GM T

gri® ModBUS Telecontrol

USER MANUAL
GMT is a "complete logic protocol for serial communication" developed for the wide range of grifo® industrial cards. This protocol is based on the famous MODBUS standard that, thanks to its world wide diffusion, ensures the employment either in old and new applications. The GMT package is the result of numerous experiences, collected in many years, during the development of applications in the industrial environments. In this field the informations must be exchanged between intelligent systems through simple and cheap connections.

The use of GMT deletes, or at least reduces considerably, the development time of a telecontrol application. Normally the long list of high level functions and the complete groups of addressable hardware resources is sufficient for the standard demands.

One of the most important features of GMT is that it is a ready to use product: the user is not forced to know the used communication protocol. Thanks to the utility programs and the libraries provided with the product, it is possible to set up, test and use the communication network in few minutes.
IMPORTANT

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

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

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

Trade Marks

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

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

VERSION ................................................................................................................................... 3

GENERAL INFORMATION ........................................................................................................ 4
DISTRIBUTION .......................................................................................................................... 6
REQUIREMENTS ......................................................................................................................... 8

GMT DESCRIPTION .................................................................................................................. 9
DIFFUSION .................................................................................................................................. 9
MESSAGES FORMAT .................................................................................................................. 10
SLAVE ADDRESS ....................................................................................................................... 10
FUNCTION CODE ....................................................................................................................... 11
FUNCTION DATA ....................................................................................................................... 12
RESOURCES ADDRESSES ......................................................................................................... 12
ERRORS CHECK ......................................................................................................................... 16
EXCEPTION SUPPORTED ............................................................................................................ 17
ELECTRIC AND PHYSIC PROTOCOL ....................................................................................... 17
CONFIGURATION OF GMT REMOTE SLAVE ........................................................................... 18
AVAILABLE IMPLEMENTATION .................................................................................................. 20

FUNCTIONS AVAILABLE IN GMT ............................................................................................ 21
READ COIL STATUS (1=01H) ..................................................................................................... 21
READ INPUT STATUS (2=02H) ................................................................................................... 22
READ HOLDING REGISTERS (3=03H) ....................................................................................... 22
READ INPUT REGISTERS (4=04H) ........................................................................................... 23
FORCE SINGLE COIL (5=05H) ................................................................................................ 23
PRESET SINGLE REGISTER (6=06H) ....................................................................................... 24
FORCE MULTIPLE COILS (15=0FH) ....................................................................................... 24
PRESET MULTIPLE REGISTERS (16=10H) ............................................................................... 25

GMT APPLICATIONS .................................................................................................................. 27

DEMO GMT PROGRAM ............................................................................................................. 31
INSTALLATION, DEINSTALLATION ............................................................................................ 32
EXECUTION AND USE ................................................................................................................. 34
SLAVE CONFIGURATION ............................................................................................................ 35
MANUAL TEST ............................................................................................................................ 36
LIMITS OF DEMO GMT .............................................................................................................. 38

HOW TO START .......................................................................................................................... 39
PRELIMINARY OPERATIONS ....................................................................................................... 39
CONFIGURATION AND FIRST USE ............................................................................................ 40

APPENDIX A: ALPHABETICAL INDEX .......................................................................................... A-1
FIGURES INDEX

FIGURE 1: GMT CONNECTION DIAGRAM ........................................................................................................................................... 7
FIGURE 2: FUNCTION AVAILABLE IN GMT .......................................................................................................................................... 11
FIGURE 3: RESOURCES ADDRESSES TABLE - PART 1 OF 2 .................................................................................................................. 13
FIGURE 4: RESOURCES ADDRESSES TABLE - PART 2 OF 2 .................................................................................................................. 14
FIGURE 5: FEATURES OF THE AVAILABLE IMPLEMENTATIONS ........................................................................................................... 20
FIGURE 6: I/O PLACEMENTS ON HOUSE APPLICATION ..................................................................................................................... 27
FIGURE 7: I/O DESCRIPTIONS ON HOUSE APPLICATION ...................................................................................................................... 28
FIGURE 8: I/O CONNECTION ON HOUSE APPLICATION ......................................................................................................................... 29
FIGURE 9: REQUIREMENTS WINDOWS OF DEMO GMT .......................................................................................................................... 32
FIGURE 10: REMOVE DEMO GMT SELECTION WINDOW ......................................................................................................................... 33
FIGURE 11: MAIN WINDOW OF DEMO GMT ........................................................................................................................................ 34
FIGURE 12: SLAVE CONFIGURATION WINDOW .................................................................................................................................. 35
FIGURE 13: RESULT WINDOW OF SLAVE CONFIGURATION .................................................................................................................... 36
FIGURE 14: ADD SLAVE WINDOW ......................................................................................................................................................... 37
FIGURE 15: MANUAL TEST WINDOWS OF SLAVE .................................................................................................................................. 37
FIGURE 16: SCAN RATE WINDOW SETTING ...................................................................................................................................... 38
FIGURE 17: DEMO GMT USE EXAMPLE .............................................................................................................................................. 39
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/s are not provided with any kind of ESD protection. Many pins of the card are directly connected to their respective pins of on board's components and these last are sensitive to electrostatic noises. So personnel who handles the product/s is invited to take all necessary precautions that avoid possible damages caused by electrostatic discharges.

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

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

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

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

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

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

All trademarks listed in this manual are copyright of the relative manufacturers.
This handbook make reference to version 1.1 of GMT firmware and following ones. The validity of the information contained in this manual is subordinated to the version number of the used firmware and the user must always verify the correct correspondence between the notations. The version number is reported on the received CD label and it can be directly acquired from the device, through a proper function.

Normally the GMT 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 phase.

In addition, this manual reports information about other different programs that are integrant parts of GMT: each one of these programs has its own version number that is specifically described when it is necessary. Finally also the hardware is provided of his version as indicated in the related technical manuals.

When the user require technical assistance it is really important that he provides a description of the problem plus the version numbers of the used hardware, firmware and software.

Like any products, also GMT is continuously changed and improved to satisfy completely the new requirements of the users and correct the discovered problems and bugs. Here follows a brief description of the changes made to the package according to the version number:

Ver. 1.0 -> First version for internal development and test.

Ver. 1.1 -> First released version.

Any eventual improvement or addition the user thinks may be interesting, can be suggested by contacting directly grifo®.
GENERAL INFORMATION

This manual gives all the software and hardware information to allow the user to take the maximum advantage of GMT (grifo® MODBUS Telecontrol) features.

The following convention are used in all the manual:

application: it is the system to develop, complete of hardware, firmware and software;
GMT remote slave: it is the grifo® card, or the group of grifo® cards, provided of GMT that have been selected to develop the application;
master: it is the main system capable to communicate with the GMT slaves, that decide the functionalities of application.

GMT is a "complete logic protocol for serial communication" developed for the wide range of grifo® industrial cards. This protocol is based on the famous MODBUS standard that, thanks to its world wide diffusion, ensures the employment either in old and new applications. The GMT package is the result of numerous experiences, collected in many years, during the development of applications in the industrial environments. In this field the informations must be exchanged between intelligent systems through simple and cheap connections.

In general the MODBUS is a logic protocol that defines the communication rules, in fact it standardizes:

- the format of communicated messages;
- the modalities of serial line holding and release;
- the identification of devices on the network;
- the number and type of executable functions;
- the data encoding technique inside the messages;
- the addressing of hardware resources on the devices;
- the modalities used to check the right communication of messages;
- the handling of errors;
- etc.

but it doesn’t specify the electric and physic protocol. So the user can freely select, for example, the desired Baud Rates, the electric driver used for connection, and so on, to better satisfy its own requirements or to easily fit the features of a communication already developed.

This documentation assumes that the user already knows MODBUS and so this protocol is not here described. For those users that want a deeper knowledge it is suggested to examine the considerable information available on Internet, for example in the web site:

http://www.modbus.org/

The use of GMT deletes, or at least reduces considerably, the development time of a telecontrol application. Normally the long list of high level functions and the complete groups of addressable hardware resources is sufficient for the standard demands. If the application needs additional functions or resources, they can be directly required to grifo®.

One of the most important features of GMT is that it is a ready to use product: the user is not forced to know the used communication protocol. Thanks to the utility programs and the libraries provided with the product, it is possible to set up, test and use the communication network in few minutes. When the master system inside the telecontrol network is a PC, it can be used the numerous programs that already manages the MODBUS protocol, as the SCADA; the software development tools VisualBasic; Visual C; Delphi; etc.
The most important features of GMT are below summarized:
- Available for numerous grifo® cards.
- New cards can be easily added to the list of GMT remote slaves.
- Management of hardware resources of GMT remote slaves through high level functions, that are transferred on serial line.
- Based on asynchronous serial line.
- Selectable electric protocol among the most diffused standards:
  - RS 232
  - RS 422
  - RS 485
  - Current Loop
- Selectable physic protocol:
  - Baud rate from 1.200 to 38.400 Baud with autodetection
  - Odd, Even or No parity, programmable through specific function
  - 1 or 2 Stop Bit, programmable through specific function
- Standard logic protocol MODBUS ASCII (7 bits for character).
- Management of all the hardware resources of the selected GMT remote slave, that are:
  - Digital input signals
  - Digital output signals
  - Analog input signals
  - Analog output signals
  - Counters signals
  - Pulse width modulation (PWM) signals
  - Non volatile memories (backed SRAM and EEPROM)
  - Real Time Clock (RTC)
- It can be used either on point to point systems or on master slave networks.
- Maximum number of devices connectable on network = 247.
- When network connection is used, the GMT remote devices are always slave.
- Implementation of the MODBUS functions suitable to solve the typical problems of industrial automation:
  - READ COIL STATUS = Returns status of the single digital outputs
  - READ INPUT STATUS = Returns status of single digital inputs
  - READ HOLDING REGISTERS = Returns management registers value
  - READ INPUT REGISTERS = Returns status of digital inputs
  - FORCE SINGLE COIL = Sets status of single digital output
  - PRESET SINGLE REGISTER = Sets value on a single management register
  - FORCE MULTIPLE COILS = Sets status of several digital outputs
  - PRESET MULTIPLE REGISTERS = Sets values on several management registers
- Management of most important MODBUS exceptions:
  - Illegal function code in the command for the slave
  - Illegal resources address in the command for the slave
  - Illegal data values in the command for the slave
- Possibility to manage all the hardware resources, also through proper management registers (holding registers).
- Resource addresses maintained for all the GMT remote slave: by using a different hardware the functionality is the same.
- Associate some resources to the device status:
  - GMT firmware version
  - Identification code of GMT remote slave
Address of slave
Delay for the response
etc.

- **Reduced execution time**: many of the received commands is executed in less than 50 µsec.
- Long list of communication programs for PC, either under the form of library, DLL, OCX, etc., or complete programs. These can be developed and supplied directly by grifo® or other different companies.
- Powerfull and complete **demo program** that configures the GMT remote slaves and then allows the managements of all the available resources.
- Wide documentation and rich list of **examples** both in source and executable format.
- The GMT package is composed by software on CD, a user documentation and a specific firmware saved on the selected card.
- It has **no license** nor additional costs. The user is free to develop all the applications that he requires.

Considering the natural evolution of software packages, please always search for poddible documentation files (*.TXT, *.DOC, *.HLP) inside the received CD. These files report all the additions, changes, and improvements made to all the software package not yet reported in the manual: if these files are present, they must be examined, printed and added to this manual. Furthermore it is suggested the reading of technical manuals of the hardware cards used in the GMT remote slave, where there are information about use and connection of the same cards.

**DISTRIBUTION**

The GMT package is distribuited as two physical elements that are supplied after an order or a specific request. The first element is the GMT firmware that is already saved on the selected card and the second is a CD that includes the following items:

- the user documentation either in English and Italian language;
- all the GMT implementation currently available for grifo® hardware;
- the example programs developed to communicate with GMT remote slaves;
- the demo program for PC, named Demo GMT.

The customer that receive the GMT must check the presence of the material listed above and then recognize the component that are usefull for his requirements. After he can use these selected components as described in the following chapters of this manual. Some of the items saved on CD, as for example the manuals, can be downloaded from grifo® web sites; it is sufficient to reach the web pages that describe the product, inside the telecontrol subject. Through the internet web site it is also possible to check, after some time from the first order, if new version of GMT have been released or new related products have been developed.
Figure 1: GMT Connection Diagram

- MODBUS
- RS 485

Products:
- GMB HR84
- GMB HR168
- GMB HR244
REQUIREMENTS

Below is described the list of the fundamental materials that are necessary to use GMT package:

a) At least one GMT remote slave ready to use. The ready to use indication means that the selected grifo® card/s must be arranged with all the necessary items that make it works correctly (power supplies, additional options, configurations, wirings, etc.) as described in the relative technical manual.

b) A master system, capable to communicate with MODBUS ASCII protocol, provided of a serial line equal to those available on the GMT remote slave/s prepared at point (a).

c) A serial connection cable that allows the communication between all the systems described in previous points, by using the selected electric protocol. For development of this cable, please use the numerous examples reported in the technical manual of the hardware cards described at point (a).

Moreover it is not necessary but it is warmly suggested:

b) A personal computer capable to execute the Demo GMT program, usefull to configure the remote slaves and to check the functionality of the developed system. This PC must have the following minimum requirements:

- Personal Computer: IBM or compatible
- RAM memory: \( \geq 64 \text{ MBytes} \)
- Hard disk: \( \geq 8 \text{ MBytes free} \)
- Video card: \( \geq 800 \times 600 \text{ pixels, 65536 colours} \)
- Monitor: Colour
- Mouse: Microsoft compatible, correctly managed by used PC and operating system
- Interfaces: One free COMx serial line, correctly managed by used PC and operating system
- Operating system: Windows 98, ME, 2000, XP
- Necessary programs: Net Framework (if not available the installation program of Demo GMT download and install this software package, directly from Microsoft web site.

In the previous description the indication "correctly managed by used PC and operating system" refers to the device in object that must be previously installed. This installation includes both hardware and software configurations as defined by the manufacturing company. In other words the Demo GMT has no dedicated software driver for these devices, but it uses those already available in the operating system.
GMT DESCRIPTION

The GMT, as previously stated, is a communication protocol suitable to solve the typical telecontrol problems of industrial and civil environments. Substantially it is based on one or more remote devices connected either with the field signals, themself and/or others devices, through an asynchronous serial line. This line uses the rules of MODBUS protocol, which is the standard GMT derives from and in detail of MODBUS in ASCII mode, dedicated to a slave device submited to another master device.

The MODBUS protocol defines the format and the modalities of communication between the master that manage the application and the slaves that responds to interrogation. It describes the process that master and slave use to start and stop a communication, the messages exchange, the data encoding inside the message, the errors detection, etc.

The MODBUS is a master slave protocol with the following general features:

- only the master initiates a communication, so the slaves can't automously start a communication but only respond to a received query;
- on the serial line can be connected a single master and a variable number of slave, up to the maximum value of 247;
- the slaves can't communicate directly among themself, but only with the master;
- only one communication at a time can be passed on the line.

In addition to the above features, the communication can have two formats:

- query-response the master send a query to a single slave and this returns its reponse;
- broadcast the master transmit a query directed to all the slave connected on the line and these don't return any response.

In other words a complete communication can include a single query message plus a single response message or only a single broadcast query message.

When a remote slave, that executes the GMT firmware, receive a message firstly it is interpreted, then it is checked for validity, recognized for the destination, executed by using the available hardware resources and finally returns a possible response message.

The MODBUS protocol and thus the GMT, can be used either in point to point connection or multipoints networks; according with selected electric protocol, it can use both full duplex or half duplex communication.

DIFFUSION

The MODBUS protocol is implemented by hundreds of companies on thousands of products. Currently it is really a lingua franca or common denominator between different manufacturer and it can be considered a de-facto standard in multi vendors integration, inside the industry processes. To underline the diffusion we remind that the protocol is open and it is published and documented in many circumstance of public use.

Consequently also the GMT take advantage from this important diffusion and application fields, with some little restrictions, described in the following paragraphs.
MESSAGES FORMAT

Any messages must have a well defined frame to allows the correct communication. The message frame for MODBUS ASCII defines some fields that must be always present either for the master and the slave. These fields are:

- the beginning of message (character ":" = 58 = 3AH);
- the slave address interested by the communication (see dedicated paragraph that follow);
- the function code that must be executed, or has been executed (see dedicated paragraph that follow);
- the data of the function that must be exchanged (see dedicated paragraphs that follow);
- the errors check (see dedicated paragraph that follow);
- the end of message (characters "CR" = 13 = 0DH and "LF" = 10 = 0AH).

All the fields must be encoded with ASCII characters, except the message beginning and end. The ASCII characters are the exadecimal digits "0"÷"9" and "A"÷"F" that correspond to the nibbles of the data to transmit; in this way any information to transfer is firstly scomposed in nibbles, from most significant to least significant, and then converted in the equivalent exadecimal character. GMT has adopted the ASCII mode to obtain a message completely printable and to avoid complications and troubles caused by the timing specifications, that are typical in other encoding modes. This simplifications have a consequence: an increased number of characters, transmitted for each message. Anyway the benefits of ASCII mode (easier management and no restriction of implementation on different systems) are certainly superior.

The described frame format is applied either to query and response messages, that are different only for the information contained in the variable fields.

In the following paragraphs there are detailed information on each field of the message, either for a general level and for the GMT implementation.

SLAVE ADDRESS

In the message there is a field dedicated to the numeric address of the slave that select the device interested by the communication. Thus each GMT remote slave must have a different address and this address must be univocally defined.

The slave address coincides with a byte (equivalent to two ASCII characters) that assume the possible following values:

- 0     = 00H     = broadcast address
- 1÷247 = 01÷F7H  = individual slave addresses
- 248÷255 = F8÷FFH = reserved addresses

The accepted addresses are those in the range 1÷247, plus the value 0 that is a broadcast address, directed to all the slaves. A master selects a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in the message field to let the master know which slave is responding. When broadcast address is used then all the slaves recognize it and no response messages are transmitted.

The modalities that define the slave address change according to the manufacturer of the slave. For the GMT slaves this process has been considerably simplified by using one of the resources for the address. In this way the master can assign all the addresses of the GMT remote slaves with a simple point to point serial connection, without any additional accessories.

Further information on slave address setting are reported on figure 3 and in chapter HOW TO START.
FUNCTION CODE

The third field of the message identifies the action executed by the slave; it coincides with a byte, equivalent to two ASCII characters, variable in the range 1÷255 = 01÷FFH.

When a query message is sent from the master, the function code field tells the slave what kind of action to perform; when the response message is transmitted by the slave, the function code field indicates the result of the executed operation. For a right execution, the slave simply echoes the original function code while when errors occur the slave transmit an exception response. In the exception response, the slave returns a function code that is equivalent to the original one with its most significant bit, set to a logic 1. For detailed information about the exception messages provided by GMT, please refer to the homonymous paragraph that follow.

In MODBUS standard are defined about 25 different functions that cover the normal requirements of telecontrol applications. Many manufacturers implement only a subset of these functions, according with the hardware features of the slave, as it happens with GMT package that provide 8 functions:

<table>
<thead>
<tr>
<th>FUNCTION CODE</th>
<th>FUNCTION</th>
<th>PERFORMED ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1=01H</td>
<td>READ COIL STATUS</td>
<td>Returns status of single digital outputs, as bits</td>
</tr>
<tr>
<td>2=02H</td>
<td>READ INPUT STATUS</td>
<td>Returns status of single digital inputs, as bits</td>
</tr>
<tr>
<td>3=03H</td>
<td>READ HOLDING REGISTERS</td>
<td>Returns status of management registers, as bytes</td>
</tr>
<tr>
<td>4=04H</td>
<td>READ INPUT REGISTERS</td>
<td>Returns status of digital inputs, as bytes</td>
</tr>
<tr>
<td>5=05H</td>
<td>FORCE SINGLE COIL</td>
<td>Sets status of single digital output, as bits</td>
</tr>
<tr>
<td>6=06H</td>
<td>PRESET SINGLE REGISTER</td>
<td>Sets single management register, as bytes</td>
</tr>
<tr>
<td>15=0FH</td>
<td>FORCE MULTIPLE COILS</td>
<td>Sets status of digital outputs, as bits</td>
</tr>
<tr>
<td>16=10H</td>
<td>PRESET MULTIPLE REGISTERS</td>
<td>Sets management registers, as bytes</td>
</tr>
</tbody>
</table>

**Figure 2: Function available in GMT**

For a detailed description of these functions please read the proper chapter FUNCTIONS AVAILABLE IN GMT while for a complete description of the remainder functions please examine the MODBUS documentation.

The choice of the implemented functions takes care of the normal requirements of the most diffused applications.
FUNCTION DATA

The fourth field of the message identifies the data of the used function and it coincides with a byte sequence, equivalent to two ASCII characters, variable in the range 0÷255 = 00÷FFH.

The data field of messages sent from a master to slave contains additional information which the slave must use to take the action defined by the function code. This can include the resources addresses, the quantity of items to be handled, the count of actual data bytes in the field, etc. For example, if the master transmit the function code 2 = 02H = READ INPUT STATUS the data field specifies the starting address of digital inputs and how many inputs must be acquired.

If no error occurs, the data field of a response message transmitted from slave contains the data requested by master. If an error occurs, the response message is not sent or it has a data field that contain an exception code, described in the homonimous following paragraph. For example, in a response to function code 2 = 02H = READ INPUT STATUS the data field includes the number of returned data and a sequence of data with the status of the acquired inputs.

Frequently the data of a message that must be exchanged, are 16 (or more) bits values: in this conditions the values are scomposed in several bytes always transmitted from most significant to least significant.

In this manual, and in all MODBUS documents, are often used some terms that identifies the data of the functions. In detail:

Resource address = In MODBUS protocol, and thus in GMT, all the hardware components necessary to develop the application are defined resources. Typical example of resources are the digital inputs, digital outputs, counters, the analog channels for A/D and D/A, etc. The resource address is a numeric value, 16 bits wide, that identifies univocally the resource; this address must be used in the function data field to specify the resource/s used by the same function. Other information about GMT resources addressing are available in the following paragraph.

Holding register = The holding register is a numeric value, 16 bits wide, associated to the used hardware resources. The term is generic and it fits to numerous usages, especially when the hardware resources join with numeric value of this size. Typical example of holding register are the numeric values returned by A/D channels, the counters values, the values saved in the slave memories, etc.

For a detailed description of the function data please read the chapter FUNCTIONS AVAILABLE IN GMT.

NOTE: The maximum number of bytes inside the field dedicated to function data, depends on the features of the used slave, and with GMT is fixed to 19 bytes.

RESOURCES ADDRESSES

In the following tables are listed the addresses of the resources managed by GMT. It is important to remind that these addresses have been chosen with sufficient increase border, that allow to identifies all the resources even in the really complex systems. All resources addresses are referenced to 0 =
00H, as it happens in software environments; it must not be confused with the resource numbers that instead starts from 1.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RESOURCES</th>
<th>ADDRESS</th>
<th>ADDRESS HEX</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARIOUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slave address</td>
<td></td>
<td>0</td>
<td>0000H</td>
<td>-</td>
</tr>
<tr>
<td>Delay for response</td>
<td></td>
<td>1</td>
<td>0001H</td>
<td>(1)</td>
</tr>
<tr>
<td>transmission in</td>
<td></td>
<td>2</td>
<td>0002H</td>
<td>(2)</td>
</tr>
<tr>
<td>msec</td>
<td></td>
<td>3</td>
<td>0003H</td>
<td>(3)</td>
</tr>
<tr>
<td>Parity of physical</td>
<td></td>
<td>4</td>
<td>0004H</td>
<td>(4)</td>
</tr>
<tr>
<td>communication</td>
<td></td>
<td>5</td>
<td>0005H</td>
<td>(5)</td>
</tr>
<tr>
<td>protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop bit of physical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firmware versione</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code for hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>identification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIGITAL INPUTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group of digital</td>
<td></td>
<td>2048</td>
<td>0800H</td>
<td>-</td>
</tr>
<tr>
<td>inputs 1÷16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital input 1</td>
<td></td>
<td>2048</td>
<td>0800H</td>
<td>-</td>
</tr>
<tr>
<td>Digital input 2</td>
<td></td>
<td>2049</td>
<td>0801H</td>
<td>-</td>
</tr>
<tr>
<td>Digital input 8</td>
<td></td>
<td>2055</td>
<td>0807H</td>
<td>-</td>
</tr>
<tr>
<td>Digital input 9</td>
<td></td>
<td>2056</td>
<td>0808H</td>
<td>-</td>
</tr>
<tr>
<td>Digital input 16</td>
<td></td>
<td>2063</td>
<td>080FH</td>
<td>-</td>
</tr>
<tr>
<td>Group of digital</td>
<td></td>
<td>2064</td>
<td>0810H</td>
<td>-</td>
</tr>
<tr>
<td>inputs 17÷32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital input 17</td>
<td></td>
<td>2064</td>
<td>0810H</td>
<td>-</td>
</tr>
<tr>
<td>DIGITAL OUTPUTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group of digital</td>
<td></td>
<td>4096</td>
<td>1000H</td>
<td>-</td>
</tr>
<tr>
<td>outputs 1÷16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital output 1</td>
<td></td>
<td>4096</td>
<td>1000H</td>
<td>-</td>
</tr>
<tr>
<td>Digital output 2</td>
<td></td>
<td>4097</td>
<td>1001H</td>
<td>-</td>
</tr>
<tr>
<td>Digital output 8</td>
<td></td>
<td>4103</td>
<td>1007H</td>
<td>-</td>
</tr>
<tr>
<td>Digital output 9</td>
<td></td>
<td>4104</td>
<td>1008H</td>
<td>-</td>
</tr>
<tr>
<td>Digital output 16</td>
<td></td>
<td>4111</td>
<td>100FH</td>
<td>-</td>
</tr>
<tr>
<td>Group of digital</td>
<td></td>
<td>4112</td>
<td>1010H</td>
<td>-</td>
</tr>
<tr>
<td>outputs 17÷32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital output 17</td>
<td></td>
<td>4112</td>
<td>1010H</td>
<td>-</td>
</tr>
</tbody>
</table>

**Figure 3: Resources addresses table - Part 1 of 2**
<table>
<thead>
<tr>
<th>TYPE</th>
<th>RESOURCES</th>
<th>ADDRESS</th>
<th>ADDRESS HEX</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTERS</td>
<td>Value of counter 1</td>
<td>6144</td>
<td>1800H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value of counter 2</td>
<td>6145</td>
<td>1801H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value of counter 3</td>
<td>6146</td>
<td>1802H</td>
<td>-</td>
</tr>
<tr>
<td>PULSE WIDTH MODULATION SIGNALS</td>
<td>Duty cycle of channel 1</td>
<td>8192</td>
<td>2000H</td>
<td>-</td>
</tr>
<tr>
<td>(PWM)</td>
<td>Duty cycle of channel 2</td>
<td>8193</td>
<td>2001H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Duty cycle of channel 3</td>
<td>8194</td>
<td>0202H</td>
<td>-</td>
</tr>
<tr>
<td>ANALOG INPUTS (A/D)</td>
<td>Value acquired from channel 1</td>
<td>10240</td>
<td>2800H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value acquired from channel 2</td>
<td>10241</td>
<td>2801H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value acquired from channel 3</td>
<td>10242</td>
<td>2802H</td>
<td>-</td>
</tr>
<tr>
<td>ANALOG OUTPUTS (D/A)</td>
<td>Value to set on channel 1</td>
<td>12288</td>
<td>3000H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value to set on channel 2</td>
<td>12289</td>
<td>3001H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Value to set on channel 3</td>
<td>12290</td>
<td>3002H</td>
<td>-</td>
</tr>
<tr>
<td>REAL TIME CLOCK (RTC)</td>
<td>Seconds</td>
<td>14336</td>
<td>3800H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Hours</td>
<td>14337</td>
<td>3801H</td>
<td>(6)</td>
</tr>
<tr>
<td></td>
<td>Minutes</td>
<td>14338</td>
<td>3802H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Day</td>
<td>14339</td>
<td>3803H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Month</td>
<td>14340</td>
<td>3804H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Year</td>
<td>14341</td>
<td>3805H</td>
<td>(7)</td>
</tr>
<tr>
<td></td>
<td>Week day</td>
<td>14342</td>
<td>3806H</td>
<td>(8)</td>
</tr>
<tr>
<td>BACKED SRAM</td>
<td>1st word for user requirements</td>
<td>16384</td>
<td>4000H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2nd word for user requirements</td>
<td>16385</td>
<td>4001H</td>
<td>-</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1st word for user requirements</td>
<td>18432</td>
<td>4800H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2nd word for user requirements</td>
<td>18433</td>
<td>4801H</td>
<td>-</td>
</tr>
</tbody>
</table>

**FIGURE 4: RESOURCES ADDRESSES TABLE - PART 2 OF 2**
The greater part of the resources described in tables of figures 3 and 4, don't require any additional explanation. The only additional information, necessary to correctly use GMT resources, are the following ones (the used number match the number reported in the column NOTE of the tables):

1) The delay for response transmission is a time, expressed in msec, that the slave waits once it has executed the function and before to transmit its response message. This delay is especially useful when master system is slow or when an RS 422, RS 485 serial network is used and there are converters or drivers that needs time, to switch its status.

2) The parity of physical communication protocol is coded with the following numeric values:
   
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No parity</td>
</tr>
<tr>
<td>1</td>
<td>Odd parity</td>
</tr>
<tr>
<td>2</td>
<td>Even parity</td>
</tr>
</tbody>
</table>

3) The stop bit of physical communication protocol is coded with the following numeric values:
   
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 stop bit</td>
</tr>
<tr>
<td>2</td>
<td>2 stop bits</td>
</tr>
</tbody>
</table>

4) The firmware version is a numeric value obtained by version number multiplied by 10. For example with a 1.1 version this resource assumes the value 11.

5) The code for hardware identification is an univocal numeric value that allow to recognize the kind of connected slaves. Indirectly, as described on figure 5, this code let the master know also the slave configuration, concerning the available hardware resources.

6) The hours of real time clock are always used in the 24 hours format and so variable in the range 0+23.

7) The year of real time clock is always managed with two digits format and so variable in the range 0+99.

8) The week day of real time clock is always managed with the following numeric values:
   
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Sunday</td>
</tr>
<tr>
<td>1</td>
<td>Monday</td>
</tr>
<tr>
<td>2</td>
<td>Tuesday</td>
</tr>
<tr>
<td>3</td>
<td>Thursday</td>
</tr>
<tr>
<td>4</td>
<td>Wednesday</td>
</tr>
<tr>
<td>5</td>
<td>Friday</td>
</tr>
<tr>
<td>6</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

9) All the resources listed in the tables of figures 3 and 4 can be accessed through holding registers, this means that they are always coded with 16 bits, even if some of them require less space. This choice has been done to allow the management of all the hardware resources with an higher number of GMT functions that, as described in the following chapter, often use 16 bits values inside their function data.
ERRORS CHECK

There are two types of errors which may occur in a serial communication system: transmission/reception errors and programming, operation errors. The MODBUS and GMT have specific methods for dealing with either type of errors.

Transmission/reception errors usually consist of a changed bit or bits within a transferred message and their most frequent cause are the electric noise or, in other words, unwanted signals that interfere on communication line (impulses, spikes, etc.).

The programming and operation errors coincide with wrong settings of the message fields with consequent access to resources or functions not available.

The methods provided by GMT for the errors check are the following ones:

Parity

itr is the parity control, applied to each characters of the message, that allows to found out possible bit variations. It can be set in the three typical modes, available on each asynchronous serial lines, that are: no parity, odd parity, even parity. Further information can be found in ELECTRIC AND PHYSIC PROTOCOL paragraph.

LRC

it is a Longitudinal Redundancy Check applied on many characters of the message contents. It coincides with the fifth field of the message frame and it is calculated as below described:
- add all the bytes (not characters) of the message contained in slave address, function code and function data fields in a byte value without wraparound carry;
- perform the two's complement of the obtained byte;
- transmit the obtained complemented byte, with the usual ASCII encoding (2 characters).

The system that transmit the message simply calculates and transmit the LRC while the system that receive a message must calculate the LRC and then check the equality with the received one. The LRC ensures that a received message that has been recognized valid, is really right; so it satisfy all the safety requirements of industrial environments, where GMT is normally applied.

Data validity

the GMT verifies that the values received in the fields function code and function data are valid. In other words it checks if the function required by master is one of the 8 implemented, that the addressed resources are available on the slave and that the data relative to resources are compatible. Whenever the described checks are not satisfied and the data are not valid, the slave returns a response that is an exception message, as described in the following paragraph.

When a slave recognizes an error in the received message, this query message is dicharged: thus the slave doesn’t execute the required function nor return the response message. Consequently the master must wait the slave response message for a maximum time out, properly defined, that avoid unacceptable endless condition when errors happens or when not available slaves are addressed.

Summarizing, one of four possible events can occur from the master’s query:

- If no slaves device receive the query because a communication error happens or the slave address is not valid, then no response message is returned.
- If the slave device receives the query without a communication error, and can handle the query normally, it executes the function and returns a normal response message unless the query was broadcast type.

- If the slave receives the query, but detects a communication error (parity or LRC), it doesn't execute the function and no response message is returned.

- If the slave receives the query without a communication error, but cannot handle due to non valid data, the slave will return an exception response informing the master of the nature of the error.

**EXCEPTION SUPPORTED**

In MODBUS protocol an exception refers to anomalous conditions in the received query message, that avoid its execution as it normally happens. Among the possible causes of these anomalies there are the not valid functions or data for the slave, the slaves engaged in other operations, the slave not able to execute the function for an hardware malfunctions, etc.

As specified in previous paragraph, when an anomaly is found the slave doesn't execute the function and it signals to master this conditions, by returning an exception response. The exception message is different from normal response message only for the function code and function data fields, while all the other fields have no differences.

In detail the function code is not returned unchanged (simple echo) but with the most significant bit set to 1. This makes the function code value in an exception response exactly $128 = 80H$ higher than the value would be for a normal response. In the function data field there aren't the value required by the function but a single byte, named exception code. The exception code identifies the anomaly occured on the slave that generate the exception and, in **GMT** implementation, it can assume the following values:

- 01 -> Illegal function code for the slave
- 02 -> Illegal data address on the resources in the data field, for the slave
- 03 -> Illegal data value in the data fiels for the resources, for the slave

With this technique the master can easily recognizes an exception response and it can examine the data field for the exception code, to decide which operations perform. In such a way the occoured anomaly is solved.

When **GMT** is used, the user can avoid the exception response by consulting the figure 5 that lists all the available resources on each different implementation.

**ELECTRIC AND PHYSIC PROTOCOL**

The MODBUS and **GMT** define the logic protocol for serial communication but they let free selectable either the electric and physic communication protocol. In this way the user can always select the best ones that satisfy his requirements, or choose the ones that perfectly cover the features of the already developed communication network.
In detail it can be used one of the most diffused standard in industrial environments as RS 232, Current Loop, RS 422 and RS 485, in fact GMT already provides the management of all these electric protocols, in all the developed implementations. For the user it is sufficient to configure the GMT remote slaves with the selected electric interfaces, by following the information on technical manuals or by ordering directly to grifo® the slave already configured with the proper serial option. Naturally, the serial connection of the devices will change according to the selected electric protocol, as described always in the technical manuals of the used cards. On these manuals in fact there are: the pin outs of the serial connectors, the configuration required on the slave and the diagrams of some usefull connection examples.

The physic protocol of the asynchronous serial line, defines how each character is transferred on the communication line and, with GMT, it is configurable as below described:

- Start bit = 1
- Bit per characters = 7
- Parity = even, odd or none, programmable with proper function on homonimous resource
- Stop Bit = 1 or 2, programmable with proper function on homonimous resource
- Baud rate = 2400, 4800, 9600, 19200, 38400 with autodetection performed on the first character received after each power on of the GMT remote slave. This first character must absolutely coincide with the beginning of message character (":" = 58 = 3AH)

The indication "programmable with proper function on homonimous resource" is detaily described in the paragraph CONFIGURATION OF GMT REMOTE SLAVE.

NOTE: The GMT remote slave based on CPUs of I51 family (GMT 932, GMT GM1, GMT AC2, etc.) must absolutely be used with a physic protocol that ensures at least 10 bits per characters inclusive. In other words when no parity is used, then the master must set 2 stop bits even if the physic protocol of the slaves has 1 stop bit, i.e. it totalizes only 9 bits. Alternatively it is sufficient that between two communicated characters, elapses a time of at least two bits.

CONFIGURATION OF GMT REMOTE SLAVE

The GMT offers an interesting and innovative possibility to configure some parameters of the slave, through a proper function on resources purposely defined. The MODBUS function forecasted for this configuration is the PRESET MULTIPLE REGISTERS that must be used in broadcast mode (with slave address 0 = 00H), while the resorces are the following ones:

<table>
<thead>
<tr>
<th>Configuration resource</th>
<th>default settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave address</td>
<td>128 = 80H</td>
</tr>
<tr>
<td>Delay for response transmission in msec</td>
<td>0 = 00H</td>
</tr>
<tr>
<td>Parity of physical communication protocol</td>
<td>0 = 00H (= None)</td>
</tr>
<tr>
<td>Stop Bit of physical communication protocol</td>
<td>1 = 01H</td>
</tr>
</tbody>
</table>
The **GMT** slave once received a query message, that includes the described function and resources, doesn't perform the normal control of parity and frame on the characters of the same message; consequently the slave always execute the function, even if the previous settings shall prevent this. The addresses and the value of the resources required for slave configuration, are described in detail into paragraph RESOURCES ADDRESSES and inside figure 3.

Trough this configuration method the user can configure the slaves according with his requirements, for example he can set a free slave address, set different slave addresses when serial networks are used, preset the physic protocol, etc.

In the following list are described the steps, that must be executed, to correctly configure a **GMT** remote slave:

1)  prearrange a master system capable to communicate with the **GMT** remote slave to configure and capable to manage a MODBUS communication;

2)  ensure that the arranged master is connected **only** to the **GMT** remote slave to configure;

3)  activate both the master and the slave to configure;

4)  on the master set the physic protocol: 9600 Baud, 7 bit x chr, no parity, 2 stop bits;

5)  with the master transmit the MODBUS message with the function PRESET MULTIPLE REGISTERS, the slave address 0 and the addresses and values of the required configuration resources, with no parity nor frame checks on the received response;

6)  turn off and then turn on the slave;

7)  on the master set the desidered physic protocol, that is the one partially configured at step 5;

8)  acquire the value of the 4 configuration resources, by using on the master the function READ HOLDING REGISTERS on the slave with the set up address, by performing the parity and frame checks on the received response;

9)  verify that step 8 is executed without errors and the acquired values are those set up at step 5 or, in other words, the desidered ones.

Please remind that the program **Demo GMT**, completely described in one of the following chapter, includes a dedicated modality for slave configuration that execute the listed steps with a simple PC that act as the master unit.

In normal circumstances the slave configuration must be performed only one time after the purchase of the **GMT** salve and before its installation in the application to develop; whenever the default settings before described, is already valid for the application, the configuration is no more necessary. When the **GMT** slave are used in existent networks, the user must know the physic protocol and the free slave addresses of this installation and then use this information to configure correctly the new added devices.
AVAILABLE IMPLEMENTATION

The GMT protocol has been designed to be easily and rapidly implemented on the maximum number of the numerous grifo® cards. At the time this manual is written, are available some implementations that have been selected after a detailed examination of the technical features and the total cost of the final product.

The features of these preselected implementation are listed in the following table:

<table>
<thead>
<tr>
<th>NAME</th>
<th>GMT 932</th>
<th>GMT AM08</th>
<th>GMT GM1</th>
<th>GMT AC2</th>
<th>GMT HGM1</th>
<th>GMT HAC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW CARDS</td>
<td>GMM 932</td>
<td>GMM AM08</td>
<td>CAN GM1</td>
<td>GMM AC2</td>
<td>CAN GM1</td>
<td>GMM AC2</td>
</tr>
<tr>
<td></td>
<td>+ GMB HR84</td>
<td>+ GMB HR84</td>
<td>+ GMB HR84</td>
<td>+ GMB HR168</td>
<td>+ GMB HR246</td>
<td>+ GMB HR246</td>
</tr>
<tr>
<td>HARDWARE IDENTIFIC. CODE</td>
<td>0 = 0000H</td>
<td>1024 = 0400H</td>
<td>9 = 0009H</td>
<td>77 = 004DH</td>
<td>135 = 0087H</td>
<td>141 = 008DH</td>
</tr>
<tr>
<td>DIGITAL INPUTS</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>DIGITAL OUTPUTS</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>ANALOG INPUTS A/D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ANALOG OUTPUTS D/A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COUNTERS</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>PWM</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BACKED SRAM</td>
<td>0</td>
<td>0</td>
<td>239</td>
<td>239</td>
<td>239</td>
<td>239</td>
</tr>
<tr>
<td>RTC</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>EEPROM</td>
<td>508</td>
<td>508</td>
<td>2044</td>
<td>2044</td>
<td>2044</td>
<td>2044</td>
</tr>
</tbody>
</table>

**Figure 5: Features of the available implementations**

The user interested to implementation on different hardwares is invited to contact directly grifo® and to explain his requirements.
FUNCTIONS AVAILABLE IN GMT

In GMT are defined eight different functions, that are sufficient to manage all the hardware features of the remote slave, where it is implemented according to the normal requirements of the most diffused applications.

The following paragraphs describe the data communicated both in the query message (from master to slave) and in the response message (from slave to master) for each one of the provided functions; between the communicated information are not described those of the constant fields of the message (begin, slave address, LRC error check and end) but only the variable fields (function code, function data). All the data are always reported in bytes, not characters. Moreover each paragraph indicate if the relative function can be used in broadcast mode.

READ COIL STATUS (1=01H)

**Broadcast**

- = NO

**Query**

| function code | = 01H |
| function data | address H of first coil = xxH |
|               | address L of first coil = xxH |
|               | number of coils H = xxH |
|               | number of coils L = xxH |

**Response**

| function code | = 01H |
| function data | total bytes count = xxH |
|               | 1st coil status byte = xxH |
|               | :    :    :    : = xxH |
|               | nth coil status byte = xxH |

**Description**

Reads the current status of discrete digital outputs in the addressed slave, at bit level. The query message specifies the starting coil address and quantity of coils to be read. The coils status in the response message is packed as one coil per bit of the data bytes. Status is indicated as: 1=ON=enabled, 0=OFF=disabled. The least significant bit of the first data byte contains the coil addressed in the query. The other coils follow toward the high order end of this byte, and from low order to high order in subsequent bytes. If the returned coils quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros. The bytes count specifies the quantity of data bytes, either complete or incomplete.
READ INPUT STATUS (2=02H)

**Broadcast**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 02H</td>
</tr>
</tbody>
</table>

**Query**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 02H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>function data</th>
</tr>
</thead>
<tbody>
<tr>
<td>address H of first input</td>
</tr>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>address L of first input</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of inputs H</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of inputs L</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 02H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>function data</th>
</tr>
</thead>
<tbody>
<tr>
<td>total bytes count</td>
</tr>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st byte with input status</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

| :                     |
| :                     |
| :                     |

<table>
<thead>
<tr>
<th>nth byte with input status</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Description**

Reads the ON/OFF status of discrete inputs in the slave, at bit level.

The query message specifies the starting input address and quantity of inputs to be read.

The inputs status in the response message is packed as one input per bit of the data field. Status is indicated as: 1 = ON = enabled; 0 = OFF = disabled. The LSB of the first data byte contains the input addressed in the query. The other inputs follow toward the high order end of this byte, and from low order to high order in subsequent bytes. If the returned input quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros. The byte count field specifies the quantity of data bytes, either complete or incomplete.

READ HOLDING REGISTERS (3=03H)

**Broadcast**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= NO</td>
</tr>
</tbody>
</table>

**Query**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 03H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>function data</th>
</tr>
</thead>
<tbody>
<tr>
<td>address H of first resource</td>
</tr>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>address L of first resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of registers H</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of registers L</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>function code</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 03H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>function data</th>
</tr>
</thead>
<tbody>
<tr>
<td>total bytes count</td>
</tr>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte H of 1st register</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte L of 1st register</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

| :                     |
| :                     |
| :                     |

<table>
<thead>
<tr>
<th>byte H of nth register</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte L of nth register</th>
</tr>
</thead>
<tbody>
<tr>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Description**

Reads the binary contents of holding registers in the slave.

The query message specifies the address of the first resources and quantity of registers to be read.

The data in the response message are packed as two bytes per holding register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.
**READ INPUT REGISTERS (4=04H)**

**Broadcast**

**Query**

- function code = 04H
- function data: address H of first group of inputs = xxH
- address L of first group of inputs = xxH
- number of register H = xxH
- number of register L = xxH

**Response**

- function code = 04H
- function data: total bytes count = xxH
- byte H of 1st register = xxH
- byte L of 1st register = xxH
- : : : : = xxH
- byte H of nth register = xxH
- byte L of nth register = xxH

**Description**

Reads the binary contents of input registers in the slave, at byte level. The query message specifies the address of the first group of digital inputs and quantity of registers to be read. The register data in the response message are packed as two bytes per holding register=group of inputs, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

**FORCE SINGLE COIL (5=05H)**

**Broadcast**

**Query**

- function code = 05H
- function data: coil address H = xxH
- coil address L = xxH
- coil status H = xxH
- coil status L = xxH

**Response**

- function code = 05H
- function data: coil address H = xxH
- coil address L = xxH
- coil status H = xxH
- coil status L = xxH

**Description**

Forces a single coil to either ON or OFF, in the addressed slave. When broadcast, the function forces the same coil reference in all attached slaves. The query message specifies the coil address to be forced. The requested ON/OFF status is specified by a constant in the query data field. A value of FF00H requests the coil to be ON=enabled. A value of 0000H requests it to be OFF=disabled; all other values are illegal and will not affect the coil. The normal response is an echo of the query.
**PRESET SINGLE REGISTER (6=06H)**

**Broadcast**
- function code = 06H
- resource address H = xxH
- resource address L = xxH
- preset byte H = xxH
- preset byte L = xxH

**Query**
- function code = 06H
- resource address H = xxH
- resource address L = xxH
- preset byte H = xxH
- preset byte L = xxH

**Response**
- function code = 06H
- resource address H = xxH
- resource address L = xxH
- preset byte H = xxH
- preset byte L = xxH

**Description**
Presets a value into a single resource on the addressed slave. When broadcast, the function presets the same resource in all the attached slaves.

The query message specifies the resource address to be preset and the requested preset value in the data field.

The normal response is an echo of the query.

---

**FORCE MULTIPLE COILS (15=0FH)**

**Broadcast**
- function code = 0FH
- address H of first coil = xxH
- address L of first coil = xxH
- number of coils H = xxH
- number of coils L = xxH
- force bytes count = xxH
- 1st force byte for coils = xxH
- : : : :
- nth force byte for coils = xxH

**Query**
- function code = 0FH
- address H of first coil = xxH
- address L of first coil = xxH
- number of coils H = xxH
- number of coils L = xxH

**Response**
- function code = 0FH
- address H of first coil = xxH
- address L of first coil = xxH
- number of coils H = xxH
- number of coils L = xxH

**Description**
Forces each coils in a sequence to either ON or OFF status, in the addressed slave. When broadcast, the function forces the same coils in all the attached slaves.

The data field of query message specifies the first coil address to be forced, the number of coils to force and the requested ON/OFF status. These status are specified by the force bytes contained in the query data field: a logical 1 in a bit position of the field requests the corresponding coil to be
ON=enabled, a logical 0 requests it to be OFF=disabled. The first byte transmitted addresses the first 8 coils group, with the least significant bit addressing the lowest coil in this group. Unused bits in the last data byte must be zero filled.

The normal response returns the starting address, and quantity of coils forced.

**PRESET MULTIPLE REGISTERS (16=10H)**

**Broadcast**

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Address H of first resource</th>
<th>Address L of first resource</th>
<th>Number of resources H</th>
<th>Number of resources L</th>
<th>Preset bytes count</th>
</tr>
</thead>
<tbody>
<tr>
<td>= YES</td>
<td>= xxH</td>
<td>= xxH</td>
<td>= xxH</td>
<td>= xxH</td>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Query**

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Address H of first resource</th>
<th>Address L of first resource</th>
<th>Number of resources H</th>
<th>Number of resources L</th>
<th>1st preset byte H</th>
<th>1st preset byte L</th>
<th>:</th>
<th>:</th>
<th>:</th>
<th>nth preset byte H</th>
<th>nth preset byte L</th>
</tr>
</thead>
</table>

**Response**

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Address H of first resource</th>
<th>Address L of first resource</th>
<th>Number of resources H</th>
<th>Number of resources L</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 10H</td>
<td>= xxH</td>
<td>= xxH</td>
<td>= xxH</td>
<td>= xxH</td>
</tr>
</tbody>
</table>

**Description**

Presets values into a sequence of resources, in the addressed slave. When broadcast, the function presets the same resources in all the attached slaves.

The query message specifies the address of the first resource to be preset, the number of data registers and the requested preset values that are all specified in the query data field, organized as two bytes per resources=holding register.

The normal response returns the starting address and quantity of registers preset.
After the technical and functional description of GMT, reported in previous paragraph, in this chapter are described his possible applications.

**FIGURE 6: I/O PLACEMENTS ON HOUSE APPLICATION**
<table>
<thead>
<tr>
<th>Ref.</th>
<th>Device Name</th>
<th>Device Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ground Floor Windows Sensor</td>
<td>Sensor that inform about the open and close status of the windows placed on the ground floor.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>2</td>
<td>Solar Sensor</td>
<td>Sensor that measure the light diffused in the surrounding environment, compare it with a presetted threshold and generate a digital ON/OFF signal.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>3</td>
<td>Rain Sensor</td>
<td>Sensor capable to recognize the rain fallen in the surrounding environment and that generates a related digital ON/OFF signal.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>4</td>
<td>First Floor Windows Sensor</td>
<td>Sensor that inform about the open and close status of the windows placed on the first floor.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>5</td>
<td>Door Sensor</td>
<td>Sensor that inform about the open and close status of the door where it is mounted.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>6</td>
<td>Fire Detection</td>
<td>Sensor capable to recognize the smoke caused by fire, in the surrounding environment. It generates a related digital ON/OFF signal.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>7</td>
<td>Presence Sensor</td>
<td>Sensor capable to recognize the presence and the movement inside the surrounding environment, where it is installed. It generates a related digital ON/OFF signal.</td>
<td>Optocoupled Input</td>
</tr>
<tr>
<td>8</td>
<td>Thermostat</td>
<td>Sensor that measures the temperature of the surrounding environment, compares it with a settable threshold and generates a digital ON/OFF signal.</td>
<td>Optocoupled Input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Device Name</th>
<th>Device Description</th>
<th>Signal Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Hooter</td>
<td>High intensity horn capable to call the attention of the surrounding peoples.</td>
<td>Relay Output</td>
</tr>
<tr>
<td>A2</td>
<td>Luminaire</td>
<td>Electric light that illuminates the external environment, with low consumption.</td>
<td>Relay Output</td>
</tr>
<tr>
<td>B1</td>
<td>Heating Pump</td>
<td>Electric pump that enable or disable the recirculation of the heating water.</td>
<td>Relay Output</td>
</tr>
<tr>
<td>B2</td>
<td>Irrigation Valve</td>
<td>Electric valve that open or close the supply of water to the irrigation plant.</td>
<td>Relay Output</td>
</tr>
</tbody>
</table>

**Figure 7: I/O Descriptions on House Application**
**Figure 8: I/O Connection on House Application**

- **Door Sensor (5)**
- **Fire Detection (6)**
- **Presence Sensor (7)**
- **Thermostat (8)**
- **First Floor Windows Sensor (4)**
- **Rain Sensor (3)**
- **Solar Sensor (2)**
- **Ground Floor Windows Sensor (1)**
- **MODBUS Line**

- **Irrigation Valve (B2)**
- **Heating Pump (B1)**
- **Luminaire (A2)**
- **Hooter (A1)**
The three previous figures show graphically a possible application of GMT in house environment, that is a really simple example of building automation. The figure 6 shows a list of sensors and actuators, that can be installed in a house, to obtain a automatic control of:

- anti intrusion allarm,
- anti fire allarm,
- the heating of the building,
- the curtesy lighting in the night,
- the irrigation of the garden.

The tabel of figure 7详细 describes these sensors/actuators, by reporting also the digital signals necessary to acquire/command them. Finally figure 8 shows how these signals can be connected to one of grifo® card, that has the GMT implemented. The user can easily use the minimum implementation GMT 932 and a master system, for example a PC, to obtain the required functionalities on the 5 described controls. By developing a program for this PC, then it can use the GMT functions that reads the digital inputs to acquire the status of the sensors, determine the status of the outputs through dedicated algorithms and finally set the actuators, by using the GMT functions that force the coils. The three figures of this chapter uses the classic green colour for the digital inputs and red for the digital outputs, on all the used signals.

In addition to the house application example, the chapter is completed with the list of some other typical GMT applications. The descriptions of these applications is reduced to minimum in fact their most important scope is to supply ideas to users and let them know, how it is easy their development.

- Management of access in and access out in a park place.
- Management of little production machines with centralized supervision.
- Management of large production machines with distributed logic.
- System for pieces counting, with additional status and command signals, driven from distance.
- Teletransport of digital signals.
- Teletransport of analog signals.
- Packaging machines at the end of production line.
- Peripherals for SCADA systems.
- Management of spy lamps and push buttons for a remote operator interfaces.
- Data logher system for field signals.
- Water control systems on the lands.
- Addition of signals to installed MODBUS networks.
- Etc.
DEMO GMT PROGRAM

In correspondence of the first purchase of GMT package, it is provided also a demo program for PC, named obviously Demo GMT, that performs all the operations normally required by each user. Among the most important features of this program, we remind:

- The execution of all the operating systems Windows 98, ME, 2000, XP.
- Few requirements for execution (see details in homonimous REQUIREMENTS paragraph).
- Supplied with proper installation and uninstallation program, that solves the problem during arrangement and first use.
- Implementation of MODBUS protocol related to modalities used by GMT.
- Based on high level communication library for GMT. These library can be easily adopted by the users to develop his own application program for PC.
- Management of serial communication with electric protocol RS 232, Current Loop, RS 422 and RS 485. For the last two protocols, the Demo GMT provides also the management of hardware handshake /RTS, to drive properly all the line converters available on the market.
- It can be used both in point to point connections, in RS 232 and networks. For network connection, the serial line of the PC that execute the program must be properly converted in one of the electric protocols Current Loop, RS 422, RS 485. This conversion can be comfortably done by using the grifo® converters and concentrators, as the CIC 232-485, CIC 232-CLOOP, IBC CL, NNI 16.
- Management of all the communication status, including the errors and exception ones, that are properly displayed on PC monitor.
- Provides several security controls, in order to ensure the highest reliability on all the communicated data.
- The PC that execute Demo GMT always acts a master device.
- Support for two different functional modalities: those for GMT slave configuration and those of manual test.
- The GMT slave configuration modality allows to arrange the GMT slaves for following use, either with Demo GMT or any other master.
- The manual test modality allows to try the communication with GMT slave and to try all the hardware resources, available on the same slave.
- Autoidentification of the connected slave type and consecutive automatic configuration of the addressable resources.
- Visualization of the hardware resources available in a GMT remote slave, in a proper windows that is easy to use and simple to guess. By using the fast and comfortable mouse all the provided operations can be performed.
- Support of two different languages (Italian and English).

Please remind that Demo GMT is an interesting program that simplifies the GMT use both to expert and inexpert staff. For this reason we suggest at least one test also to the users that already know MODBUS protocol and already have other PC communication program for this protocol. Probably they consider Demo GMT useless, but it isn't.

The following paragraphs report the information about Demo GMT use, accompanied by explanation figures.
INSTALLATION, DEINSTALLATION

Before using Demo GMT the user must correctly install it. For this reason a specific program has been developed, as described in the following steps:

I1) Once found the installation file SETUP.EXE, from one of the distribution forms, execute it with a simple double click on its icon.

I2) At this point the installation program starts and first of all it checks the presence of the Net Framework package and, if it is not available, it predispose for a direct download from Microsoft web site and following installation.

FIGURE 9: REQUIREMENTS WINDOWS OF DEMO GMT

Note: The message will be in the operating system language: in figure 9 it is Italian.

I3) At the end of the checks it shows a presentation window and then, when the Next > button has been pushed, it shows a window that let the user select the destination folder of Demo GMT. Normally it is suggested the default folder C:\Programmi\Grifo®\Demo GMT\ but the user can easily modify it with the Change... push button; in this circumstance it is displayed a typical selection window that allows also to create new folders.

I4) Once the installation folder has been selected, press again the Next > button to show the following window. This schedules a summary with the selected folder and some information on used PC: if the performed selections are not right the user can correct them by pressing the < Back button, viceversa he can start the real installation, by pushing the Next button.

I5) During installation are shown some windows with respective progress bar that inform about the execution of the current phase and at the end (normally after few seconds) it is displayed another window with the installation results. This window must be closed, by pressing the Close button, that exit also the installation program.

I6) The Demo GMT installation program prepares the executable file in the folder selected at point I3, and additionally it creates the links inside the Start menu of Windows, for the program. It is suggested to check the presence of this link inside the path Start | Program | Grifo® | Demo GMT | Demo GMT

I7) If the point I5 has been successfully executed, and the link described in point I6 is available, the installation of Demo GMT is correctly executed and the program is ready to be launched.

The reverse operation that removes the program, can be performed through the dedicated manager of the operating system, by executing the following steps:
R1) Execute the installation manager placed in Start | System settings | Control panel | Install applications.

R2) Select the application Demo GMT from the displayed list.

R3) Press the Add/Remove... button and wait that the installation and deinstallation window is displayed.

R4) In the Installation maintenance window that is shown at this point, select the Remove Demo GMT, as described in following figure.

![Figure 10: Remove Demo GMT selection window](image)

Note: The message will be in the operating system language; in figure 10 it is Italian.

R5) Confirm the removal of Demo GMT by pushing the Finish button;

R6) During the disinstallation are shown some windows with respective progress bar that inform about execution of the current phase and at the end (normally after few seconds) it is displayed another window with the results.

R7) Press the Close button to exit the deinstallation program.

Normally the installation of Demo GMT must be done only one time and so it is not necessary to remove and reinstall it; the unusual conditions that may require the reinstallation are the variation of the installation folder or the availability of a new version of Demo GMT.
EXECUTION AND USE

As specified in previous chapter the Demo GMT program can be simply executed by using the fast link, placed in the start menu of Windows, from the installation program:

Start | Program | Grifo® | Demo GMT | Demo GMT

This link can’t be modified but the user can freely create other links and place them where he prefers, as for example in the GMT work directory, on the desktop, on Windows application bar, inside other frequently used folders, etc.

At this point the Demo GMT starts and it check the presence of the equipment files, that contains the languages support and the graphics backgrounds: if they are all available it proceeds, viceversa it shows a dedicated error message. When all the necessary files and elements are loaded, the Demo GMT shows its main window, where are available all the menus and options for the provided functionalities.

![Main window of Demo GMT](image)

**Figure 11: Main window of Demo GMT**

The functions performed by each one of these menus, are below described:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set Comm Parameters</td>
<td>Opens the homonimous windows for the selection of the free communication port (COMx), of Baud rate, Stop Bit and Parity of PC. Among the communication ports are listed both the internal and external to PC, but only if they are correctly installed and managed by the used operating system. The remaining selections musty be done according with physic protocol set on GMT remote slave.</td>
</tr>
<tr>
<td></td>
<td>Configure Slave</td>
<td>Manages the configuration process of a GMT remote slave, as described in the following paragraph.</td>
</tr>
<tr>
<td></td>
<td>Exit</td>
<td>Stops execution of Demo GMT and return definitely under Windows control.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>Add slave</th>
<th>Enables the communication with a GMT remote slave, in the manual test modality. The identification of the slave to add is based on its slave address, as described in MANUAL TEST paragraph.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Remove slave</td>
<td>Disables the communication with a GMT remote slave, in the manual test modality. The identification of the slave to remove is based on its slave address, as described in MANUAL TEST paragraph.</td>
</tr>
</tbody>
</table>
Menu | Option  Function

Scan Rate |  Allows to change the frequency used by Demo GMT to interrogate the GMT remote slaves, in the manual test modality. For detailed information please refer to MANUAL TEST paragraph.

Set Scan Rate

Language |  Shows a drop down list where the user can choose the used language between Italian and English.

Set Language

Help |  Opens the homonimous windows that show all the information about the Demo GMT in execution. Among these information are included the program version, the address, phone number, fax number, etc. of grifo® company and a direct link to its web site.

About

SLAVE CONFIGURATION

The configuration modality of the GMT remote slave, provided by Demo GMT, executes the operations dedicated to master, already described in previuos paragraph CONFIGURATION OF GMT REMOTE SLAVE.

Below are listed the sequence of steps that must be executed, to succesfully use this modality:

C1) Arrange the PC for the communication with GMT remote slave to be configured.
C2) Ensure that the prepared PC is connected only with the GMT remote slave that must be configured; the possible network connections with different electric protocols can be anyway used, but it is necessary to remove all the other slaves.
C3) Enable the PC, executes the Demo GMT, select the option File | Configure Slave and confirm the information window that inform about the single slave connection. At this point the following window is displayed:

![Slave Configuration Window](image)

**Figure 12: Slave Configuration Window**
C4) Inside the slave configuration window selects the PC serial line connected on point C2 and set the value of the 4 configuration resources of the slave, according with the application requirements.

C5) Enable the slave to be configured.

C6) Press the Set button and wait for the visualization of the information window that indicates to turn off and on the slave. By executing this step the Demo GMT transmits to slave the 4 configuration resources defined on point C4 and the slave receive and save them.

C7) Turn off and on the slave.

C8) Press the OK button on the information window still visualized and wait the following window that show the result of the performed configuration. With this step the Demo GMT acquires the 4 configuration resources and checks that the read values corresponds to those defined on point C4.

**FIGURE 13: RESULT WINDOW OF SLAVE CONFIGURATION**

C9) Close the result window through the OK button and if configuration has been correctly executed, it can be closed also the configuration window, through the Exit button. Viceversa when errors happens it is suggested to verify the serial connection and then repeat the steps with attention.

**MANUAL TEST**

The manual test modality for GMT remote slave, offered by Demo GMT, allows the interaction with all the hardware resources of the slaves, connected to the PC that execute the program. Through this modality the user is capable to verify the functionality of the serial connection and of all the remote slaves, even for the wiring to the field signals.

Below are listed the sequence of steps that must be executed, to succesfully use this modality. In these steps are always managed more slaves in order to show all the powerfull features of Demo GMT, but the user can naturally work only with a single GMT slave.

T1) Configure all the remote slaves with the same values, but with different slave address.

T2) Arrange the remote slaves and the PC for network communication, by selecting the desidered electric protocol and by interposing the specific serial converter on PC side.

T3) Enable the PC, executes the Demo GMT, select the option File | Set Comm Parameters and insert the data in the displayed window. These data coincide with the serial port connected at point T2, with desidered baud rate and the physic communication protocol defined on the slaves, that is those set on step T1.

T4) Enable all the slave to test.

T5) Select the option Network | Add slave and in the window that requires the new slave, insert the address of the first slave, that is one of the value defined on step T1. Once the insertion has been confirmed, please wait the visualization of the windows related to added slave.
T6) Repeat the step T5 for all the slaves that must be tested, taking care that the same slave address is not inserted more times.

T7) At this point on PC monitor must be shown a window for each one of the added slaves; these windows includes a fixed graphic area, relative to the maximum hardware resources of the slaves, and a variable area relative to the resources that are effectively available on the connected slave. In detail the Demo GMT acquires the hardware identification code from the slave and through its value it decides the slave photo, the digital inputs and digital outputs numbers, the number of counters, A/D, PWM, etc.

T8) By using the manual test windows of the slaves, the user can examine the status of the input resources and set the output resources. In fact the program acquires and shows the slaves status, at regular time period defined by the scan rate frequency, and contemporaneously manages the user intervents on the output resources. For example it is possible to click on the red LED to change the state of the corresponding outputs, click on numeric boxes of PWM and counters to set new values.

T9) Please remind that in this phase the Demo GMT uses the GMT functions to obtain and set the resources status and during each communication it also verifies the communication results, either for exceptions and communication errors. The global communication state is reported on the title bar of each window and during a regular work, it should visualize the message: Communication OK.
T10) During normal use of **Demo GMT**, the user can change the frequency used by the program to interacts with the **GMT** remote slaves, through the specific option **Scan Rate | Set Scan Rate**. Once the option has been selected it is required the new scan rate expressed in multiply of 100 milliseconds.

![Set Scan Rate](image)

**FIGURE 16: SCAN RATE WINDOW SETTING**

T11) Finally it is possible to remove a slave from the interaction process before described, through the option **Network | Remove slave**. The option ask for the address of the slave to remove and then it close the relative window and stop the communication with the same slave. A slave can be easily removed by the tipycal `X` close button, in the high right corner of the window.

**LIMITS OF DEMO GMT**

The **Demo GMT** that is freely distribuited during the first purchase of **GMT**, is a reduced version of the same program, that has the following limits in confront of the complete version:

- maximum number of **GMT** remote slaves = 4
- minimum time for scan rate in remote slaves acquisition = 1 second

Anyway these limits don't prevent the user in the execution of all the operations described in the chapter.
HOW TO START

In this chapter are described the operations necessary to start working with GMT software package. In detail it is reported the correct sequence of operations that must be executed by the user, first to configure and then to basically use the GMT remote devices. The starting phase is moreover simplified by the use of a simplified system available besides any customer, that is the one composed by a PC and a single GMT remote slave, connected in RS 232.

PRELIMINARY OPERATIONS

1) Read carefully all the received documentation.

2) Prearrange the remote slave to work correctly (power supply, right jumpers configuration, etc.)

3) Perform serial connection between GMT slave and PC as described in figure RS 232 POINT TO POINT CONNECTION EXAMPLE, available in the technical manual of the used hardware.

4) Turn on the Personal Computer.

5) If the serial line of PC is on a converter (i.e. USB <-> RS 232 interface), then perform all the operations described in the documentation of the same product, and check its functionality.
CONFIGURATION AND FIRST USE

6) Install on PC the Demo GMT program, as described in INSTALLATION, DEINSTALLATION paragraph.

7) Execute the Demo GMT, that in correspondence of its first execution, is already configured with the English language that can be anyway selected by the command Language | Set language | English.

8) Perform the configuration of the single connected slave, by following the indications of SLAVE CONFIGURATION paragraph, and by defining the 4 configuration resources as:

- Free COMs (those connected at point 3)
- Slave Stop Bit 1
- Slave Parity None
- Slave Address (dec.) 128
- Slave Delay (ms) 0

9) Use the manual test modality of Demo GMT, as described in homonymous paragraph, by selecting the following values for the communication port:

- Free COMs (those connected at point 3)
- Baud Rate 9600
- Stop Bit 2
- Parity None

10) Interact with all the hardware resources of the connected slave and check the communication state that must report no errors (Communication OK).

11) When test and verifies are completed, exit from Demo GMT program and return to Windows operating system, through the dedicated option File | Exit.
APPENDIX A: ALPHABETICAL INDEX

A
A/D  12, 14, 20, 37
Add slave  37
Address  10, 12
Analog inputs  12, 14, 20
Analog outputs  12, 14, 20
Application  4, 27, 30
ASCII  9, 10
Assistance  1
Autodetection  18

B
Baud rate  18, 34, 40
Bit per characters  18
Broadcast  9, 10, 17, 21

C
CD rom  6
Coil  11, 13, 21, 23, 24
Communication errors  16, 37
COMx  8, 34, 35, 40
Concentrators  31
Configuration  18, 31, 35, 40
Configuration resources  13, 18, 36
Connection  29
Connection diagram  7
Container  1
Counters  12, 14, 20, 37
Current Loop  18, 31

D
D/A  12, 14, 20
Data validity  16
Default settings  18
Deinstallation  32
Delay for response  13, 15, 18, 35, 40
Demo GMT  6, 8, 31, 40
Description  9
Destination folder  32
Diffusion  9
Digital inputs  11, 12, 13, 20, 22, 28, 37
Digital outputs  11, 12, 13, 20, 21, 23, 24, 28, 37
Directive  1
Distribution  6
Documentation  1, 4, 6

E
EEPROM  14, 20
Electric protocol  17, 31, 35
Electrostatic noises  1
Encoding  10, 12
English  6, 35
Errors  10, 16, 31, 37
ESD  1
Exception  11, 16, 17, 37
   Code  17
   Message  17

F
Field  10, 11
First use  40
FORCE MULTIPLE COILS  24
FORCE SINGLE COIL  23
Frame  10
Frequency  38
Full duplex  9
Function
   FORCE MULTIPLE COILS  24
   FORCE SINGLE COIL  23
   PRESET MULTIPLE REGISTERS  19, 25
   PRESET SINGLE REGISTER  24
   READ COIL STATUS  21
   READ HOLDING REGISTERS  19, 22
   READ INPUT REGISTERS  23
   READ INPUT STATUS  22
Function code  10, 11
Function data  10, 12
Functions  11, 21

G
General information  4
GMT remote slave  4, 8, 18, 39

H
Half duplex  9
Handshake  31
Hard disk  8
Hardware identification  13, 15, 20, 37
Holding registers  11, 12, 15, 22, 25
Hours  15
House application  27, 30
How to start  39

I
Implementations  20
Information  35
Installation  32
Interfaces  8, 15, 31, 39
Introduction  1
ITALIAN TECHNOLOGY

L
Language 35
LED 37
LImits 18
Limits 12, 38
Link 32, 34
LRC 16, 17

M
Main window 34
Malfunctions 17
Manual test 31, 36
Master 4, 8, 9, 31
Memory 8, 14, 20
Message 10, 16
Monitor 8
Mouse 8

N
Name 20
Net Framework 8, 32
Network 9, 15, 31, 36
Nibbles 10
Noise 16

O
Operating systems 8, 31
Options 18

P
Parity 13, 15, 16, 18, 34, 35, 40
PC 8, 31, 39
Photo 37
Physic protocol 13, 15, 16, 17, 19, 35
Point to point 9, 31
PRESET MULTIPLE REGISTERS 19, 25
PRESET SINGLE REGISTER 24
Protection 1
PWM 14, 20, 37

Q
Query 9, 16

R
READ COIL STATUS 21
READ HOLDING REGISTERS 19, 22
READ INPUT REGISTERS  23
READ INPUT STATUS  22
Registers  11
Remove slave  38
Requirements  8
Resources  12, 13, 14, 18, 20, 25
Response  9, 16
Result  36
RS 232  18, 31, 39
RS 422  15, 18, 31
RS 485  15, 18, 31
RTC  14, 15, 20
Rules  1

S
Safety  1
Scan rate  35, 38
Serial cable  8, 36
Slave  4, 8, 9, 18, 35
Slave address  10, 13, 18, 35, 40
Spikes  16
SRAM  14, 20
Standard  9
Start bit  18
Stop Bits  13, 15, 18, 34, 35, 40

T
Time  15, 38
Time out  16
Trademarks  2

V
Version  3, 13, 15, 35
Video card  8

W
Warranty  1, 2
Web sites  6, 35
Week day  15
Windows xxxx  8, 31

Y
Year  15