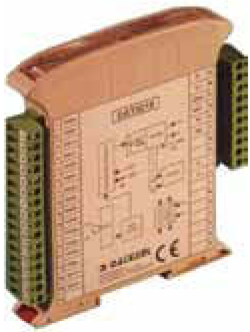
DAT 3010

DATA ACQUISITION MODULE

Universal Analog Input with MODBUS protocol



Characteristics

Sensor to Computer interface
for remote data acquisition
Universal analog inputs for:
Tc, mV, mA, V, RTD, Resistance
Standard MODBUS ASCII/RTU protocol
Remotely configurable input signal
RS485 or RS232 line serial communication
Isolated Galvanically at 2000Vac (3 ways)
EMC compatibility - CE Mark
Housing thikcness of only 17,5 mm.
DIN rail mounting

APPLICATIONS

- Data Acquisition and Controls
- Monitoring of Industrial Process
- Factory and Building Automation
- Distributed Measure and Control

Table of contents

1 - Technical Specifications

2 - Protocol

3 - Operation Principle

Chapter 1

TECHNICAL SPECIFICATIONS

1.1 Introduction
1.2 Pin Assignment & Wiring
1.3 Specifications

1.1 Introduction

The modules of the DAT 3000 series find, for their peculiar characteristics, wide application in the treatment of the signals in the systems for the Automation and the Control of the Industrial Processes. They allows the conditioning of a wide range of signals, both analogical and digital, with an effective protection from the noises.

Thanks to the systematic employment of the microprocessor the DAT3000 modules are able to assure the conditioning of the signal, the supervision of system, the alarm outputs and the reliable generation of the desired value.

The module has been studied for being able to be assembled on the DIN-rail in simple way and with the maximum use of the spaces. In fact, if the conditions of dissipation allow it, the modules can be assembled one beside the other allowing so a considerable reduction of the space occupied. Extractable type screw terminals are employed for the connection. Thanks to this, the user can directly remove the modules so simplifying their maintenance.

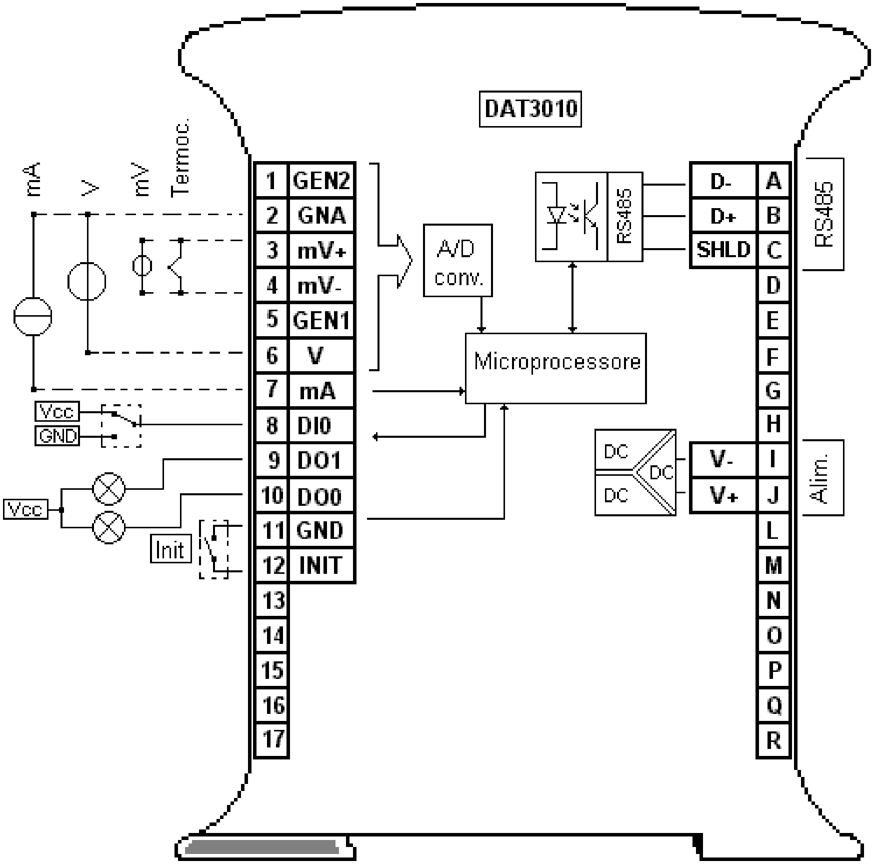
The MODBUS (RTU or ASCII) protocol, known as a spread standard in Field-Bus, is useful for efficient and reliable management of a plant with great quantity of variables. Thank to this standard, it is possible to directly interface DAT3000 series to the larger part of PLCs and SCADA applications available on the market, with the possibility to connect on the same net DAT3000 devices with other different devices (PLC, Operator Panels, CNC, etc...).

DAT3010 Characteristics

The DAT3010 device converts the analog input signal in digital engineering units. It is able to handle a great variety of input signals. It provides also two digital outputs, that can be used as alarms, and a digital input.

- **Universal Analog Input**
  for mV, V, mA, Tc, RTD, Res
- **Digital I/O**
- **Watchdog Alarm**
- **Asyncronous serial data transmission**
- **Speed up to 38,4 Kbps warranted**
- **Distance up to 1,2 Km**
- **DIN-rail mounting housing**
- **Complete isolation at 2000 Vac**
  with optocouplers and DC/DC converter

1.2 DAT 3010 Pin Assignment

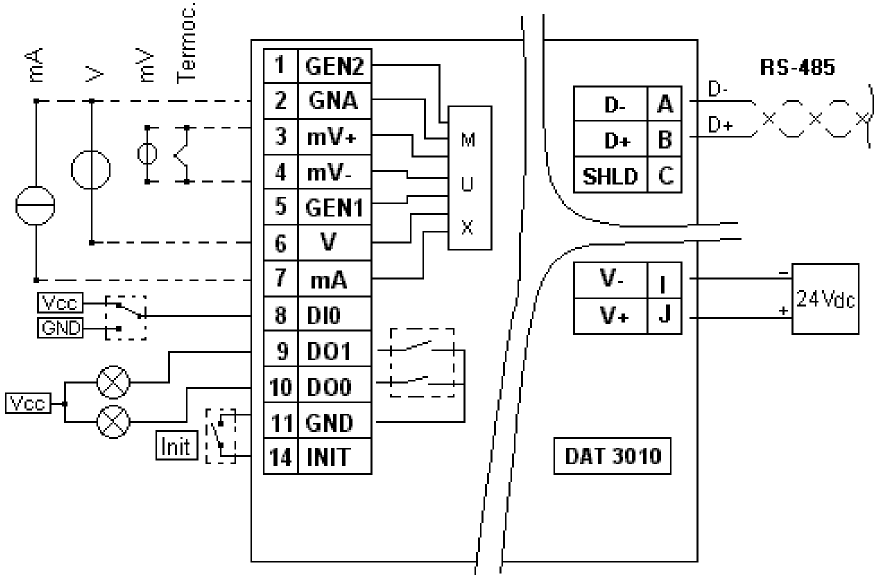


Pin Assignment Table:

PIN	Name	Description
A	D -	D - RS485
B	D+	D + RS485
C	SHLD	RS485 Shield
I	V -	Power Supply Ground
J	V+	+24V Power Supply

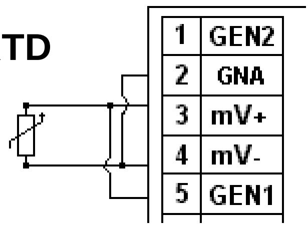
PIN	Name	Description
1	GEN2	Current Generator #2
2	GNA	RTD, V, mA Reference
3	mV+	Differential Input +
4	mV-	Differential Input -
5	GEN1	Current Generator #1
6	V	Volt Input
7	mA	mA Input
8	DI0	Digital Input
9	DO1	Digital Output #1
10	DO0	Digital Output #0
11	GND	INIT and DI/O Ground
12	INIT	INIT Input

Wiring

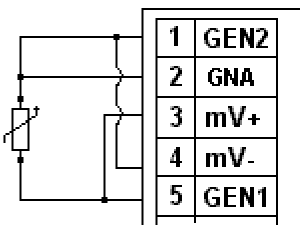


RTD Connections:

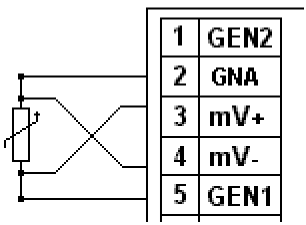
2-wire RTD



3-wire RTD



4-wire RTD



1.3 Technical Specifications

(typical @ 25°C and under nominal conditions)

Analog Input:

Channels: 1

TC		
Input	Min	Max
J	-210 °C	+1200 °C
K	-210 °C	+1372 °C
R	-50 °C	+1767 °C
S	-50 °C	+1767 °C
B	+400 °C	+1825 °C
E	-210 °C	+1000 °C
T	-210 °C	+400 °C
N	-210 °C	+1300 °C

Tensione		
Input	Min	Max
+/- 50 mV	- 50 mV	+ 50 mV
+/- 100 mV	- 100 mV	+ 100 mV
+/- 500 mV	- 500 mV	+ 500 mV
+/-1000 mV	-1000 mV	+1000 mV

Corrente *		
Input	Min	Max
+/- 20 mA	- 20 mA	+ 20 mA

RTD		
Input	Min	Max
PT100	-200 °C	+850 °C
PT1000	-200 °C	+200 °C
NI100	-80 °C	+180 °C
NI1000	-60 °C	+150 °C

Resistance		
Input	Min	Max
Low	0 Ohm	500 Ohm
High	0 Ohm	2000 Ohm

Sampling Rate	10 samples/sec.
Rise Time	0.3 sec.
Input Impedance	
TC, mV	> 10 Mohm
10 V	> 1 Mohm
Lead wire resistance influence	
TC, mV, V	0.8 uV/ohm
RTD 3-wires	0.05% /ohm (50 ohm/wire max.)
RTD 4-wires	0.005% /ohm (100 ohm/wire max.)
RTD excitation current	0.350 mA typ.
Linearity	TC ±0.2% F.S.
	RTD ±0.1% F.S.
Accuracy	
RTD(100)	±0.05% F.S.
RTD(1000)	±0.1% F.S.
Res.(2000 ohm)	±0.2% F.S
Res.(500 ohm)	±0.1% F.S
mV, V	±0.05% F.S.
Tc	the larger of ±0.05% F.S. and ±5uV
mA	±0.05% F.S.
Cold Junction Comp.	±0.5 °C
Thermal Drift	
Full Scale	±0.005 %/°C
Cold Junction Comp.	±0.02 °C/°C

Digital Input:
Channels: 1

ON State	+10V to +30V
OFF State	0V to +5V
Input Impedance	4,7Kohm

Digital Output:
Channels: 2

Open Collector	30V, 30mA max. load
Power dissipation	0.3W

Power Supply:

Supply Voltage	+10 to +30 Vdc
Power Consumption	1W @ 24V typ.
Isolating Voltage	2000 Vac for 60 sec.
Polarity Protected	60 Vdc

Temperature and Humidity

Ambient Temperature	-10 ÷ +60 °C
Store Temperature	-30 ÷ +90 °C
Hmidity (not condensing)	0 ÷ 90 %

EMC

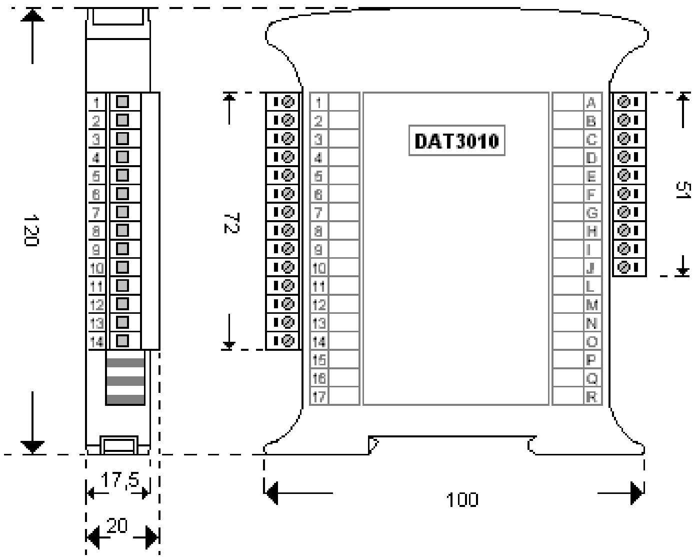
Emission	EN50081-2
Immunity	EN50082-2
RF Immunity	tested @ 10 V/m up to 1000 Mhz

Housing

Material	Selfestinguishing plastic
Mounting	DIN Rail
Weight	Approx. 100 g.

Dimensions
(W x H x T) in mm: 100 X 120 X 17,5

Mechanical Dimensions



Chapter 2
PROTOCOL

2.1 Settings
2.2 MODBUS
2.3 Tables
2.4 Registers, Coils & Functions
2.5 LRC & CRC Generation

2.1 Settings

Default Settings :

Baud Rate	**=**	**9600 bps**
Address	**=**	**1**
Protocol	**=**	**RTU**

Character Format :

Parity	**=**	**None**
Bit Number	**=**	**8**
Stop Bit	**=**	**1**

2.2 MODBUS

Introduction :

MODBUS was used principally in industrial environments thanks to the strong contribute of Modicon in PLC market, that developed this protocol as a standard for its devices.
When the specifications has been open and public, the Modbus protocol was adopted in many automation environments and extended to other industrial fields.
Becoming a “standard de facto”, Modbus can be implemented in any intelligent device (IFD - Intelligent Field Device):
programmable controllers, CNC, drivers, Human-Machine interfaces, measure instruments, etc...

Comunication model :

The connection is made by the multipoint RS-485, throug a twisted pair cable.
The comunication type is a multi-point, half-duplex, Master-Slave, where the Master (tipically a PC) can send a request (“Query”), than the Slaves answers with a message (“Response”) only at the queries to them individually addressed.
Modbus protocol define the query format, composed by the address of the queried slave, a function code that define the requested operation, other fields for the data exchange (coils, registers, etc...), and a field for the communication error checking (CRC).
Slaves answers with a message composed as the quesry; when the slave can’t to develop the requested operation, it will send an error message.

Transmission mode :

This devices can be configured for the standard Modbus comunication, with two transmission modalities: ASCII or RTU. The user can select the desired mode and the serial port settings (baud-rate, parity, etc...) during the device configuration. The communication mode and the serial port setting must be the same for all the devices connected on the net.

ASCII mode

In the ASCII modality (*American Standard Code for Information Interchange*),
the data, composed by 8-bit bytes are sent as two 8-bit ASCII characters. This modality permits time lapses between characters up to one second, without generate an error.

The byte format in ASCII mode is:

Code system: Hexadecimal, ASCII characters 0-9 A-F
Each hexadecimal character corresponds to an ASCII character of the message. Two charascters corresponds to a byte of data.

Character format:
1 bit start
7 bit data, the first bit sent is the last significative
1 bit parity Even, Odd, Mark or Space
1 bit stop

Error Check Field: Longitudinal Redundancy Check (LRC)

RTU mode

In the RTU modality (*Remote Terminal Unit*), the data, composed by 8-bit bytes are sent as one 8-bit binary character. This modality permits better features for the quality and the higher rate of transmission compared to ASCII modality. Each message must be transmitted without interruptions between characters.

The byte format in ASCII mode is:

Code system: 8 bit binary character, 00 to FF
Each 8-bit data correspond to one character of the message.

Character format:
1 bit start
8 bit data, the first transmitted is the last significative.
None parity
1 bit stop

Error Check Field: Cyclical Redundancy Check (CRC)

Modbus message Framing :

ASCII Framing

In ASCII modality, the message begin with the “ : ” charactere (ASCII 3A hex), and termitate with a CR-LF (“*Carriage-Return, Line-Feed*”) character sequence (ASCII 0D & 0A hex).
Permitted characters in other fields are Hexadecimals:
“0” to “9” and “A” to “F”.
Devices in the net starts to monitor the message after receiving the “ : ” character. When a device (slave) recognizes this character, it decode next field with the address, to recognize the mesages to its addressed.
Between the transmission of two characters can be there up to one second time intervals; if this time is longer, the receiving device notices a communication error.

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS

RTU Framing

In RTU modality, the message can begin after a 3.5 character time of *silence* (without data transmission); this time is then depending of the communication baud-rate.
Devices on the net costantly monitor the bus.
Permits characters in all fields are 8-bits numbers, then 0 to 255.
The first field transmitted is the slave address.
When the first field (address) is received, each device decode it, to recognize the messages to its addressed.
After the last character transmittet, must be there a silence time of at least 3.5 characters that identify the end of the message. A new message can start after this interval. If a new message begin before this time, the receiving device will consider it as a continuum of the precedent mesage; this will generate an error, because the the CRC value will not to be valid.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	1 CHAR (8 bits)	1 CHAR (8 bits)	n CHARS (n x 8 bits)	2 CHARS (16 bits)	T1-T2-T3-T4

Data addressing in Modbus messages:

The data exchanged between master and slave can be:
“Coil” (1 coil = 1bit) and “Registers” (1 register = 16bit).
Each coil and each register correspond to a predefined address.
Data are organized in banks with the following addresses:

0xxxx	Input Coils
1xxxx	Holding Coils
3xxxx	Input Registers
4xxxx	Holding Registers

In the Data field of the message data are addressed from 0.
For example, the first holding register has the ‘40001’ address and it is addressed as ‘0000’, the register 40108 is adressed as ‘006B’ hex (107 decimal).
In the Function field, each code address to a specify bank, i.e. the 06 function is referred to Holding registers, then the‘4xxxx’ is implicit.

2.3 ADDRESS TABLES

All the data variables shared by a Modbus module are shown in tables, where each data is linked to an address.
Each data can be of two types:
- "COIL", composed by a single bit, can be associated to digital input (switch), digital outputs (relays) logic states (alarms).
- "REGISTER", composed by 2 bytes (16 bits), can be associated to analog input or outputs, variables, set-point, etc...

A register can also include the image of more coils, for example the 16 digital inputs of a device can be read and write as bit, one by one, addressing the input relative coil , or they can be read or write as a single 16-bit port addressing the associated register, where for example the last significant bit will respond to the first coil.

In the Modbus protocol, coils and registers are divided in banks:

0xxxx	=	Memory Coils	(read/write)
1xxxx	=	Input Coils	(read only)
3xxxx	=	Memory Registers	(read/write)
4xxxx	=	Input Registers	(read only)

DAT3000 modules don't make distinction between Memory and Input banks, then every coil can be addressed as 0xxxx or 1xxxx; every register can be addressed as 3xxxx or 4xxxx.
In this way the functions 01 (Read Memory Coils) and 02 (Read Input Coils) make the same funcion.

REGISTER TABLE

Register	Register Name	Low Limit	High Limit	Access	EP
40001	Test	N.A.		R/W	
40002 40003	Firmware Version	ASCII		R	
40004 40005	Module Name	ASCII		R/W	*
40006	Communication	0x0000	0x007F	R/W	*
40007	Address	0x0001	0x00F7	R/W	*
40008	Rx/Tx Delay	0x0000	0x00FF	R/W	*
40009	Analog Input Lo	N.A.		R	
40010	Analog Input Hi	N.A.		R	
40011	Analog Input Int	N.A.		R	
40012	Analog Input Sync	N.A.		R	
40013	CJC	-32.000	+32.000	R	
40014	WatchDog Timer	0x0000	0x00FF	R/W	*
40015	Hi Threshold	N.A.		R/W	*
40016	Lo Threshold	N.A.		R/W	*
40017	CJC Offset	-32.000	+32.000	R/W	*
40018	Input Type	0x0000	0x001A	R/W	*
40019	Coils	N.A.		R/W	*
40020	Hysteresis	0x0001	0x00FF	R/W	*

- Bank 3xxxx registers are a mirror of bank 4xxxx registers.
- To write a value out of allowed limits will cause a default value writing.
- Registers marked on the “EP” column are saved in eprom each time they are written.

COILS TABLE

Indirizzo COIL	Nome COIL	Accesso	EP
00001	Out 0 Safe Value	R/W	*
00002	Out 1 Safe Value	R/W	*
00003	Out 0 PowerUp	R/W	*
00004	Out 1 PowerUp	R/W	*
00005	Threshold Alarm Type	R/W	
00006	Threshold Alarm Enable	R/W	
00007	WatchDog Alarm Enable	R/W	
00008	--	R/W	
00009	Out 0	R/W	
00010	Out 1	R/W	
00011	In 0	R	
00012	Open Detect	R	
00013	WatchDog Event	R/W	
00014	PowerUp Event	R/W	
00015	Hi Threshold Alarm	R/W	
00016	Lo Threshold Alarm	R/W	

- Bank 0xxxx coils are a mirror of bank 1xxxx coils.
- Coils marked on the “EP” column are saved in eprom each time they are written.

2.4 REGISTERS DESCRIPTION

40001 : Test

Write a number in this register commands the module to perform a function as shown in the table below:

Value	Function
10	Perform Synchronized Sampling
20	Perform ZERO Calibration
30	Perform SPAN Calibration

The user can use this register in broadcast command to perform the Synchronized Sampling.
Attention! The Calibration can take effect only in INIT condition.

40002 / 40003 : Firmware Version

Read-only 4 byte field that contains the Datexel identifier of the module firmware version.

40004 / 40005 : Module Name

4 byte or 4 ASCII characters of free-use, can contain the name of the device or a string to identify its function in the system. Each byte can contain any value.
The default value of this field contains the Datexel module type in ASCII characters (i.e.. “3010”).

40006 : Communication

Set each bit of this register as shown in the table below, to set baud-rate, bit number, parity and communication mode of the module:

Bit 0	Baud Rate B0	See Table
Bit 1	Baud Rate B1	
Bit 2	Baud Rate B2	
Bit 3	N.ro bit	0 = 7 bit 1 = 8 bit
Bit 4	Parity P0	See Table
Bit 5	Parity P1	
Bit 6	Mode	0 = ASCII 1 = RTU
Bit 7..15	--	Not used

Baud Rate	B2	B1	B0
1200	0	0	0
2400	0	0	1
4800	0	1	0
9600	0	1	1
19200	1	0	0
38400	1	0	1
38400	1	1	0
38400	1	1	1

Parity	P1	P0
Mark	0	0
Even	0	1
Odd	1	0
Space	1	1

For example, the value 0x003D set the module at 38,4 kbps, 7 bit, Space parity, ASCII mode.
Bit number is ignored, because it is fixed to 7 in ASCII mode and it is fixed to 8 in RTU mode. In RTU mode the parity is fixed to None.
Default setting is 9600 bps, RTU mode.

40007 : Address

Contains the field address of the module. Allowed address are 1 to 255
Default address is 1.

40008 : Rx / Tx Delay

Contains the delay value between the receiving of a query and the transmission of a response expressed in 2 ms steps.
0x0001 = 2 ms.
0x00FF = 255x2 = 510 ms.

40009 / 40010 : Analog Input (float)

Value sampled in input expressed in 32-bit Floating Point.

40011 : Analog Input (int)

Value sampled in input expressed in integer (16 bit) with sign (most significant bit). The decimal numbers depends on the input type:

Ingresso	Decimali	Formato
+/- 50 mV	3	+50.000
+/- 100 mV	2	+100.00
+/- 250 mV	2	+250.00
+/- 1000 mV	1	+1000.0
+/- 20 mA	3	+20.000
Tc J .. Tc N	1	+1200.0

Ingresso	Decimali	Formato
RTD	1	+850.0
Res	1	+2000.0

40012 : Analog Input (sync)

When Sinchronized Sampling is performed, sampled values are stored in these registers.

40013 : CJC

The value of Cold Junction Compensation, expressed in 16-bit Integer with 2 decimals (hundredth of degrees).
0x09C4 = 2500 = 25.00°C

40014 : WatchDog Timer

Contains WatchDog Timer value, expressed in 0.5 sec. steps.
If the WatchDog is enabled and the module don’t receive commands during a time equal the value of this register, a Watchdog Event alarm happens.
0x0001 = 0.5 sec.
0x00FF = 255x0.5 = 127.5 sec.

40015 : Hi Threshold

Sets the High Threshold Value.

40016 : Lo Threshold

Sets the Low Threshold Value.

40017 : CJC Offset value

Sets the offset value for the Cold Junction Compensation calibration. The value is expressed in hundredth cents of degree:
0x0001 = 0.01°C
0x7FFF = +327.67°C
0x8000 = -327.67°C

40018 : Input type

Selects type and range of the input, as shown in the table below.

Value	Input type
0x0001	+/- 50 mV
0x0002	+/- 100 mV
0x0003	+/- 250 mV
0x0004	+/- 1000 mV
0x0005	+/- 10 V
0x0006	+/- 20 mA
0x000E	Tc J
0x000F	Tc K
0x0010	Tc T
0x0011	Tc E
0x0012	Tc R
0x0013	Tc S
0x0014	Tc B
0x0015	Tc N

Value	Input type
0x0007	Res. 0~2000 Ohm
0x0008	Res. 0~500 Ohm
0x0017	Pt100
0x0018	Ni100
0x0019	Pt1000
0x001A	Ni1000

40019 : Coils

This register is the mirror of the Coils Table. It is possible to use this register to read and write at the same time all the coils without the necessity to implement the coils read/write functions (01-02-15). Writing the register, read only coils are disguised.

Registro	COIL corrispondente (000xx)															
40019	8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9

40020 : Hysteresis

Sets the value for the Hysteresis used in the Threshold Alarm.

COILS DESCRIPTIONS

00001 : Out 0 Safe Value

When the Wathdog Alarm is active, the Output #0 will go in the specified logic level.

00002 : Out 1 Safe Value

When the Wathdog Alarm is active, the Output #1 will go in the specified logic level.

00003 : Out 0 PowerUp Value

At the Power Up of the module, the Output #0 will go in the specified logic level.

00004 : Out 1 PowerUp Value

At the Power Up of the module, the Output #1 will go in the specified logic level.

00005 : Threshold Alarm Type

Sets the Threshold alarm type.
1 = Momentary
0 = Latch

00006 : Threshold Alarm Enable

Enable the Threshold Alarm.
1 = Enabled
0 = Disabled

00007 : WatchDog Timer Enable

Enable the WatchDog Timer Alarm.

00008 : free

This coil is free.

00009 : Out 0

Commands the Out 0 output port.
1 = ON 0 = OFF

00010 : Out 1

Commands the Out 1 output port.
1 = ON 0 = OFF

00011 : IN 0

Reads the IN 0 input port.
This coil is read-only.
1 = ON 0 = OFF

00012 : Open Detect

When the sensor connected to the input goes in open condition, this coil is forced to high logic level.
1 = Open sensor 0 = Sensor Connected

00013 : WatchDog Event

Indicates the WatchDog Alarm state. To reset the alarm, set this coil to 0.
1 = Alarm condition 0 = Normal condition

00011 : PowerUp Event

Indicates that the module has beeing turned off and on or has beeing resetted. This coils is forced to 1 at each power-on.

1 = Reset happened 0 = Reset not happened

00015 : Hi Threshold

If the Threshold Alarm is enabled and the sampled value overcomes the Hi Threshold value, this coil will be forced to 1.

1 = Alarm Condition 0 = Normal Condition

00016 : Allarme Basso

If the Threshold Alarm is enabled and the sampled value is lower of the Lo Threshold value, this coil will be forced to 1.

1 = Alarm Condition 0 = Normal Condition

Supported Modbus Functions

01 : Read Memory Coils

Reads the ON/OFF state of many memory coils.
- Coils are addressed from 0 : coils 1-16 are addressed as 0-15
- Coils 0xxxx are a mirror of coils 1xxxx.
- Broadcast not allowed.

QUERY form

Example: Read coils 3..12 of slave # 11h:

Add.	Fun.	DATA			
		Start Hi	Start Lo	Num Hi	Num Lo
11	**01**	**00**	**02**	**00**	**0A**

“Start” contains the address of the first coil (3, addressed as 02).
“Num” contains the amount of requested coils (10 coils = 0A).

RESPONSE form

Add.	Fun.	DATA		
		Byte Count	Data (8-1)	Data (16-9)
11	**01**	**02**	**56**	**01**

“Byte Count” contains the count of bytes (8 bits) transmitted in the DATA field.
“Data” is composed by n bytes, the first with the first 8 coils requested.
If the number of coils requested is not a multiple of 8, the most significant bits of the last byte will be forced to 0.

02 : Read Input Coils

Reads the ON/OFF state of more input coils. Such coils 0xxxx are a mirror of coils 1xxxx, the 02 function is equal to 01 function.

03 : Read Memory Registers

Reads the value of many memory registers.

- Registers are addressed from 0:
  registers 1-16 are addressed as 0-15
- Registers 3xxxx are a mirror of registers 4xxxx.
- Broadcast is not allowed.

QUERY form

Example: Read of registers from 5 to 7 of slave # 01h:

Add.	Fun.	DATA			
		Start Hi	Start Lo	Num Hi	Num Lo
01	**03**	**00**	**04**	**00**	**03**

"Start" contains the address of the first register (5, addressed as 04).
"Num" contains the amount of requested registers (3 registers = 03h).

RESPONSE form

Add.	Fun.	DATA						
		Byte Count	Data0 LO	Data0 HI	Data1 LO	Data1 HI	Data2 LO	Data2 HI
11	**01**	**06**	**12**	**34**	**56**	**78**	**9A**	**BC**

"Byte Count" contains the number of bytes transmitted in the Data field (3 reg.s x 16bits = 6 bytes).
"Data" is composed by n bytes, 2 for each register (the first with the high part of the register).

04 : Read Input Registers

Reads the value of many input registers. Registers 3xxxx are a mirror of registers 4xxxx, the 04 function is equal to 03 function.

05 : Write Single Coil

Set the logical state of a single coil to 0 or 1.
- Coil are addressed from 0 : coils 1-16 are addressed as 0-15
- Coils 0xxxx are a mirror of coils 1xxxx.

QUERY form

Example: Set the coil 14 of slave 01h to 1:

Add.	Fun.	DATA			
		Add Coil	Add Coil	Force Hi	Force Lo
01	**05**	**00**	**0D**	**FF**	**00**

“Address Coil” contains the address of coil (14, addressed as 0D).
“Force” defines the logical set of coil.
Allowed values are:
Force = FF 00 set the coil to 1
Force = 00 00 set the coil to 0
Other values are not allowed.

RESPONSE form

Add.	Fun.	DATA			
		Add Coil	Add Coil	Force Hi	Force Lo
01	**05**	**00**	**0D**	**FF**	**00**

If the query is correct, the response is an echo of the query.

06 : Write Single Register

Force the value of a single register
- Registers are addressed from 0:
  registers 1-16 are addressed as 0-15
- Registers 3xxxx are a mirror of registers 4xxxx.

QUERY form

Example: Write the value 137 in register 14 of slave 01h:

Add.	Fun.	DATA			
		Add Reg	Add Reg	Value Hi	Value Lo
01	**06**	**00**	**0D**	**00**	**89**

“Register Address” contains the address of the register (14, addressed as 0D).
“Value” defines the value to write in the register
The values out of the allowed ranges are not accepted

RESPONSE

Add.	Fun.	DATA			
		Add Reg	Add Reg	Value Hi	Value Lo
01	**06**	**00**	**0D**	**00**	**89**

If the query is correct, the response is an echo of the query.

15 : Write multiple coils

Set the logical state of many coils to 0 or 1.
- Coil are addressed from 0 : coils 1-16 are addressed as 0-15
- Coils 0xxxx are a mirror of coils 1xxxx.

QUERY form

Example: Write coils from 1 to 12 of slave 01h to 1:

Add.	Fun.	DATA						
		Start Hi	Start Lo	Num Hi	Num Lo	Byte Count	Data (8-1)	Data (16-9)
01	**15**	**00**	**00**	**00**	**10**	**02**	**FF**	**0F**

“Start” contains the address of the first coil (1, addressed as 00).
“Num” defines the number of coils to write (12 coils = 10h).
“Byte Count” indicates the number of bytes transmitted on the Data field.

RESPONSE form

Add.	Fun.	DATA			
		Start Hi	Start Lo	Num Hi	Num Lo
01	**15**	**00**	**00**	**00**	**10**

The response indicates the start coil and the number of coils written.

16 : Write multiple Registers

Set the value of many registers.
- Registers are addressed from 0:
  registers 1-16 are addressed as 0-15
- Registers 3xxxx are a mirror of registers 4xxxx.

QUERY form

Example: Write registers from 1 to 3 of slave 01h:

Add.	Fun.	DATA										
		Start Hi	Start Lo	Num Hi	Num Lo	Byte Count	Data0 Hi	Data0 Lo	Data1 Hi	Data1 Lo	Data2 Hi	Data2 Lo
01	**16**	**00**	**00**	**00**	**03**	**06**	**12**	**34**	**56**	**78**	**9A**	**BC**

“Start” contains the address of the first register (1, addressed as 00).
“Number” specify the number of registers to write (3).
“Byte Count” indicates the number of bytes transmitted on the Data field.
The values written in registers are:
40001 = 0x1234
40002 = 0x5678
40003 = 0x9ABC

RESPONSE form

Add.	Fun.	DATA			
		Start Hi	Start Lo	Num Hi	Num Lo
01	**16**	**00**	**00**	**00**	**03**

The response indicates the start register and the number of registers written.

08 : Diagnostic

It provides diagnostic functions to verify the functionality of the module.
Data field is composed by 2 bytes for the sub-function and 2 bytes for data.
Implemented sub-functions:

00 = Data Loop. Response is an echo of the received message

QUERY form

Example: Data Loop request.

Add.	Fun.	DATA			
		SubF Hi	SubF Lo	Data Hi	Data Lo
01	**08**	**00**	**00**	**00**	**00**

"SubFun" contains the identifier of the Sub-function.
"Data" specify the value to be re-transmitted.

RESPONSE form

Add.	Fun.	DATA			
		SubF Hi	SubF Lo	Data Hi	Data Lo
01	**08**	**00**	**00**	**00**	**00**

If the query is correct, the response is an echo of the query.

2.5 LRC generation

The Longitudinal Redundancy Check (LRC) field is one byte, containing an 8–bit binary value.

The LRC value is calculated by the transmitting device, which appends the LRC to the message. The receiving device recalculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results.

The LRC is calculated by adding together successive 8–bit bytes in the message, discarding any carries, and then two's complementing the result. The LRC is an 8–bit field, therefore each new addition of a character that would result in a value higher than 255 decimal simply 'rolls over' the field's value through zero. Because there is no ninth bit, the carry is discarded automatically.

A procedure for generating an LRC is:

1. Add all bytes in the message, excluding the starting 'colon' and ending CRLF. Add them into an 8–bit field, so that carries will be discarded.

2. Subtract the final field value from FF hex (all 1's), to produce the ones–complement.

3. Add 1 to produce the twos–complement.

When the the 8–bit LRC (2 ASCII characters) is transmitted in the message, the high–order character will be transmitted first, followed by the low–order character.
For example, if the LRC value is 4F hex (0100 1111):

:	ADDR	FUNC	DATA	DATA	DATA	DATA	LRC Hi	LRC Lo	CR	LF
	0 A	0 1	0 4	A 1	0 0	0 1	4	F		

CRC generation

The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16–bit binary value.
The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16–bit register to all 1's. Then a process begins of applying successive 8–bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits, and the parity bit, do not apply to the CRC.
During generation of the CRC, each 8–bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.
This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8–bit character is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the characters of the message have been applied, is the CRC value.

A procedure for generating a CRC is:

1. Load a 16–bit register with FFFF hex (all 1's). Call this the CRC register.
2. Exclusive OR the first 8–bit byte of the message with the low–order byte of the 16–bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the LSB), zero–filling the MSB. Extract and examine the LSB.
4. (If the LSB was 0): Repeat Step 3 (another shift).
   (If the LSB was 1): Exclusive OR the CRC register with the polynomial value A001 hex (1010 0000 0000 0001).
5. Repeat Steps 3 and 4 until 8 shifts have been performed. When this is done, a complete 8–bit byte will have been processed.
6. Repeat Steps 2 through 5 for the next 8–bit byte of the message. Continue doing this until all bytes have been processed.
7. The final contents of the CRC register is the CRC value.
8. When the CRC is placed into the message, its upper and lower bytes must be swapped as described below.

When the 16–bit CRC (two 8–bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte.
For example, if the CRC value is E1A9 hex (1110 0001 1010 1001):

:	ADDR	FUNC	DATA	DATA	DATA	DATA	CRC Hi	CRC Lo
	0 1	0 4	0 0	0 3	7 E	0 5	E 1	A 9

Chapter 3

OPERATION PRINCIPLES

3.1 INIT

If the exact configuration of a module is unknown, it can result impossible to establish a communication with it. The INIT pin gives a solution to this problem. In fact, connecting the INIT pin to the GND pin (ground) and turning on the device, it will automatically goes in the default configuration:

address	=	01
baud-rate	=	9600, n, 8, 0
mode	=	RTU

The user can send the "read configuration" command to the module and it will answer supplying the configuration stored into the memory; by this way the user knows the exact configuration of the module. If the INIT pin is disconnected from GND, at the next power-up the module automatically returns to the original configuration, and the communication can restart.

3.2 Threshold Alarm

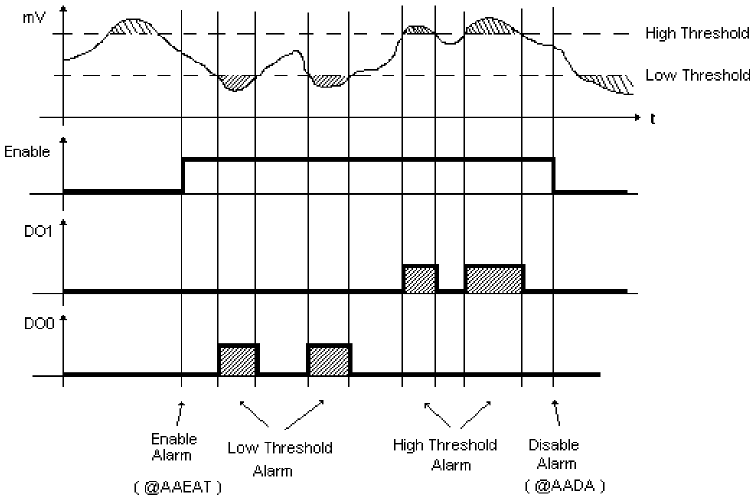
The DAT3010 module is able to autonomously manage the two digital output ports, relating the status of the outputs to the sampled value of the analog input.The module creates a condition of alarm when the value overcomes established threshold values, set by the user.
To use this function it is necessary first of all to set the thresholds. Write the desired values in 40015 (Hi Threshold) and 40016 (Lo Threshold). At this point you must decide if the alarm has to be Momentary or Latch type, and activate the alarm. Set coils 00005 and 00006.
After this operation, when the value of input signal overcomes the high threshold, the high alarm is activated and the DO1 output is forced to logical level 1.

Likewise, when the value goes down under the low threshold, the low alarm is activated and the DO0 output is forced to logical level 1.
When the value returns to a level included between the two thresholds, the alarm resets and the outputs returns to logical level 0.

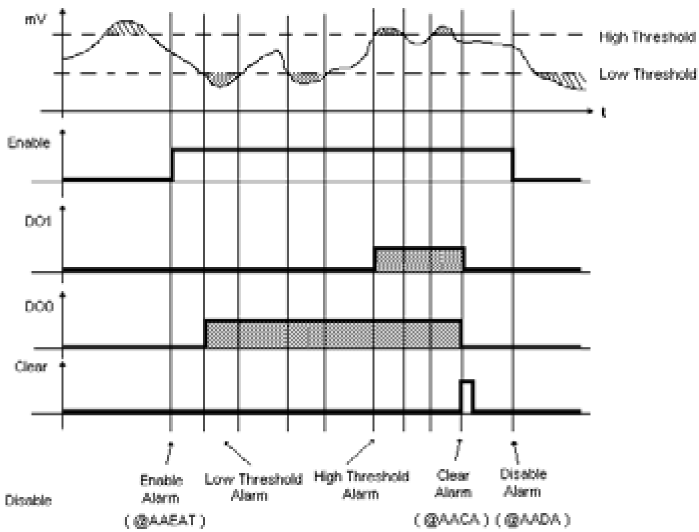
To avoid unacceptable oscillations, the module applies to the input signal an hysteresis which value depend on input type signal.

The outputs are automatically reset if the alarm has been set as Momentary. If the alarm is Latch type, the outputs will stay active until the user won't reset the corresponding coils (00015 and 00016).
In every case, from the moment in which the alarm is enabled, the outputs are exclusively commanded by the module. These coils will become available as soon as the alarm will be disabled

Momentary Alarm example:



Latch Alarm example:



3.3 Dual WatchDog

The DAT3000 modules has been designed for working in industrial environments, where many noises or energy transient can be exist.
If these noises are too much high, the modules could suffer of them and can work incorrectly, even being not able anymore to communicate with the host. This situation can be very dangerous especially in the case that the module commands some attuator because it should do it in a not correctly way. For this reason the modules has been provided of a Host Watchdog timer which, when it is enabled, makes to start the alarm each time the communication between the module and the host is inactive for a period time greater then the programmed one. When the alarm goes on, the values of the outputs are automatically converted to the values set as 'safety value', that corresponding to the state in which the outputs must be putted, and therefore the attuators are putted, to avoid damages to the system in case of failure.

There is also a Module Watchdog timer. This is a hardware circuit that watchs the operation of the internal CPU and operates when, expired the established of time, reset doesn't come from the same CPU. If the CPU breaks for some reasons and no reset of the Module Watchdog happens within the preset time, it starts and resets the CPU, making to restart again its program. After the reset of the module, the digital outputs will assume their initial value (default), which may not to be identical to the value of the outputs before the reset. For this reason the user, after the reset of the module, has to resend the output setting command.

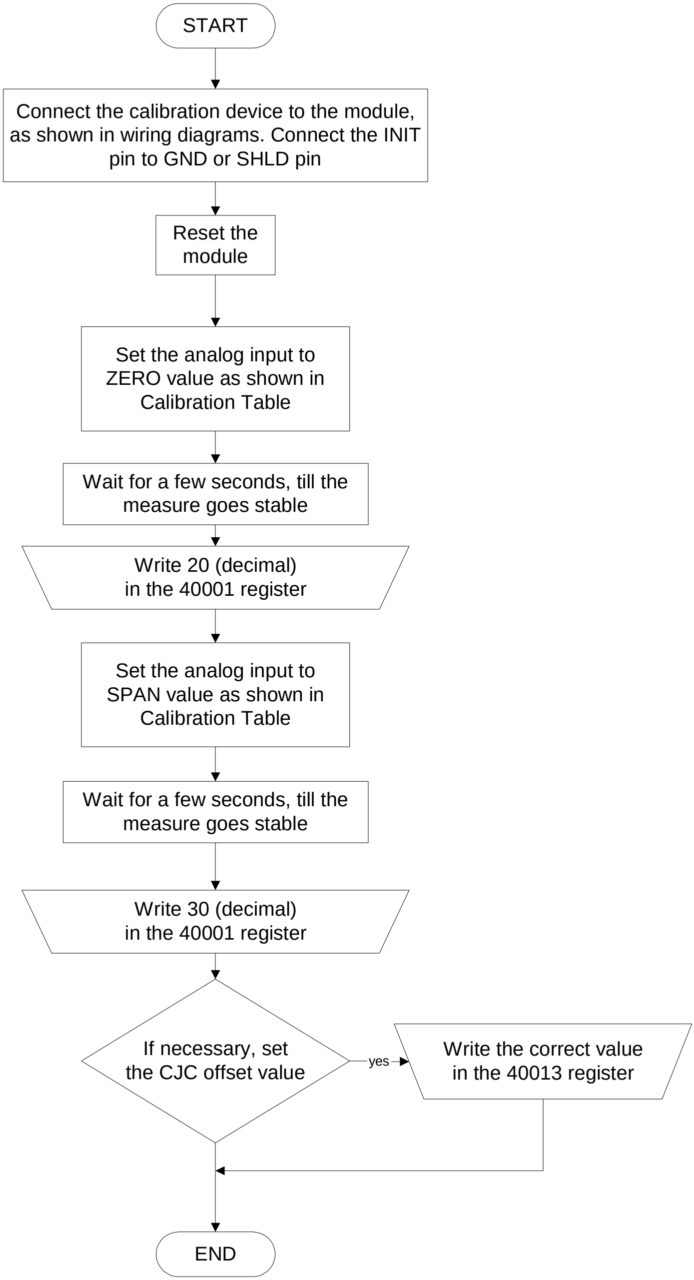
3.4 Calibration

The procedure of calibration is performed in factory on all the modules during the testing phase. However it can be convenient to make another calibration of the module according to the requirements of the user.

To make this, it is necessary to use precision instruments and to correctly perform all the necessary steps, because any error reduces the accuracy and the good operation of the device.

Furthermore this operation must be carried out following the procedure and the data illustred in the following pages.

If a device able to simulate the sensor (calibrator) is not available, it is possible to replace it by applying the corresponding value in mV (for the Tc) or in Ohm (for the RTD).


START
Connect the calibration device to the module, as shown in wiring diagrams. Connect the INIT pin to GND or SHLD pin
Reset the module
Set the analog input to ZERO value as shown in Calibration Table
Wait for a few seconds, till the measure goes stable
Write 20 (decimal) in the 40001 register
Set the analog input to SPAN value as shown in Calibration Table
Wait for a few seconds, till the measure goes stable
Write 30 (decimal) in the 40001 register
If necessary, set the CJC offset value
yes
Write the correct value in the 40013 register
END

Calibration Table:

TT	Tipo	ZERO	SPAN
01	50 mV	+0.000 mV	+25.000 mV
02	100 mV	+0.000 mV	+100.00 mV
03	500 mV	+0.000 mV	+250.00 mV
04	1000 mV	+0.000 mV	+1000.0 mV
05	10 V	+0.000 V	+10.000 V
06	20 mA	+0.000 mA	+20.000 mA
07	Res.H	0.0 Ohm	2000.0 Ohm
08	Res.L	0.0 Ohm	500 Ohm
0E	Tc J	0.0 °C 0.000 mV	+1200.0 °C 69.536 mV
0F	Tc K	0.0 °C 0.000 mV	+1370.0 °C 54.807 mV
10	Tc T	0.0 °C 0.000 mV	+400.0 °C 20.869 mV
11	Tc E	0.0 °C 0.000 mV	+1000.0 °C 76.358 mV
12	Tc R	0.0 °C 0.000 mV	+1760.0 °C 21.006 mV
13	Tc S	0.0 °C 0.000 mV	+1760.0 °C 18.612 mV
14	Tc B	0.0 °C 0.000 mV	+1820.0 °C 13.814 mV
15	Tc N	0.0 °C 0.000 mV	+1300.0 °C 47.502 mV
17	Pt100	0.0 °C 100.0 Ohm	+850.0 °C 390.26 Ohm
18	Ni100	0.0 °C 100.0 Ohm	+180.0 °C 223.22 Ohm
19	Pt1000	0.0 °C 1000.0 Ohm	+200.0 °C 1785.3 Ohm
1A	Ni1000	0.0 °C 1000.0 Ohm	+150 °C 1986.6 Ohm

ED. 02.03 REV. 00

VIA OSLAVIA, 21
21049 TRADATE (VA) - ITALY
TEL. 0331/841070 - FAX 0331/841950