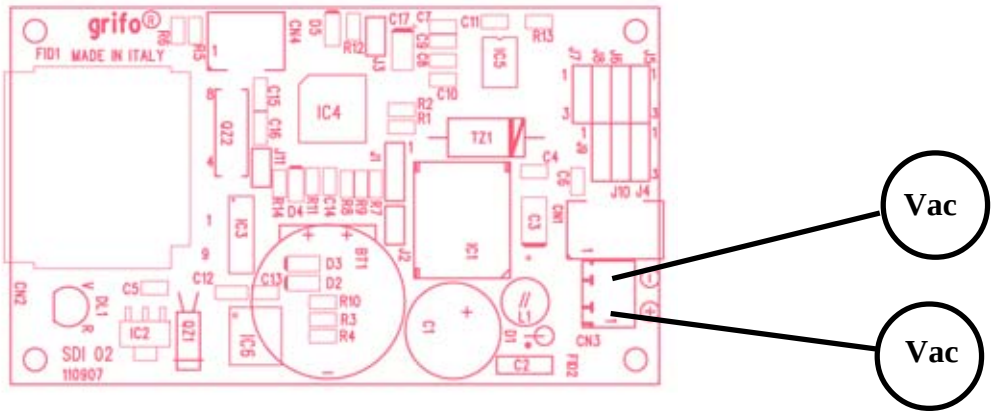


FIGURE 7: 8÷24 VAC ALTERNATE POWER SUPPLY CONNECTION (WITH .SW OPTION)

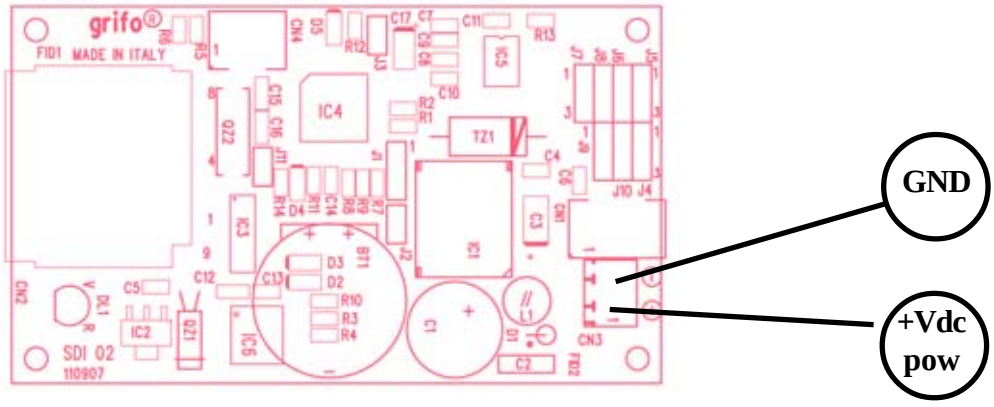


FIGURE 8: 10÷38 VDC DIRECT POWER SUPPLY CONNECTION (WITH .SW OPTION)

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.

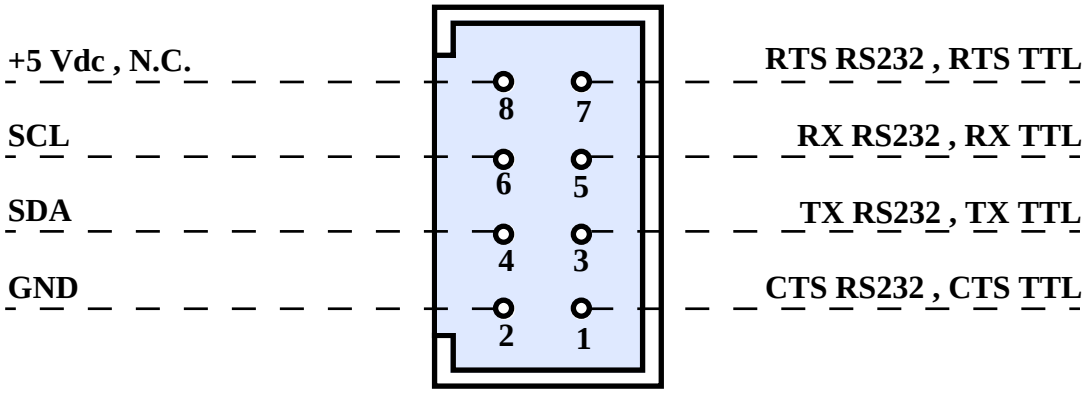


FIGURA 9: CN1 SERIAL COMMUNICATION LINE AND SWITCHING CONNECTOR

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 SUPPLYparagraphs and see figures 6÷8.
For details about asynchronous cominication we recommend to examine the ASYNCHRONOUS SERIAL LINE COMMUNICATION paragraph and to see figures 10÷16 that show connection examples.

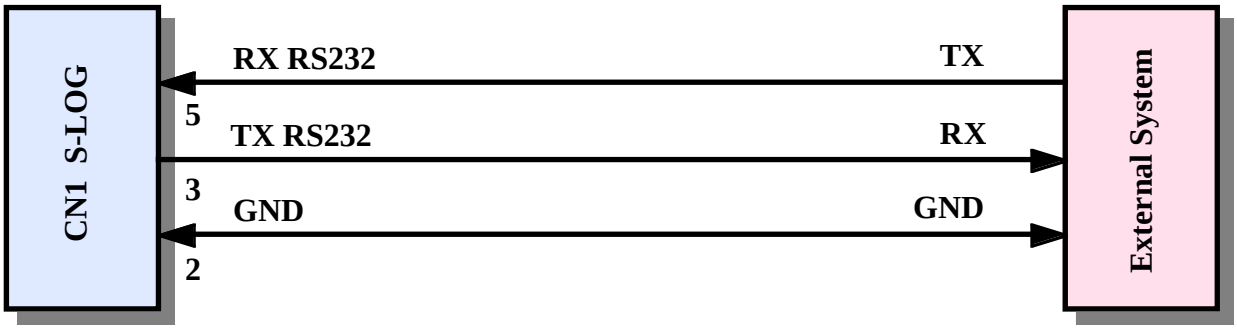


FIGURE 10: CONNECTION EXAMPLE IN RS 232 WITH THREE WIRES (WITHOUT HANDSHAKE)

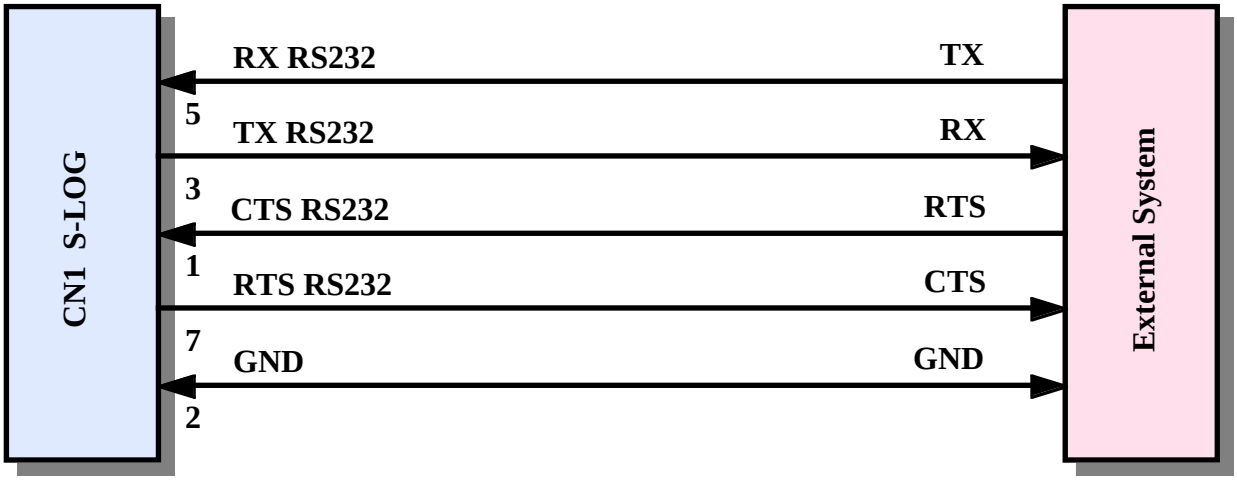


FIGURE 11: CONNECTION EXAMPLE IN RS 232 WITH FIVE WIRES (WITH HANDSHAKE)

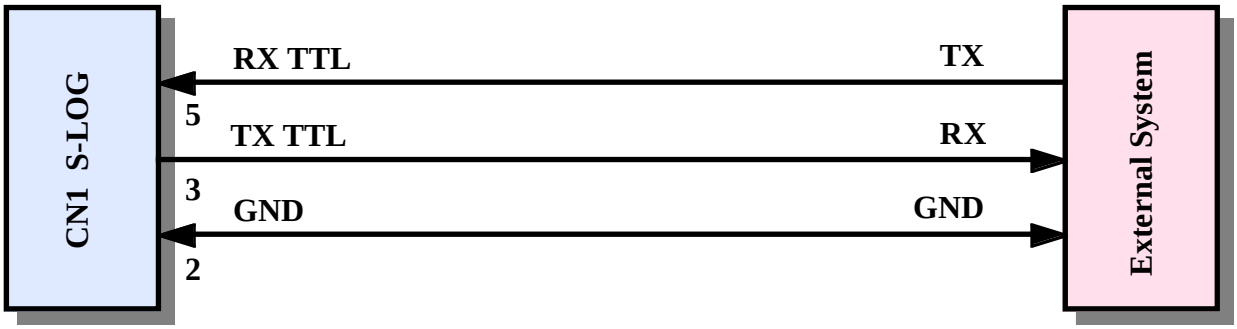


FIGURE 12: CONNECTION EXAMPLE IN TTL WITH THREE WIRES (WITHOUT HANDSHAKE)

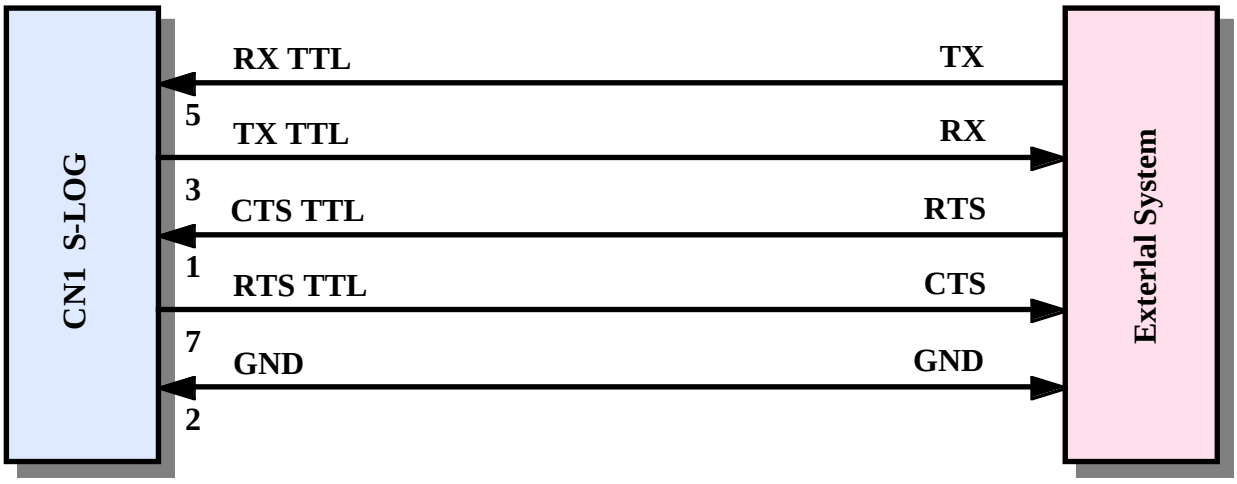


FIGURE 13: CONNECTION EXAMPLE IN TTL WITH FIVE WIRE (WITH HANDSHAKE)

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.

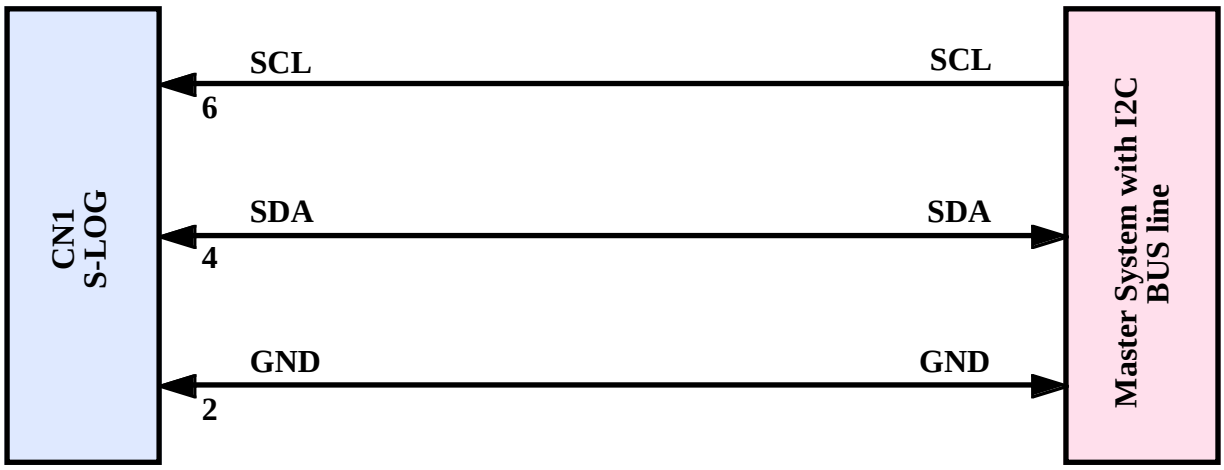


FIGURE 14: CONNECTION EXAMPLE POINT TO POINT IN I2C BUS WITHOUT POWER SUPPLY

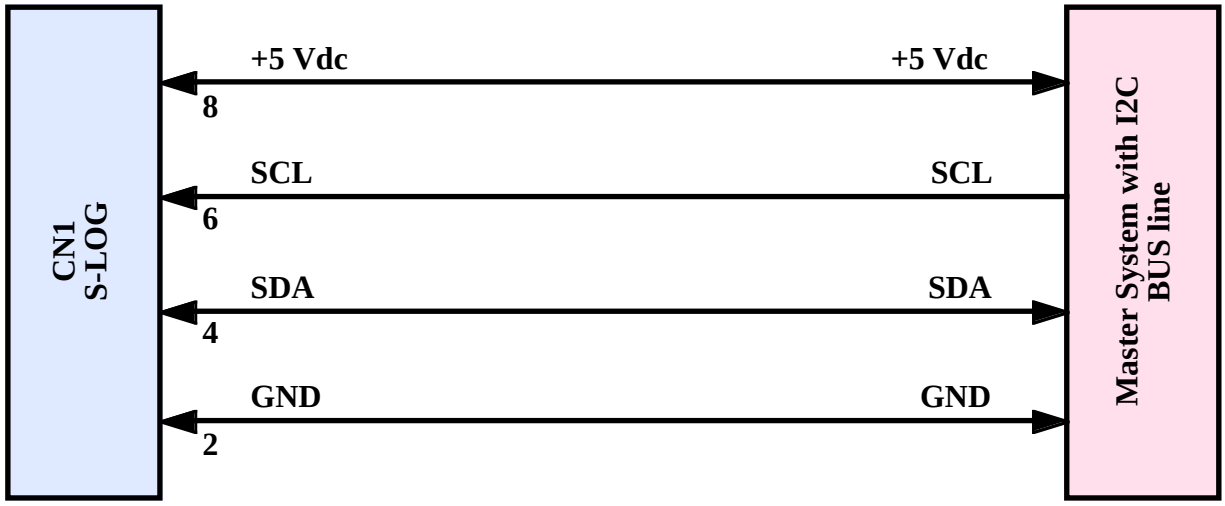


FIGURE 15: CONNECTION EXAMPLE POINT TO POINT IN I2C BUS WITH POWER SUPPLY

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.

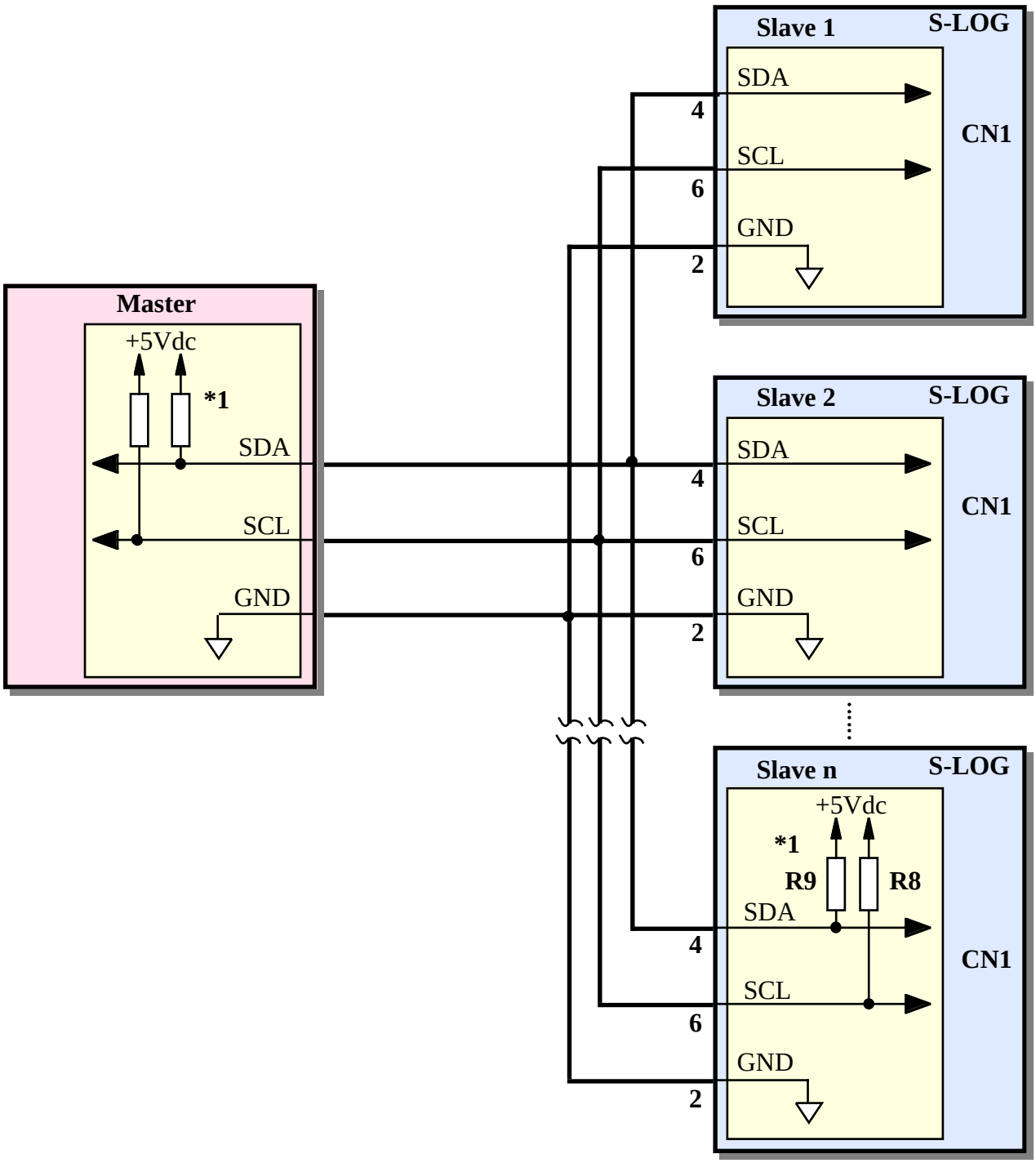


FIGURE 16: NETWORK CONNECTION EXAMPLE I2C BUS COMMUNICATION

Please remind that in a **I2C BUS** must be present two pull up resristor 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 paragrph. 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 fo those line as shown in the previous figure on slaves 1and 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, infact 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 **machanical 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.

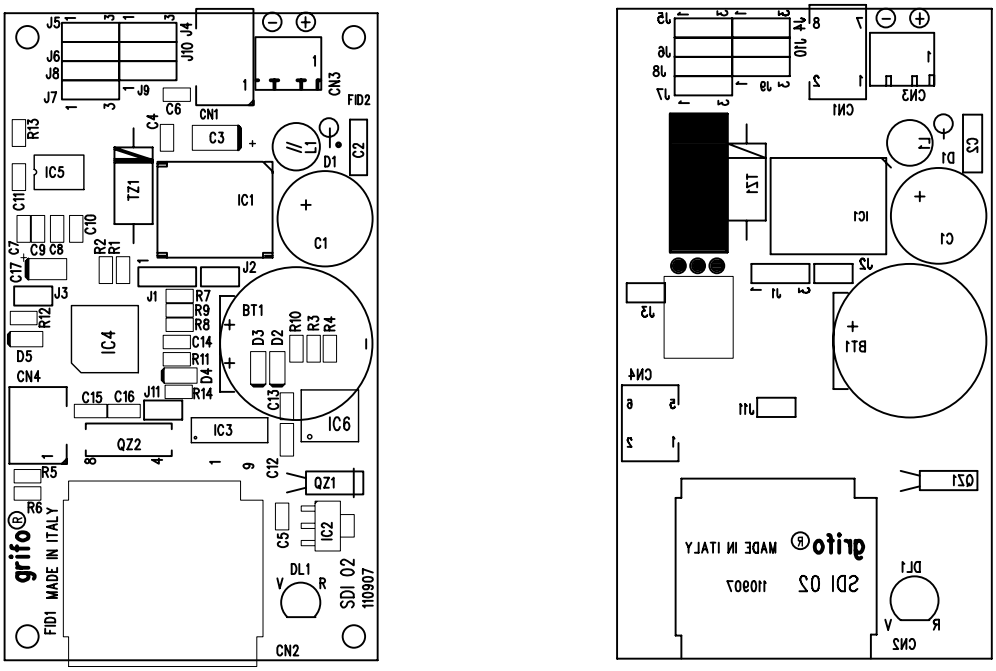


FIGURE 17: COMPONENTS MAP COMPONENTS SIDE AND SOLDER SIDE

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:

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

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, minths 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 timeand 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 otbain the the time of some events, calculate production data in a period, to make graphic on base time, etc... The user can cannect 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:

STATUS	*CONDITIONS*
switch off	**S-LOG** not powered or out of order.
red, green, yellow alternating in about each half second	**S-LOG** working but without memory card.
green	**S-LOG** is working with memory card inserted.
red	**S-LOG** is saving received data, on memory card.
yellow	**S-LOG** is working and is performing a configuration command.
blinking yellow	**S-LOG** is working but with errors.

FIGURE 19: SIGNALLING LED

As shown in figure 4 the LED is easily visible in the insertion side if memory card; per 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 choise 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 choosen aynchronous serial, selction of electric protocol (RS 232 or TTL) is via hardware and reach a proper jumper configuration, as described on figur 18 table.
Following there are two figure that show jumper configuration involved in this selection, to make realization easy:

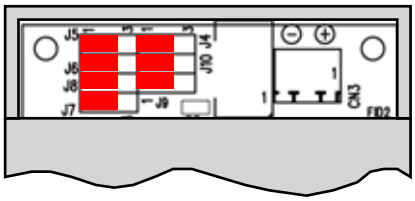


FIGURE 20: ASYNCHRONOUS SERIAL RS 232 JUMPER CONFIGURATION

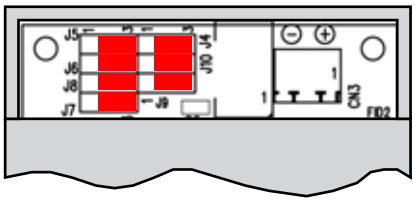


FIGURE 21: ASYNCHRONOUS SERIAL TTL JUMPER CONFIGURATION

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 seral 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 efficent 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 voltageas +5 Vdc. For alternate current it can ordered with a 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 decribed 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 distribuited 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: PWER 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 exernal system, and in order to be able to satisfy the huge number of applications requests, it is also avalaible 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 avalaible 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 indipendent and separated, selected by the proper jumper J1.

J1 in 1-2 position -> selects the configutation mode.
J1 in 2-3 position -> selets 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 comunication 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 malfuctionings, 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 mdule, 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 estabilished 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.

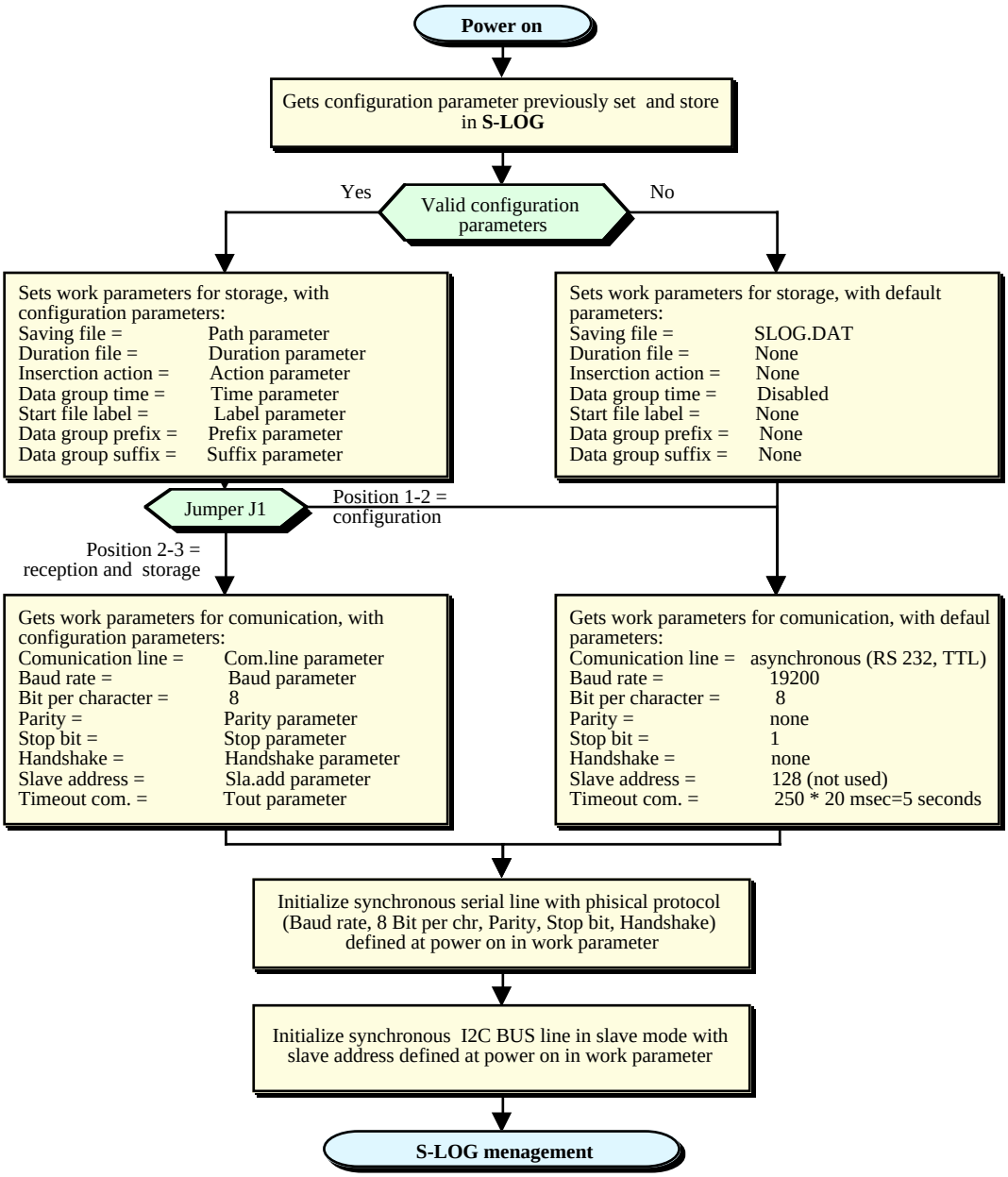


FIGURE 23: WORK PARAMETERS DEFINITION

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 cominication 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. waithing 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.

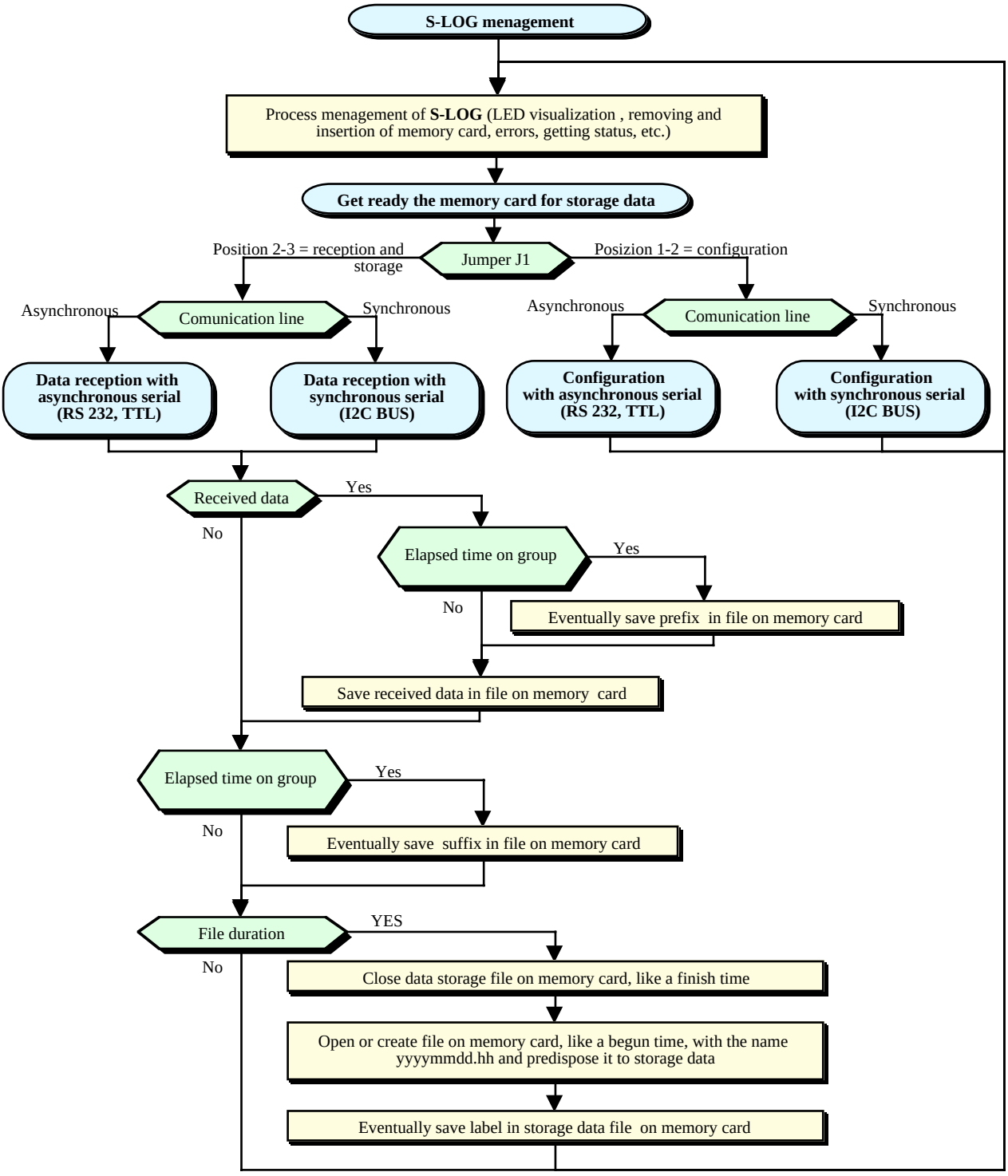


FIGURE 24: PERFORMED OPERATIONS

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;
- aget 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 dipends on several factors as described in figures 23 and 24. When asynchronous serial line is choosen, 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 semplify and increase speed the preparation and check out of the moddule 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 impotant 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 separed 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 potentialcommunication handshakes and timeout, as described in paragraphs bearing the same name.

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

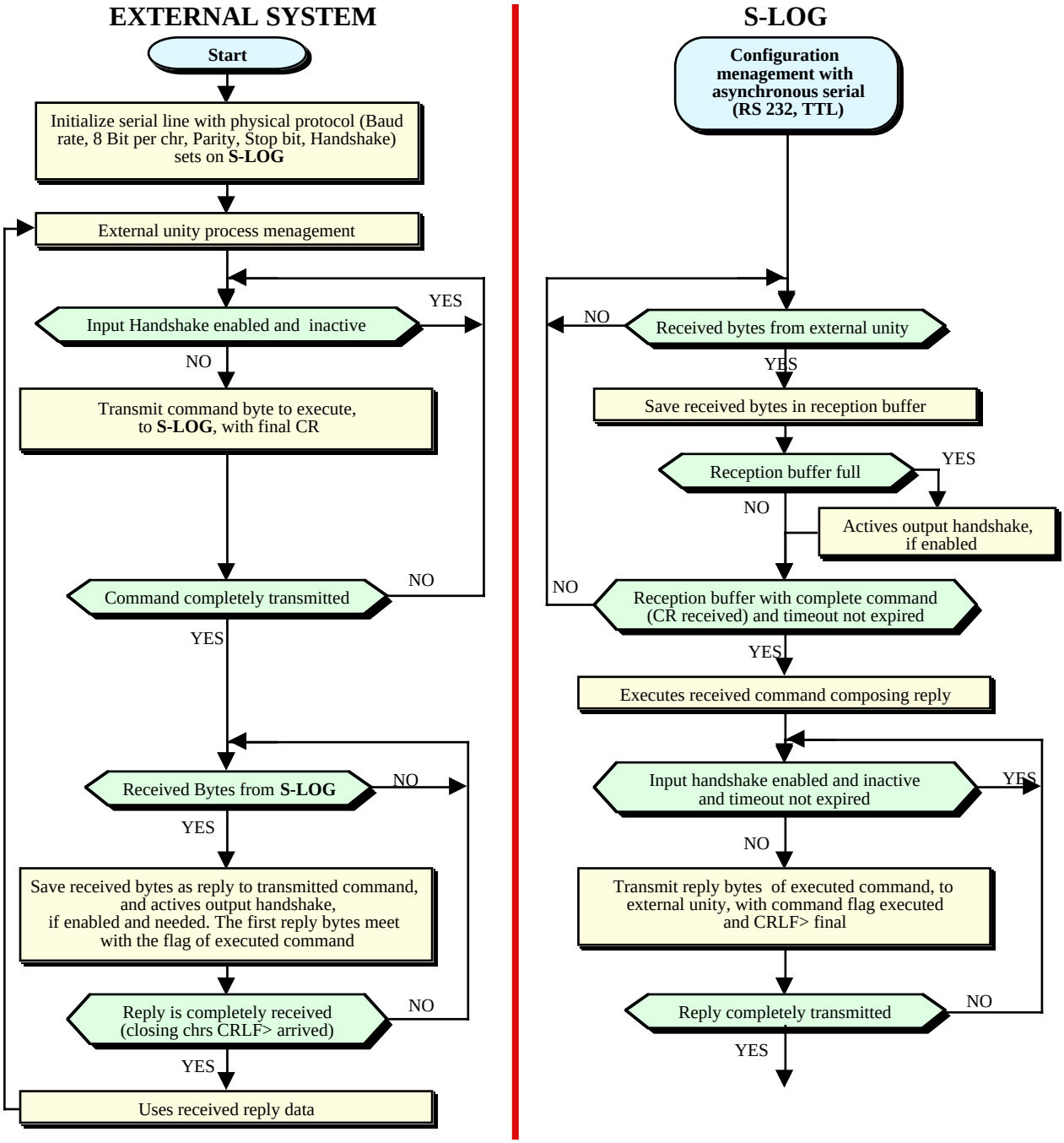


FIGURE 25: ASYNCHRONOUS COMMUNICATION RS 232, TTL FOR CONFIGURATION

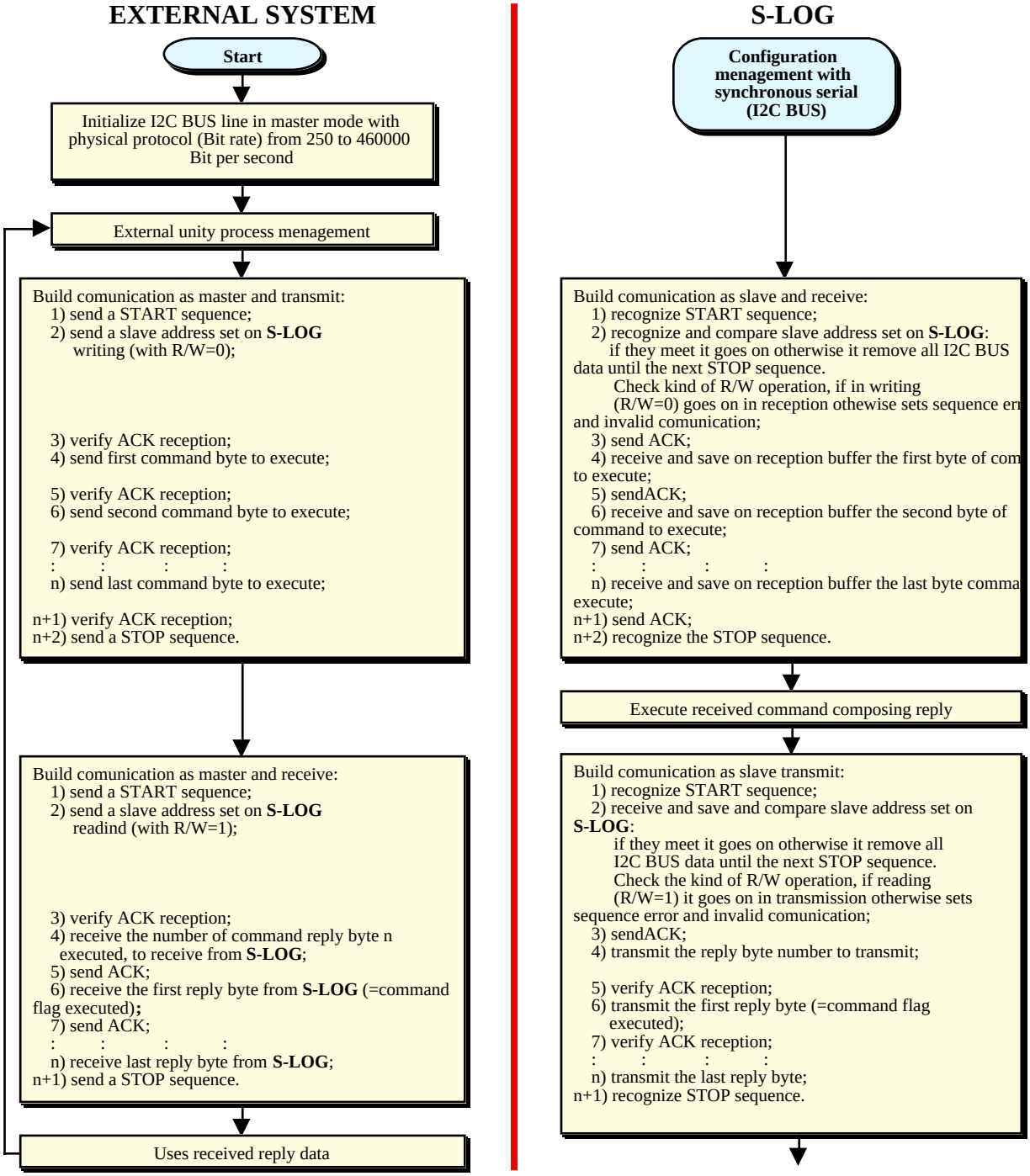


FIGURE 26: SYNCHRONOUS COMMUNICATION I2C BUS FOR CONFIGURATION

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 modeis selected by J1 in 2-3 position as described in OPERATIVE SELECTION MODE paragraph and allows to user to saveon 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 provides data to save and can actives some saving finctions previous 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.

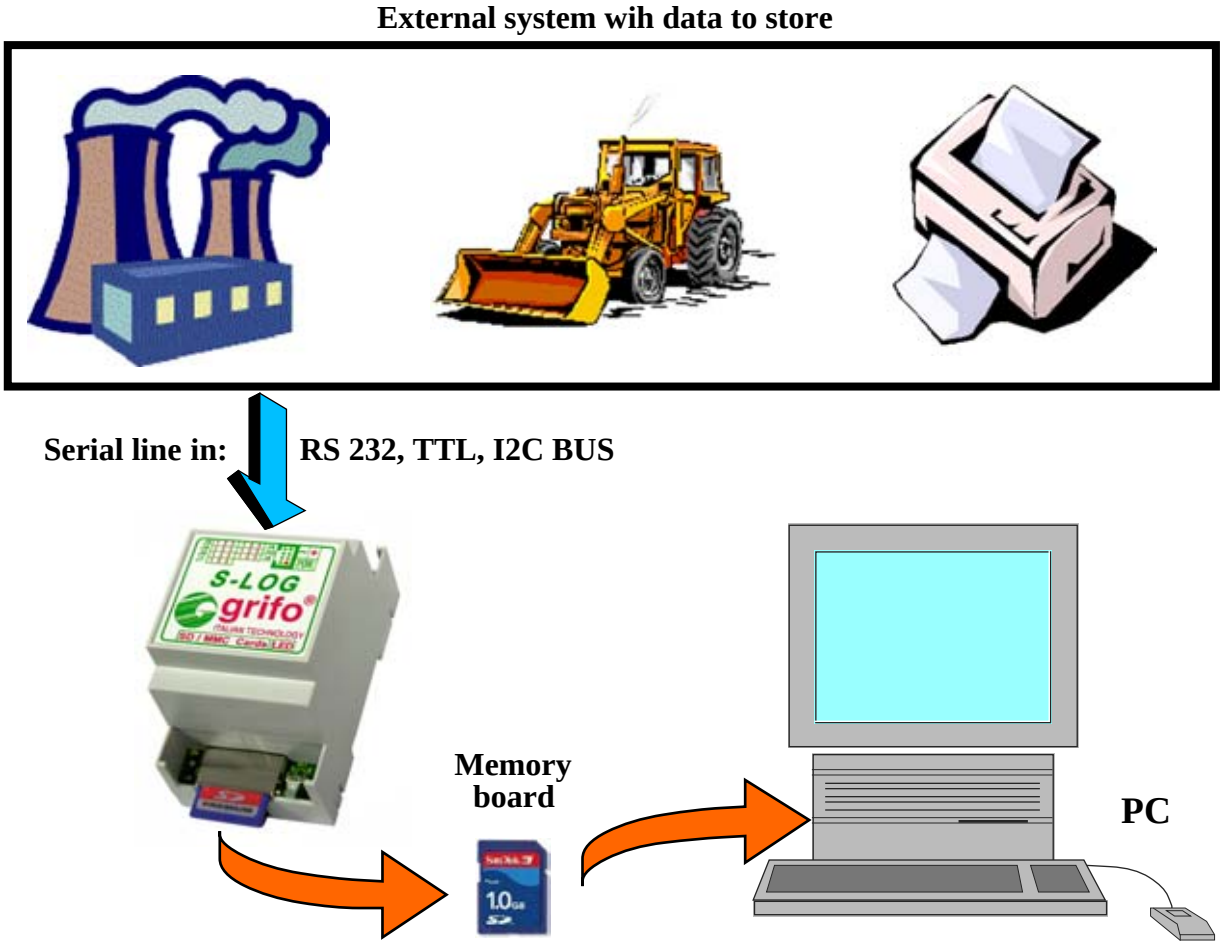


FIGURE 27: RECEPTION AND SAVING DATA MODE

Data are saved on memory card with format set from external system and can be easily copied, registered, woked, printed, etc. by a normal PC, provided of an interface of memory cards. To semplify memory cards menagement, 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 paragraph are described communication protocols (electric, physical, logic) used by this 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 dipends 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 connctions 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 comunication 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.

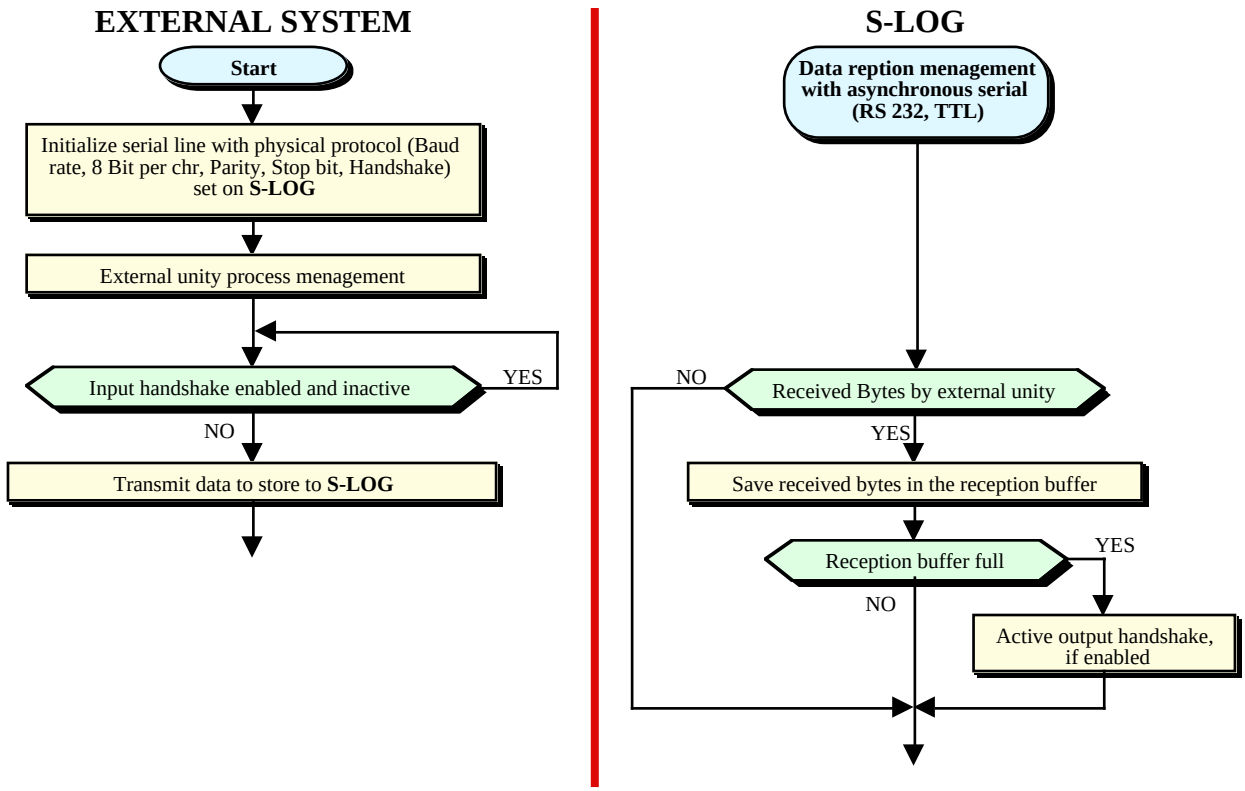


FIGURE 28: ASYNCHRONOUS COMMUNICATION RS 232, TTL FOR DATA SAVING

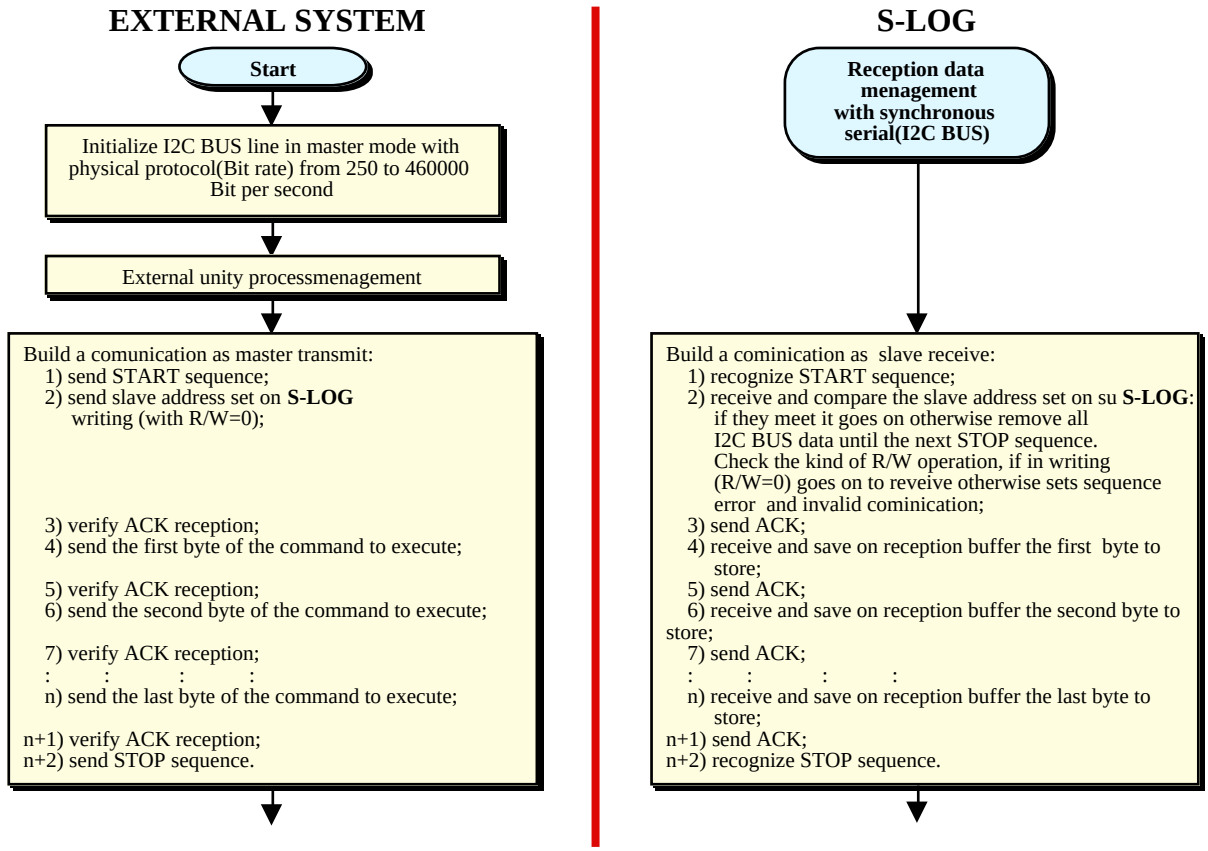


FIGURE 29: SYNCHRONOUS COMMUNICATION I2C BUS FOR DATA SAVING

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 = OVEN1\OUTPUT\PRNOVEN.TXT
  duration = N (none)
  Than in file OVEN1\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 **yyyymmdd.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:

duration	->	file name
hour	->	yyyymmdd.hh
day	->	yyyymmdd.00
month	->	yyyymm01.00
year	->	yyyy0101.00

  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 (code19=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.
f the memory card is valid (see CN2 paragraph -MEMORY CARD CONNECTOR) are performed operations described in following flow chart:

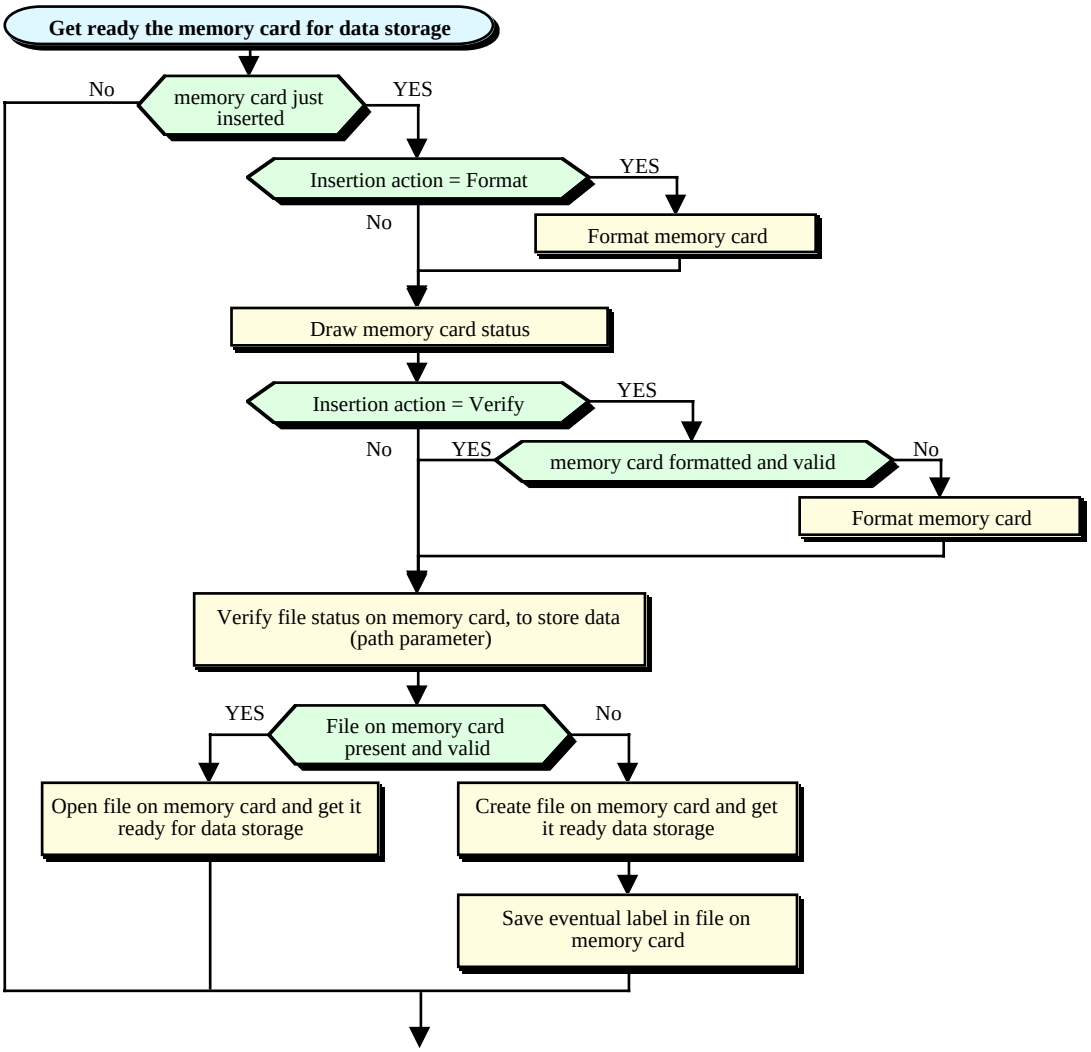


FIGURE 30: MEMORY CARD INSERTION

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.

DEMO PROGRAMS

For **S-LOG** module are available demo programs both in source level and executable level; these programs can be used without no changes for a first check out of the product and than modified, or used in pieces, to sodisfy 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 it 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 comminication 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 descripted 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 semplify the management, the first dara 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 comunication 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 BUSnetwork connection perform configuration described in paragraph CN1 - SERIAL COMINICATION 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 waithing during the communication there is the **tout** work parameter that ensures that **S-LOG** is always operative. The **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 corrispondence to any configuration command in communication, the **S-LOG** actives a counter of elapsed time and during the waithing 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 comunication event, so if **S-LOG** has to waith 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.

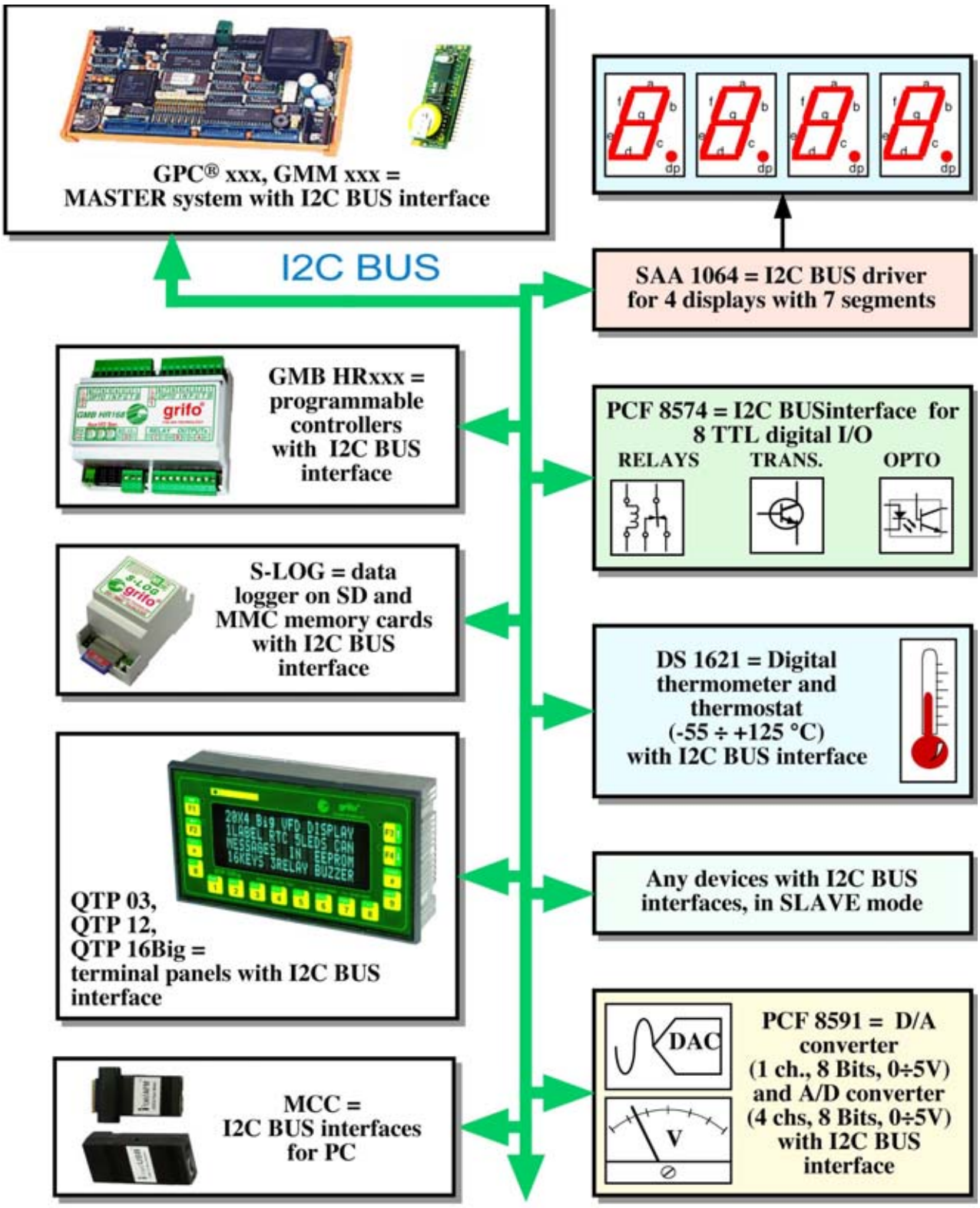


FIGURE 31: NETWORK OF DIFFERENT I2C BUS DEVICES

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 comands and associated parameters are coded with rules already described in LOGIC CONFIGURATION PROTOCOL paragraph and sumarized 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 thefirst 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> (13,10,62= 0DH,0AH,3EH) -> 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 of response**> -> at the begin of response with synchronous communication, before 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 sumarizing 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	or
49 49 48 57 48 55 32 52 46 50	or
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:

Parameter	*dec. weight*	*Hex Weight*		*Description*
general status.00	00001	0001	->	Communication overflow error.
general status.01	00002	0002	->	Parity error in asynchronous line.
general status.02	00004	0004	->	Statu error in synchronous line.
general status.03	00008	0008	->	Invalid communication sequence error.
general status.04	00016	0010	->	RTC access error.
general status.05	00032	0020	->	Invalid configuration error (set work parameter with default values).
general status.06	00064	0040	->	Not recognizsd command error.
general status.07	00128	0080	->	Command parameters error.
general status.08	00256	0100	->	Memory card present.
general status.09	00512	0200	->	Memory card write protect (WP selector enabled).
general status.10	01024	0400	->	Configuration mode active(J1 in 1-2).
general status.11	02048	0800	->	Elapsed Timeout of communication.
general status.12	04096	1000	->	Not used.
general status.13	08192	2000	->	Not used.
general status.14	16384	4000	->	Not used.
general status.15	32768	8000	->	Not used.
disk status.00	00001	0001	->	Disk initialization error.
disk status.01	00002	0002	->	Disk formatting error.
disk status.02	00004	0004	->	Creating folder error in data path file.
disk status.03	00008	0008	->	Change folder error of data path file.
disk status.04	00016	0010	->	Opening data file error.
disk status.05	00032	0020	->	Writing data file error.
disk status.06	00064	0040	->	Flushing buffer error on data file.
disk status.07	00128	0080	->	Writing on protect disk error.
disk status.08	00256	0100	->	Reception data and following saving error with not present disk.
disk status.09	00512	0200	->	Closing data file error.
disk status.10	01024	0400	->	Not used.
disk status.11	02048	0800	->	Not used.
disk status.12	04196	1000	->	Not used.
disk status.13	08192	2000	->	Not used.
disk status.14	16384	4000	->	Not used.
disk status.15	32768	8000	->	Not used.

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 or
49 55 57 50 32 49 50 56 or
31 37 39 32 20 31 32 38 Hex

SET REAL TIME CLOCK

Command:	***T gg/mm/aaaa oo:mm:ss***	***Risposta:***
Command Dec:	***84 32 dd 47 mm 47 yyyy 32 hh 58 mm 58 ss***	***Response Dec:***
Command Hex:	***54 20 dd 2F mm 2F yyyy 20 hh 3A mm 3A ss***	***Response Hex:***

The command set the real time clock of **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 rangeof validity:

Parameter	*Range*		*Means*
dd	1÷31	->	Day
mm	1÷12	->	Month
yyyy	0÷9999	->	Year
hh	0÷23	->	Hours
mm	0÷59	->	Minutes
ss	0÷59	->	Seconds

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 fiel data too.
Note. Because of the year with four numbers managed by clock has to be always valid, it is necessary to provide the set clock command, once every 4 years.

GET REAL TIME CLOCK

Command:	***t***	***Response:***	***dd/mm/yyyy hh:mm:ss***
Command Dec:	***116***	***Response Dec:***	***dd 47 mm 47 yyyy hh 58mm 58 ss***
Command Hex:	***74***	***Response Hex:***	***dd 2Fmm 2F yyyy hh 3A mm 3Ass***

Are reposrted 6 data, grouped in two parameters, that meet temporal parameters with current date and time, get from the clock of **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:	***C com.line***	***Response:***
Command Dec:	***67 32 com.line***	***Response Dec:***
Command Hex:	**43 20 com.line**	**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 **com.line parameter**:

Value		*Means*
R (82=52H)	->	asynchronous communication line(RS 232,TTL)
I (73=49H)	->	synchronous communication line (I2C BUS)

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:	***c***	***Response:***	***com.line***
Command Dec:	***99***	***Response Dec:***	***com.line***
Command Hex:	**63**	**Response Hex:**	**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.

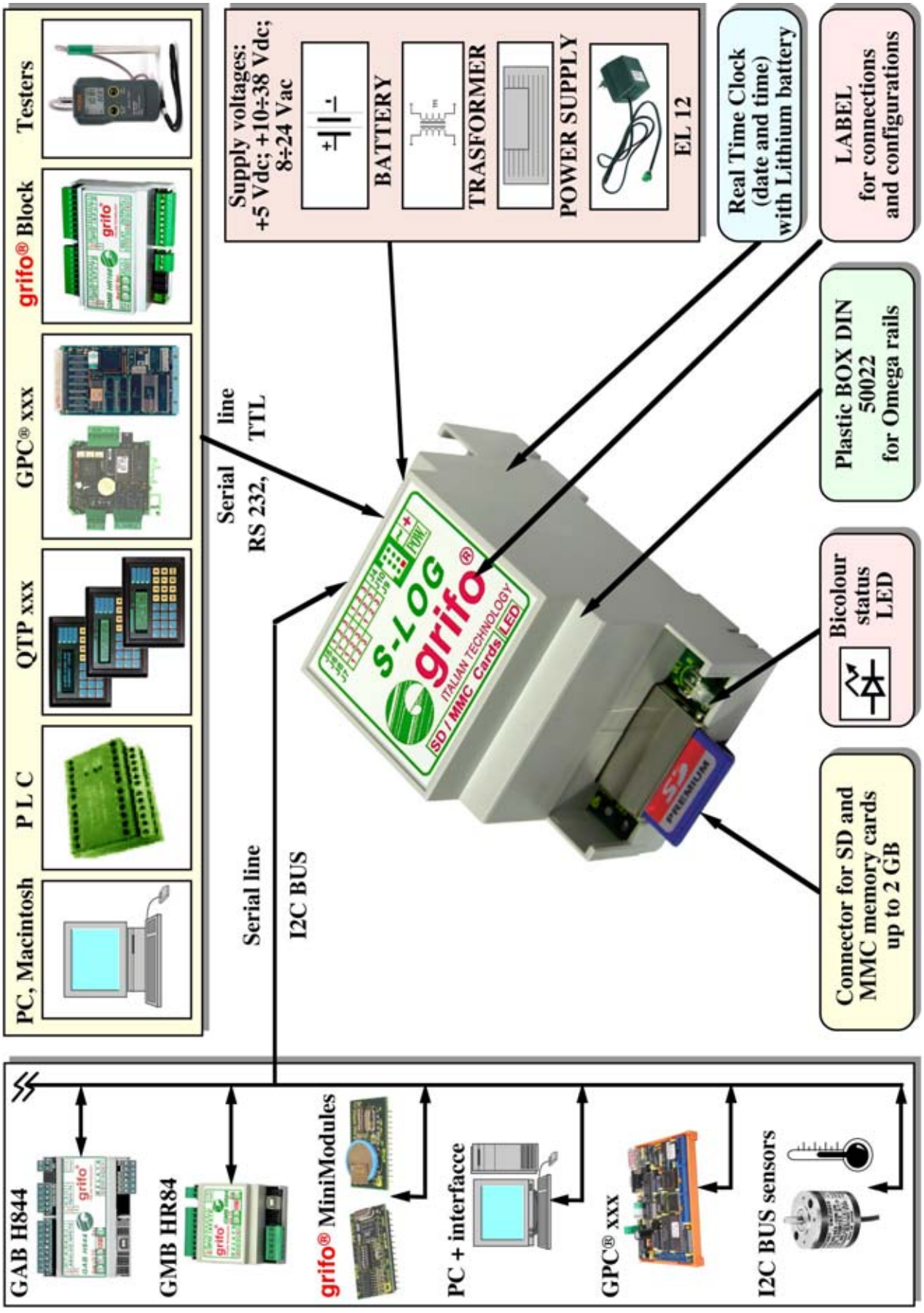


FIGURE 33: AVAILABLE CONNECTIONS DIAGRAM

SET BAUD RATE

Command:	***R baud***	***Response:***	
Command Dec:	***82 32 baud***	***Response Dec:***	
Command Hex:	***52 20 baud***	***Response Hex:***	

The commans set the speed for serial asynchronous line RS 232, TTL used by **S-LOG** for the communication woth the external system, with the passed parameter.
Following there are the description of availabel valueds for **baud** parameter:

Value		*Means*
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

FIGURE 35: BAUD RATE 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.

GET BAUD RATE

Command:	***r***	***Response:***	***baud***
Command Dec:	***114***	***Response Dec:***	***baud***
Command Hex:	***72***	***Response Hex:***	***baud***

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.

SET PARITY

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

The commans set the parity for serial asynchronous line RS 232, TTL used by **S-LOG** for the communication woth the external system, with the passed parameter.
Following there are the description of availabel valueds for **parity** parameter:

Values		*Means*
N (78=4EH)	->	None parity
E (69=45H)	->	Parity even
O (79=4FH)	->	Parity odd

FIGURE 36: PARITY 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.

GET PARITY

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

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

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.
Following there are the description of availabel valueds for **stop** parameter:

Value		*Means*
1 (49=31H)	->	1 stop bit
2 (50=32H)	->	2 stop bit

FIGURE 37: STOP 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.

GET STOP BIT

Command:	***s***	***Response:***	***stop***
Command Dec:	***115***	***Response Dec:***	***stop***
Command Hex:	***73***	***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***	***Response:***	
Command Dec:	***72 32 handshake***	***Response Dec:***	
Command Hex:	***48 20 handshake***	***Response Hex:***	

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

Value		*Means*
N (78=4EH)	->	None handshake
H (72=48H)	->	Handshake hardware (RTS, CTS)
S (83=53H)	->	Handshake software (XON, XOFF)
R (82=52H)	->	Handshake software repeated

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***	***Response:***	***handshake***
Command Dec:	***104***	***Response Dec:***	***handshake***
Command Hex:	***68***	***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***	***Response:***	
Command Dec:	***65 32 sla.add***	***Response Dec:***	
Command Hex:	***41 20 sla.add***	***Response Hex:***	

The commans set the slave address for serial asynchronous line RS 232, TTL used by **S-LOG** for the communication woth the external system, with the passed parameter.
Available values to **sla.add** parameter are all even numeric vaues 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 fo range value not valid too: the external system has to guarantee the validity.

GET SLAVE ADDRESS

Command:	***a***	***Response:***	***sla.add***
Command Dec:	***97***	***Response Dec:***	***sla.add***
Command Hex:	***61***	***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***	***Response:***	
Command Dec:	***79 32 tout***	***Response Dec:***	
Command Hex:	***4F 20 tout***	***Response Hex:***	

The commans set the max. time to wath from **S-LOG** for communication event from the external system, with the passed parameter.
Following there are the description of availabel valueds for **tout** parameter:

Value		*Means*
0 (00H)	->	No waithing
1÷250 (01÷FAH)	->	Waithing 20÷5000 millseconds
255 (FFH)	->	infinite waithing

FIGURE 39: TIMEOUT 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.
The parameter writes the timeout in twenties of milliseconds, with two exeption: with value 0 (00H) the **S-LOG** don't waith the communication event while with 255 value (FFH) waith 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***	***Response:***	***tout***
Command Dec:	***111***	***Response Dec:***	***tout***
Command Hex:	***6F***	***Response Hex:***	***tout***

A parameter is returned a relative parameter to the max. waithing 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***	***Response:***	
Command Dec:	***70 32 path***	***Response Dec:***	
Command Hex:	***46 20 path***	***Response Hex:***	

The command set the file name of saving data, woth pathname, with passed parameter. This file can be created on memory card of **S-LOG**, as described in OPERATIONS ON MEMORY CARD paragrph,e 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 withou 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 paragrph.
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***	***Response:***	***path***
Command Dec:	***102***	***Response Dec:***	***path***
Command Hex:	***66***	***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 WIEV

SET DURATION FILE

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

The commans set the saving data file duration wrute on **S-LOG**, with the passed parameter.
Following there are the description of availabel valueds for **duration** parameter:

Value		*Means*
N (78=4EH)	->	None duration
H (72=48H)	->	One hour duration file.
D (68=44H)	->	One day duration file.
M (77=4DH)	->	One month duration file.
Y (89=59H)	->	One year duration file.

FIGURE 41: DURATION 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.
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 **yyyymmdd.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 temporaly 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 from12:00:00 to 12:59:59 of 22/08/ 2008 day.

GET DURATION FILE

Command:	***d***	***Response:***	***duration***
Command Dec:	***100***	***Response Dec:***	***duration***
Command Hex:	***64***	***Response Hex:***	***duration***

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***	***Rescponse:***	
Command Dec:	***73 32 action***	***Response Dec:***	
Command Hex:	***49 20 action***	***Response Hex:***	

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:

Value		*Means*
N (78=4EH)	->	None action
V (86=56H)	->	Verify if the card is formatted and format it
F (70=46H)	->	Always format iserted memory card.

FIGURE 42: ACTION 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.
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

Command:	***i***	***Response:***	***action***
Command Dec:	***105***	***Response Dec:***	***action***
Command Hex:	***69***	***Response Hex:***	***action***

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

Command:	***G time***	***Response:***	
Command Dec:	***71 32 time***	***Response Dec:***	
Command Hex:	***47 20 time***	***Response Hex:***	

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:

Value		*Means*
0 (00H)	->	Recognizing group time empty, that is each received data meet the group.
1÷250 (01÷FAH)	->	Recognizing group time from 20 to 5000 millseconds
255 (FFH)	->	Recognizing group time disabled.

FIGURE 43: GROUP TIME PARAMETER

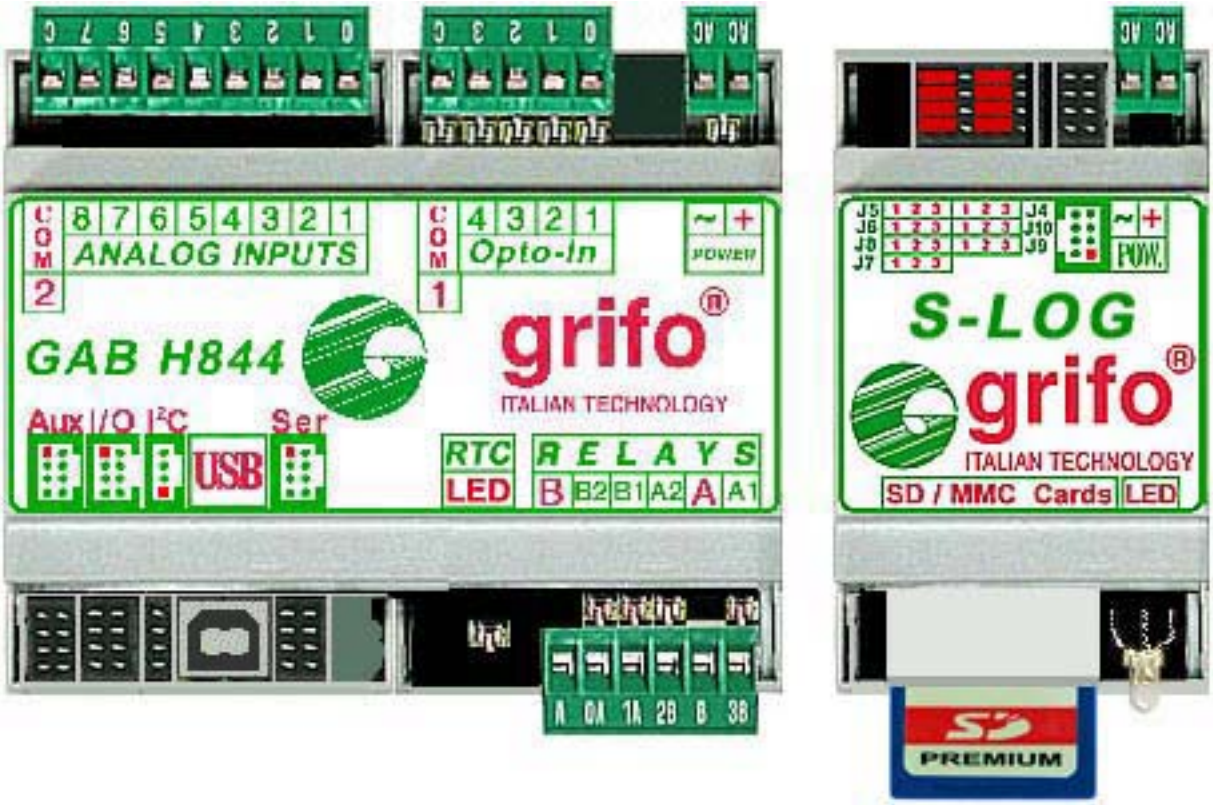


FIGURE 44: DATA SAVING APPLICATION WITH GAB H844

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 minimun 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 exeption : 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

Command:	***g***	***Response:***	***time***
Command Dec:	***103***	***Response Dec:***	***time***
Command Hex:	**67**	**Response Hex:**	**time**

A parameter is returns a relative parameter relative to recognizind 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

Command:	***L label***	***Response:***	
Command Dec:	***76 32 label***	***Response Dec:***	
Command Hex:	**4C *20 label***	**Response Hex:**	

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 exeption: some special sequence that allow to format and add attributes and other data. In detail:

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

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\b\d\b\h\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

Command:	***l***	***Response:***	***label***
Command Dec:	***108***	***Response Dec:***	***label***
Command Hex:	***6C***	***Response Hex:***	***label***

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

Commando:	***B prefix***	***Response:***	
Command Dec:	***66 32 prefix***	***Response Dec:***	
Command Hex:	***42 20 prefix***	***Response Hex:***	

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 identifythe 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 sequnces that allow to format and add attibutes and other data. In details:

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

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 **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:

prefix = Print\bdel\b\d\bore\b\h:

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

GET GROUP PREFIX

Command:	***b***	***Risponse:***	***prefix***
Command Dec:	***98***	***Resonse dec:***	***prefix***
Command Hex:	***62***	***Response Hex:***	***prefix***

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.

SET GROUP SUFFIX

Command:	***E suffix***	***Response:***	
Command Dec:	***69 32 suffix***	***Response Dec:***	
Command Hex:	***45 20 suffix***	***Response Hex:***	

The command set the suffix saved by **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 **suffix** parameter meet the string of 30 max. 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 detail:

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

FIGURE 47: SPECIAL SEQUENCE OF SUFFIX 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.
For example, If the **S-LOG** has to save dataperiodically 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\r\n

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

Command:	***e***	***Response:***	***suffix***
Command Dec:	***101***	***Response Dec:***	***suffix***
Command Hex:	**65**	**Response Hex:**	**suffix**

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 supposeto replacea 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 i s 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.

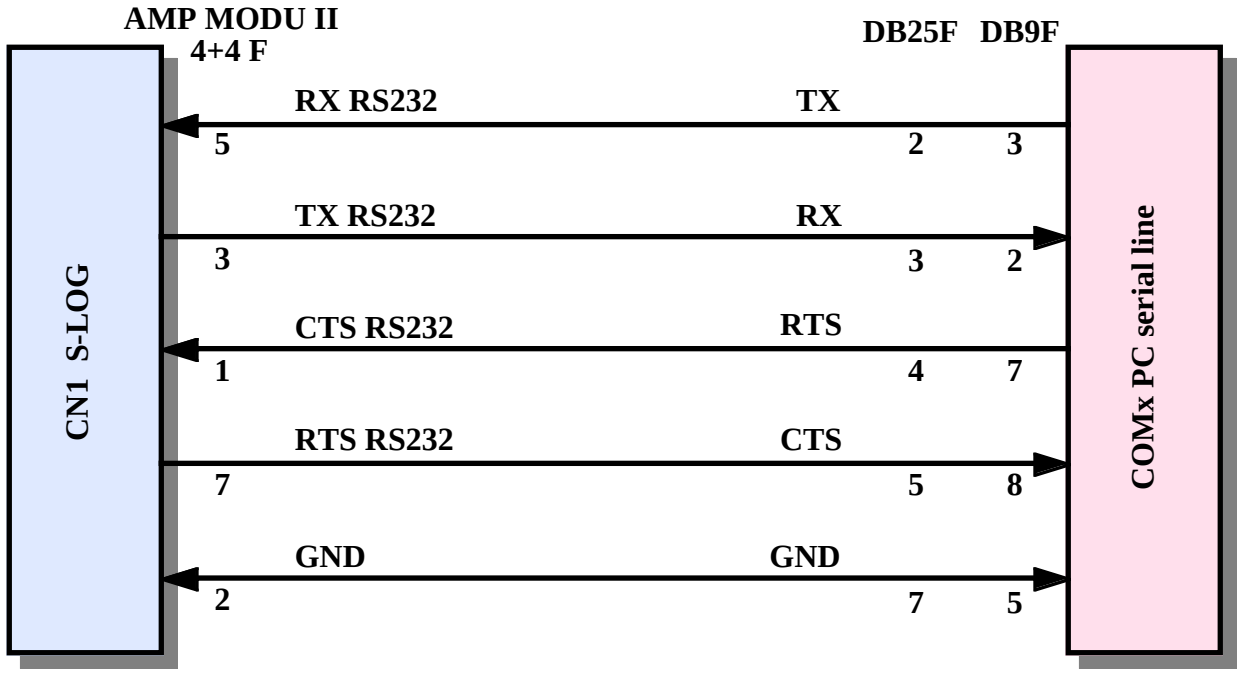


FIGURE 48: RS 232 CONNECTION WITH PC

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 mantain 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.

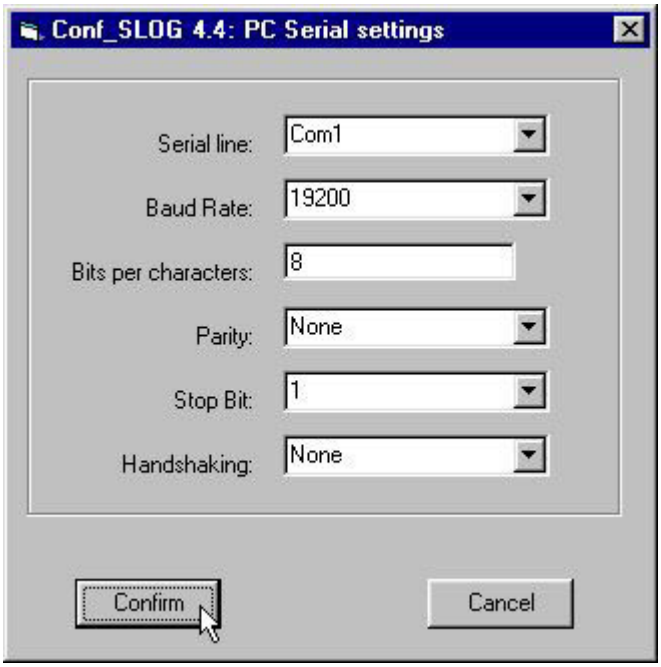


FIGURE 49: PC SERIAL SETTING WITH CONF_SLOG

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.

FIGURE 50: ACQUIRE VERSION WITH CONF_SLOG

B6) Set the kind of serial communication that **S-LOG** has to use to receive data to save, through *Configure* | *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 ti press before the *Preleva configurazioni* button, than selectthe proper settings as described in follow 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.

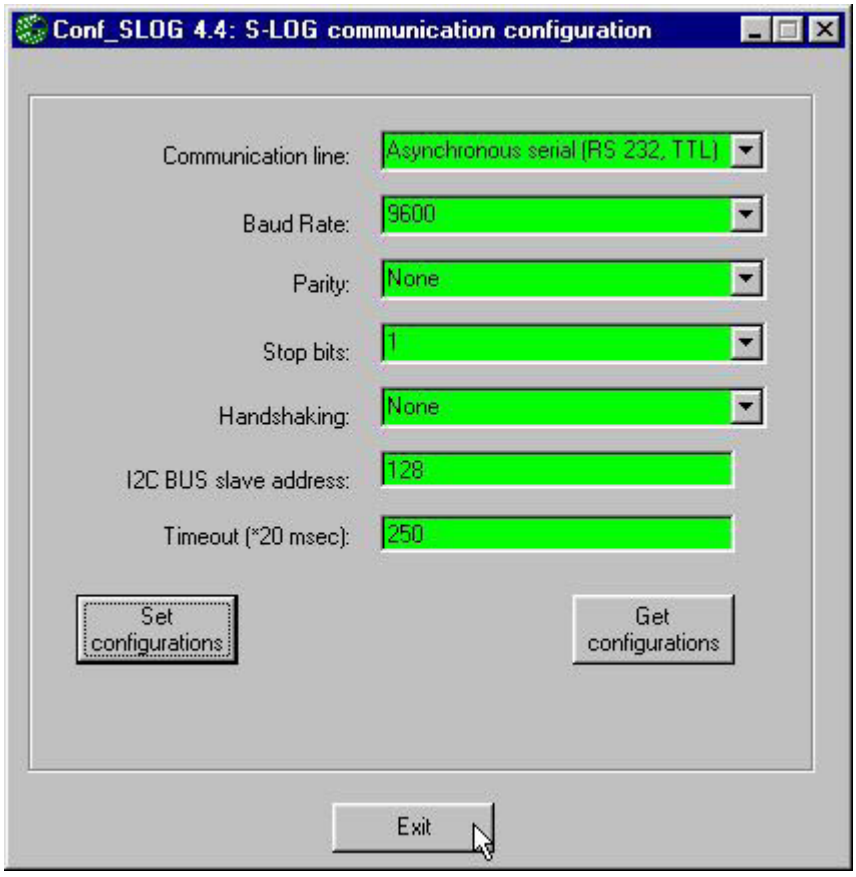


FIGURE 51: COMMUNICATION CONFIGURATION WITH CONF_SLOG

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 messagges 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.

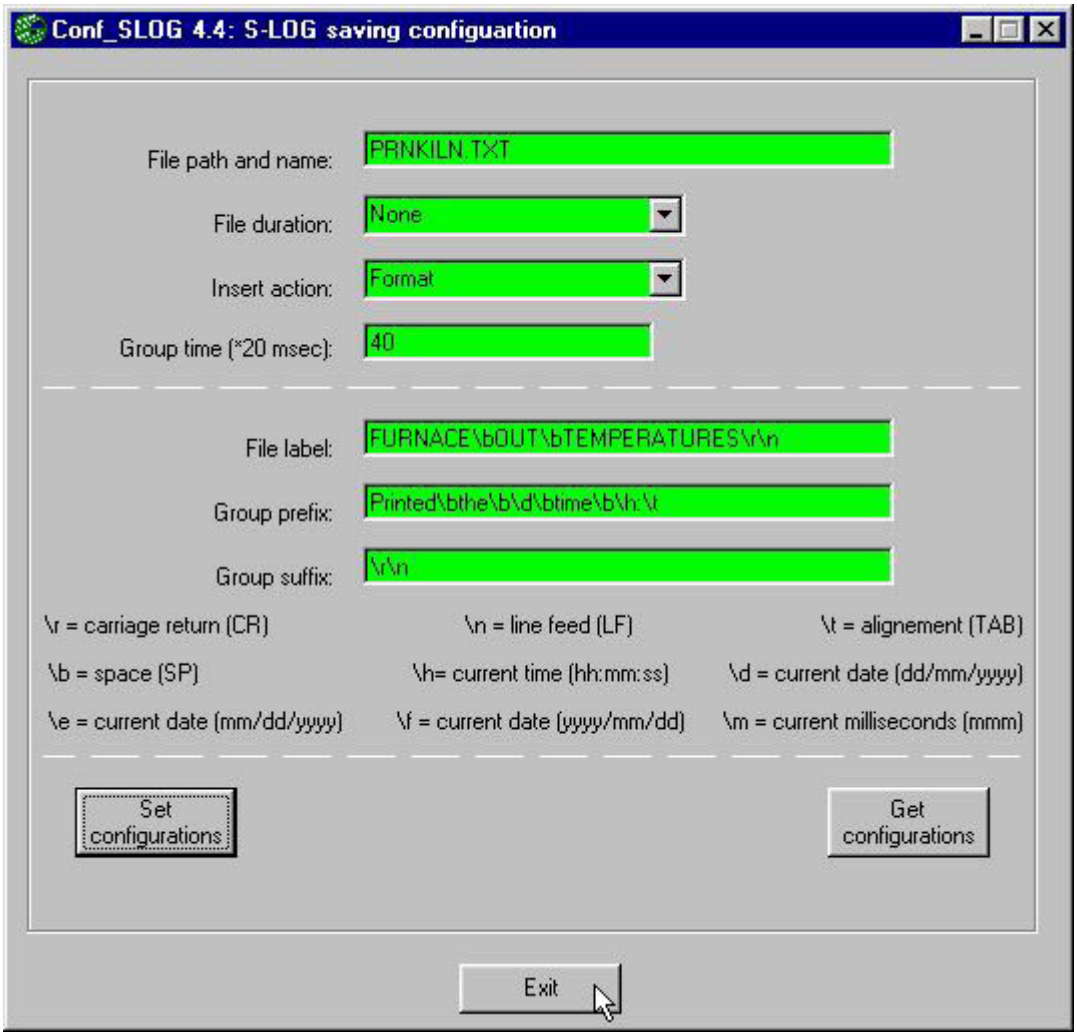


FIGURE 52: SAVING CONFIGURATION WITH CONF_SLOG

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 necessaries 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 comunicated data with differnce 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 configurated 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 configuratiohn 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 finel user's requirements.

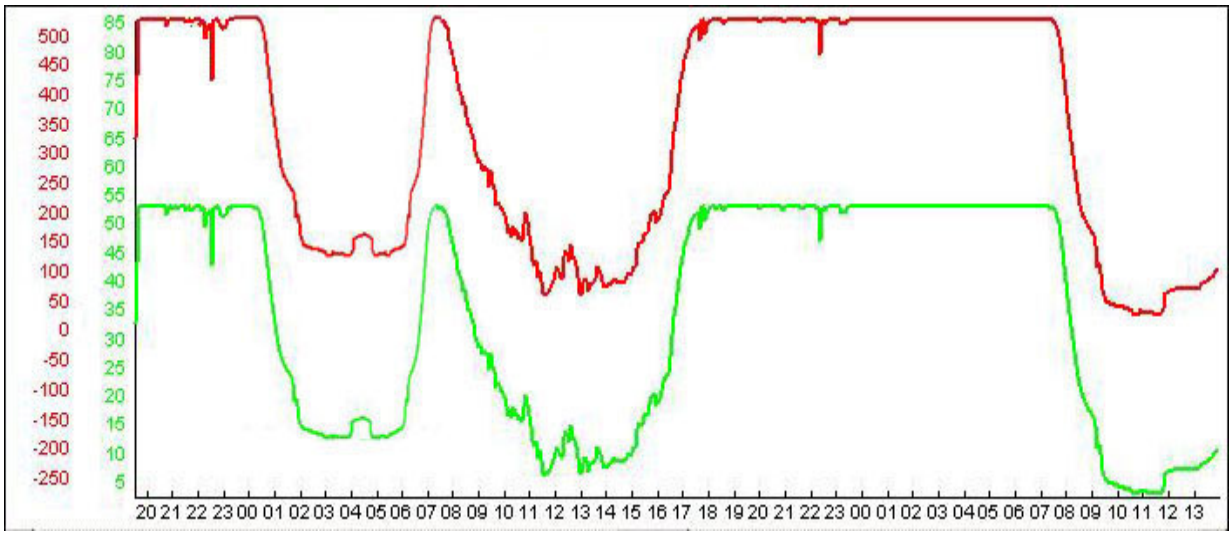


FIGURE 53: GRAPHIC WITH SAVED DATA BY S-LOG

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 nmemonic 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.

1Command	Comunication from main system to S-LOG	Comunication from S-LOG to main system
Clear errors	Z	-
Get status	z	general status disk status
Get versions	v	hw ver. fw ver.
Set Real Time Clock	T dd/mm/yyyy hh:mm:ss	-
Get Real Time Clock	t	gg/mm/aaaa oo:mm:ss
Set Comunication line	C com.line	-
Get Comunication line	c	com.line
Set Baud Rate	R baud	-
Get Baud Rate	r	baud
Set Parity	P parity	-
Get Parity	p	parity
Set Stop Bit	S stop	-
Get Stop Bit	s	stop
Set Handshake	H handshake	-
Get Handshake	h	handshake
Set Slave Address	A sla.add	-
Get Slave Address	a	sla.add
Set Comunication Timeout	O tout	-
Get ComunicationTimeout	o	tout

FIGURE A1: CONFIGURATION COMMANDS TABLE (1 OF 2)

Command	Comunication from main system to S-LOG	Comunication from S-LOG to main system
Set File	F pathname	-
Get File	f	pathname
Set file duration	D duration	-
Get file duration	d	duration
Set Insertion action	I action	-
Get insertion action	i	action
Set Group Time	G time	-
Get Group Time	g	tempo
Set File Label	L label	-
Get File Label	l	label
Set Group Prefix	B prefix	-
Group Group Prefix	b	prefix
Set Group Suffix	E suffix	-
Get Group Suffix	e	suffix

FIGURE A2: CONFIGURATION COMMANDS TABLE (2 OF 2)

APPENDIX B: DEFAULT CONFIG., OPTIONS, ACCESSORIES

In corrispondence 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.

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

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.

JUMPER		*FUNCTION*
J1	position 2-3	Selects the configuration mode
J2	connected	Connect BT1 battery to back up circuitry.
J4	position 1-2	Does not connect power supply +5 Vdc to CN1 connector.
J5÷J10	position 1-2	Configure serial line asynchronous on CN1 for the RS232 electrical standard

FIGURE B2: DEFAULT JUMPERS CONFIGURATION TABLE

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:

OPTION	*DESCRIPTION*
.SW	Power supply switching section with wide range input

FIGURE B3: OPTIONS TABLE

All these options are described in the pragraphs 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.

n 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;

FIGURE B5: AMP4.CABLE CONNECTION ACCESSORY

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

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;

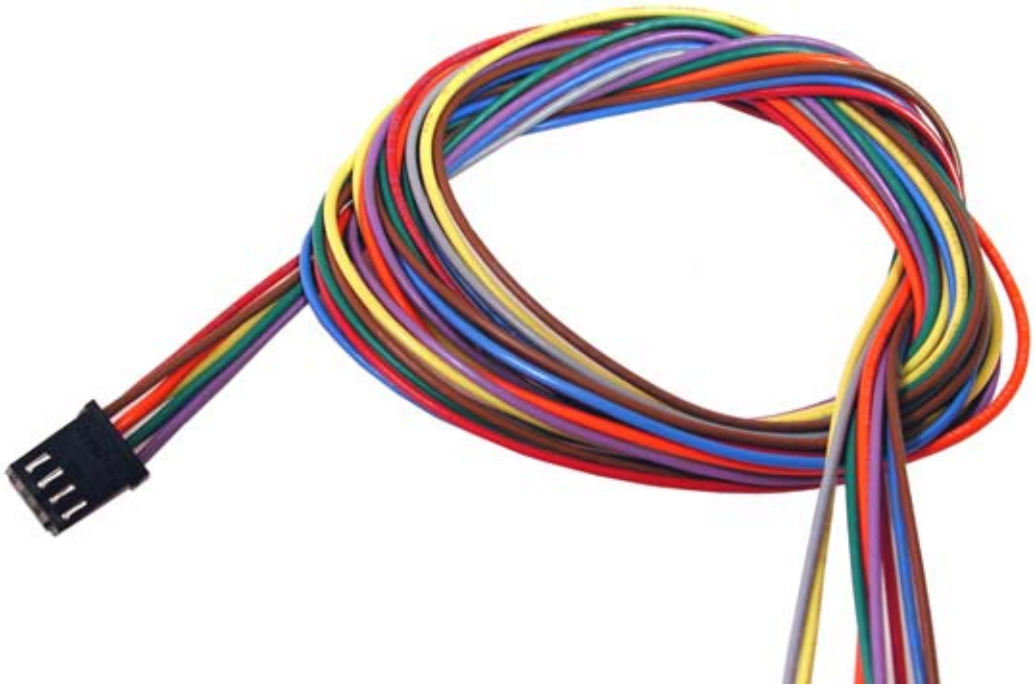


FIGURE B7: AMP8.CABLE CONNECTION ACCESSORY

- **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.