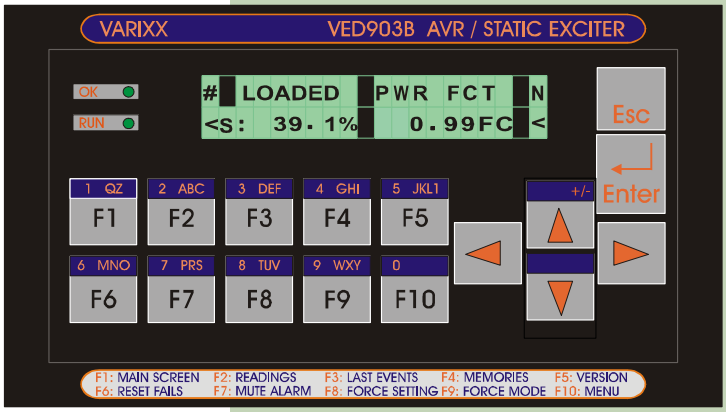
Bulletin 261DE

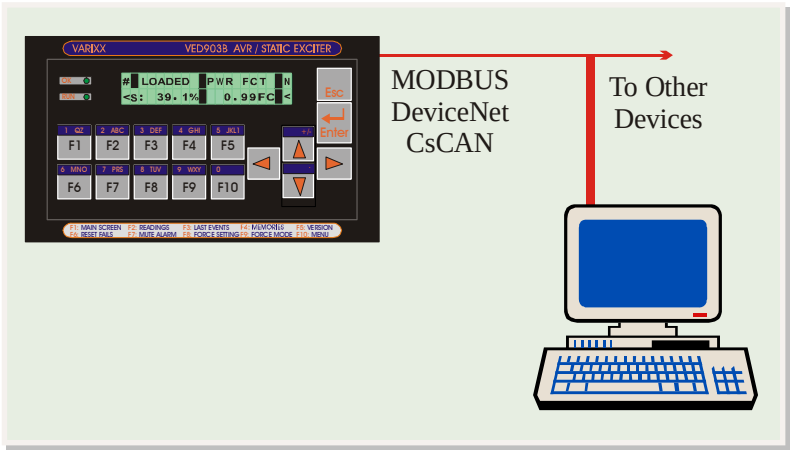
VED903B MODBUS CONTROL



Junho 2007 - Bios 2.26N

BLANK

VED903B Static Exciter Series



The **MODBUS** protocol describes an industrial communications and distributed control system developed by Gould-Modicon to integrate PLC's, computers, terminals, and other monitoring, sensing, and control devices. MODBUS is a Master/Slave communications protocol, whereby one device, (the Master), controls all serial activity by selectively polling one or more slave devices. The protocol provides for one master device and up to 247 slave devices on a common line. Each device is assigned an address to distinguish it from all other connected devices.

Only the master initiates a transaction. Transactions are either a query/response type, (only a single slave is address), or a broadcast/no response type, (all slaves are addressed). A transaction comprises a single query and single response frame or a single broadcast frame.

Certain characteristics of the MODBUS protocol are fixed, such as the frame format, frame sequences, handling of communications errors and exception conditions, and the functions performed.

Other characteristics are user selectable. These include a choice of transmission media, baud rate, character parity, number of stop bits, and the transmission modes, (ASCII or RTU). The user selected parameters are set, (hardwired or programmed), at each station. These parameters cannot be changed while the system is running.

Modes of Transmission

The mode of transmission is the structure of the individual units of information within a message, and the numbering system used to transmit the data. Two modes of transmission are available for use in a MODBUS system. Both modes provide the same capabilities for communicating with slaves; the mode is selected depending on the equipment used as a MODBUS Master. One mode must be used per MODBUS system; mixing of modes is not allowed. The modes are ASCII (American Standard Code for Information Interchange), and RTU, (Remote Terminal Unit.) The characteristics of the two transmission modes are defined below:

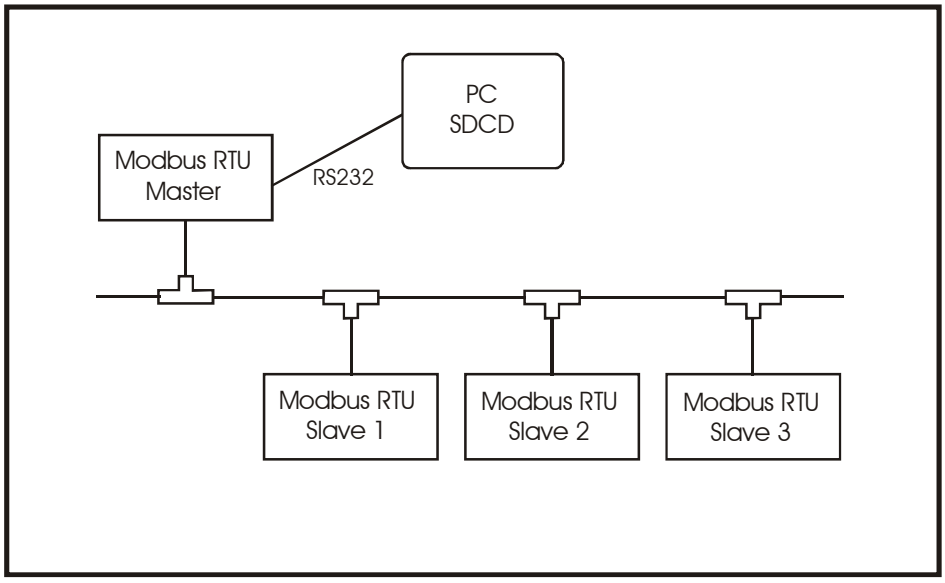
Characteristic	**ASCII (7-bit)**	**RTU (8-bit)**
Coding System	hexadecimal (uses ASCIIprintable characters (0-9, A-F)	8-bit binary
Number of bits per character:		
Start bits	1	1
Data bits (least significant first)	7	8
Parity (optional)	1	1
	(1-bit sent for even or odd parity, no bits for no parity)	(1-bit sent for even or odd parity, no bits for no parity)
Stop bits	1 or 2	1 or 2
Error Checking	LRC (Longitudinal Redundancy Check)	CRC (Cyclical Redundancy Check)

ASCII printable characters are easy to view when troubleshooting and this mode is suited to computer masters programmed in a high level language, such as FORTRAN, as well as PLC masters. RTU is suited to computer masters programmed in a machine language, as well as PLC masters.

In the RTU mode, data is sent in 8-bit binary characters. In the ASCII mode, each RTU character is first divided into two 4-bit parts, (high order and low order), and then represented by the hexadecimal equivalent. The ASCII characters representing the hexadecimal characters are used to construct the message. The ASCII mode uses twice as many characters as the RTU mode, but decoding handling the ASCII data is easier. Additionally, in the RTU mode, message characters must be transmitted in a continuous stream. In the ASCII mode, breaks of up to one second can occur between characters to allow for a relatively slower master.

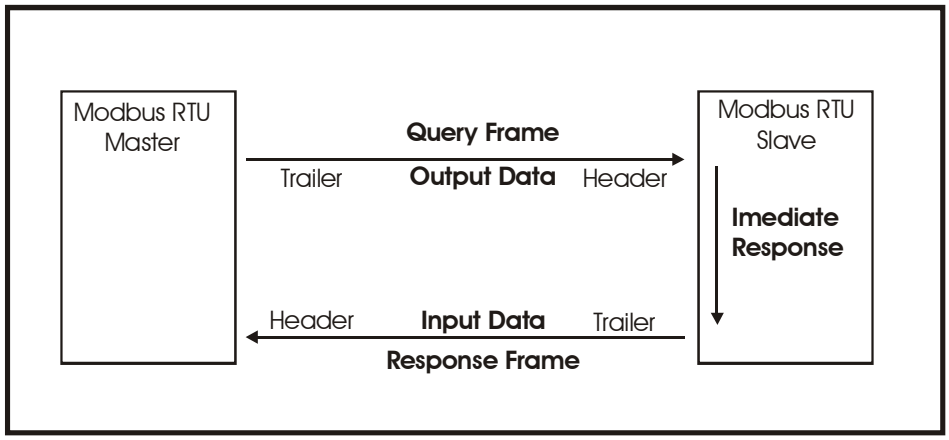
Connection Topology

Devices communicate using a master-slave technique, in which only one device (the master) can initiate transactions (called 'queries'). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels. Typical slaves include programmable controllers, motor controllers, load monitors etc, see Fig.



Data Topology

The master can address individual slaves. Slaves return a message (called a 'response') to queries that are addressed to them individually.
The Modbus protocol establishes the format for the master's query by placing into it the device address, a function code defining the requested action, any data to be sent, and an error
checking field. The slave's response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned and an error-checking field. If an error occurred in receiving the message, or if the slave is unable to perform the requested action, the slave will construct an error message and send this as its response, see Fig.



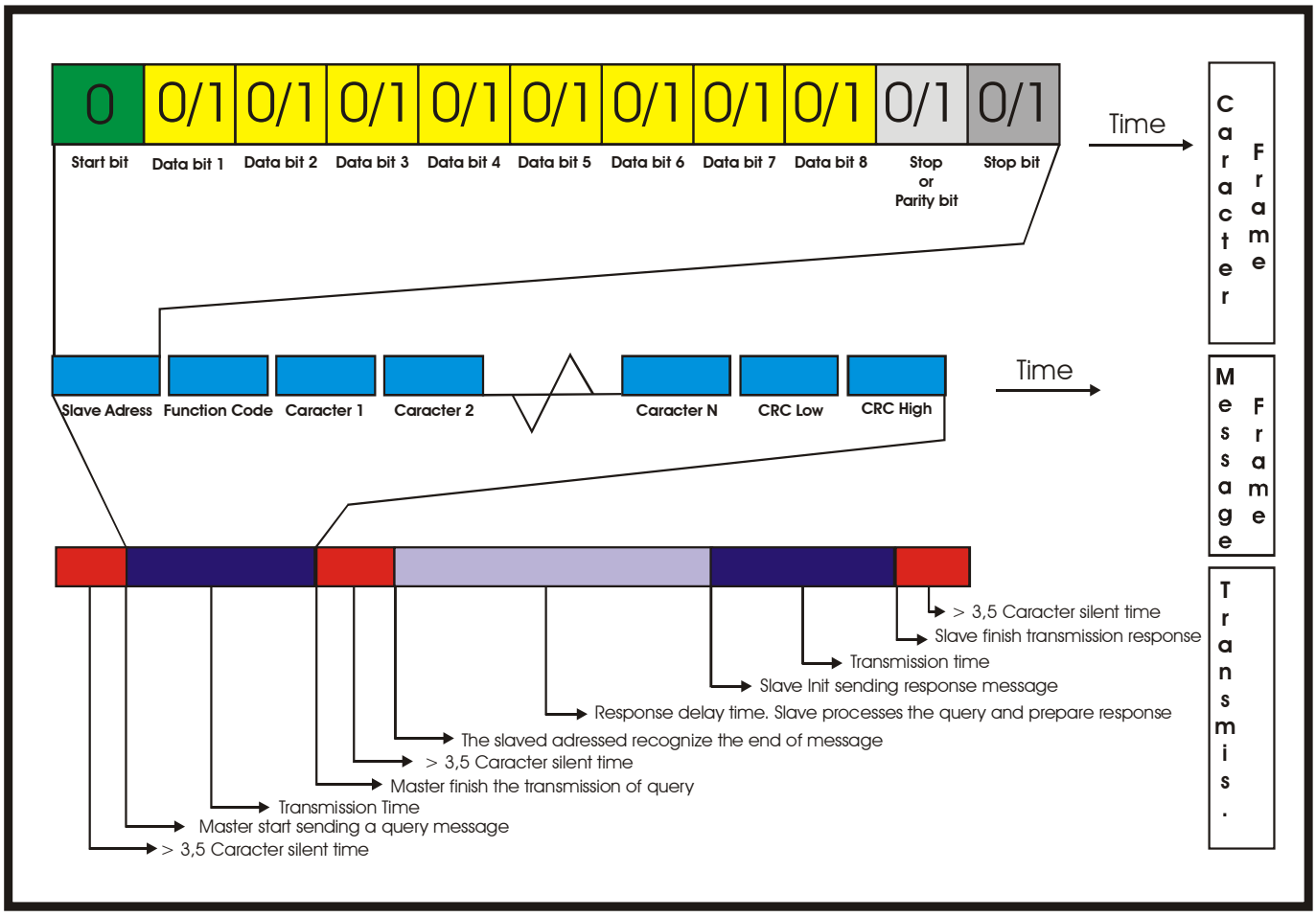
Timing Diagram

Modbus RTU uses a binary transmission protocol. If even parity is used, each character (8 bit data) is sent as:
If no parity is used each character (8 bit data) is sent as:

Table 1 Character frame with no parity.
1 Start bit.
8 Data bits, hexadecimal 0-9,A-F, least significant bit sent first.
1 Even parity bit.
1 Stop bit.

Table 2 Character frame with parity.
1 Start bit.
8 Data bits, hexadecimal 0-9,A-F, least significant bit sent first.
2 Stop bit.

Timing diagram for a transaction (query and response messages) (bottom in figure), a message frame (middle in figure) and a character frame (top in figure).



CRC Generation

The CRC is started by first pre-loading a 16-bit register to all 1's. Then a process begins of applying successive eight-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 eight-bit character is exclusive ORed with the register contents. 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 OR-ed 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 eight-bit character is exclusive OR-ed 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.

Generation in steps:

• **Step 1** Load a 16-bit register with 0xFFFF (all 1's). Call this the CRC register.

• **Step 2** Exclusive OR the first eight-bit byte of the message with the low order byte of the 16-bit CRC register, putting the result in the CRC register.

• **Step 3** Shift the CRC register one bit to the right (toward the LSB), zero-filling the MSB. Extract and examine the LSB.

• **Step 4** If the LSB is 0, repeat Step 3 (another shift). If the LSB is 1, Exclusive OR the CRC register with the polynomial value 0xA001 (1010 0000 0000 0001) .

• **Step 5** Repeat Steps 3 and 4 until eight shifts have been performed. When this is done, a complete eight-bit byte will have been processed.

• **Step 6** Repeat Steps 2 ... 5 for the next eight-bit byte of the message. Continue doing this until all bytes have been processed. Result The final contents of the CRC register is the CRC value.

• **Step 7** When the CRC is placed into the message, its upper and lower bytes must be swapped as described below.

• Placing the CRC into the Message

When the 16-bit CRC (two eight-bit bytes) is transmitted in the message, the low order byte will be transmitted first, followed by the high order byte - e.g., if the CRC value is 0x1241.

Message

CRC LO	41
CRC HI	12

Example of CRC Generation

An example of a C language function performing CRC generation is shown on this page.
The function takes two arguments:
• Unsigned char *puchMsg; A pointer to the message buffer containing binary data to be used for generating the CRC.
• Unsigned int usDataLen; The quantity of bytes in the message buffer. The function returns the CRC as a type unsigned int.
• Unsigned int CRC16 (unsigned int usDataLen, unsigned char *puchMsg)

CRC example.

```
#define CRC_POLYNOMIAL 0xA001
 unsigned int crc_reg;
 unsigned char i,k;
 crc_reg = 0xFFFF;
 for (i=0 ; i<usDataLen ; i++)
  {
    crc_reg ^= *puchMsg++;
    for (k=0 ; k<8 ; k++)
     {
      if (crc_reg & 0x0001)
         {
           crc_reg >>= 1;
           crc_reg ^= CRC_POLYNOMIAL;
         }
         else
             crc_reg >>= 1;
     }
  }
 return crc_reg;
```

Data Frames

The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 3.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.
Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will set
an error, as the value in the final CRC field will not be valid for the combined messages.
A typical message frame is shown below.
Header
START T1-T2-T3-T4
ADDRESS 8 bits
FUNCTION 8 bits
Data DATA n x 8 bits
Trailer
CRC CHECK 16 bits
END T1-T2-T3-T4

Address field

Address field
The address field of a message frame contains eight bits. The individual slave devices are assigned addresses in the range of 1 - 247. A master addresses 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 this address field of the response to let the master know which slave is responding.

Function field

Function field
The function code field of a message frame contains eight bits. Valid codes are in the range of 1 - 6, 15, 16 and 23. When a message is sent from a master to a slave device, the function code field tells the slave what kind of action to perform.
Examples are:
• to read the ON/OFF states of a group of inputs;
• to read the data contents of a group of parameters;
• to read the diagnostic status of the slave;
•to write to designated coils or registers within the slave.

When the slave responds to the master, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most significant bit set to a logic 1.
In addition to its modification of the function code for an exception response, the slave places an unique code into the data field of the response message. This tells the master what kind of error occurred, or the reason for the exception.
The master device's application program has the responsibility of handling exception responses. Typical processes are to post subsequent retries of the message, to try diagnostic messages to the slave and to notify operators. Additional information about function codes and exceptions comes later.

Data field

Data field

The data field is constructed using sets of two hexadecimal digits (8 bits), in the range of 00 to FF hexadecimal.

The data field of messages sent from a master to slave devices contains additional information which the slave must use to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled and the count of actual data bytes in the field.

For example, if the master requests a slave to read a group of holding registers (function code 03), the data field specifies the starting register and how many registers are to be read. If the master writes to a group of registers in the slave (function code 10 hexadecimal), the data field specifies the starting register, how many registers to write, the count of data bytes to follow in the data field, and the data to be written into the registers.

If no error occurs, the data field of a response from a slave to a master contains the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken.

CRC Error Checking Field

CRC Error checking field

The error checking field contains a 16 bit value implemented as 2 bytes. The error check value is the result of a Cyclical Redundancy Check (CRC) calculation performed on the message contents.

The CRC field is appended to the message as the last field in the message. When this is done, the low-order byte of the field is appended first, followed by the high-order byte. The CRC high-order byte is the last byte to be sent in the message. Additional information about CRC calculation are found in this manual.

Standard Functions

Functions

Standard MODBUS function codes.

Function name	Function code
Read Coil (Bit) Status	1 (01h)
Read Input Status	2 (02h)
Read Holding Registers	3 (03h)
Read Input Registers	4 (04h)
Force Single Coil (Bit)	5 (05h)
Force Single Register	6 (06h)
Force Multiple Coils (Bits)	15 (0Fh)
Force Multiple Registers	16 (10h)
Force/Read Multiple Holding Registers	23 (17h)

Reading Coil

Read Coil Status

Read the status of digital changeable parameters.
EXAMPLE
Requesting the coil (Bit) 29 input state. Suppose it is On
30 input: Modbus no = 29 (1Dh)
On = Yes = 1 Coil = 0001
1 byte of data: Byte count=01

Request message.

Field name	Hex value
Slave address	01
Function	01
Start address HI	00
Start address LO	1D
Number of Coils HI	00
Number of Coils LO	01
CRC LO	6D
CRC HI	CC

Response message.

Field name	Hex value
Slave address	01
Function	01
Byte count	01
Coil no.29 (1Dh) status	01
CRC LO	90
CRC HI	48

Reading Input

Read Input Status

Read the status of digital read-only information.
EXAMPLE : Request the digital input 2. Assuming that It is no active.
Status: Modbus no = 2.

Request message.

Field name	Hex value
Slave address	01
Function	02
Start address HI	00
Start address LO	02
Number of Inputs HI	00
Number of Inputs LO	01
CRC LO	18
CRC HI	0A

Response message.

Field name	Hex value
Slave address	01
Function	02
Byte count	01
Input no.2 (02h)status	00
CRC LO	A1
CRC HI	88

Reading Holding Registers

Read Holding Registers

Read the value of analogue changeable information.
Example, requesting some Voltage, Frequency and Current. Their values are 400.0 V, 60 Hz and 15.5 A.
400.0V, unit 0.1V - 4000 (0FA0h)
60Hz unit 1Hz - 60 (003Ch)
15.5A, unit 0.1A - 155 (009Bh)

Request message.

Field name	Hex value
Slave address	01
Function	03
Start address HI	00
Start address LO	00
Number of Registers HI	00
Number of Registers LO	03
CRC LO	05
CRC HI	CB

Response message

Field name	Hex value
Slave address	01
Function	03
Byte count	06
Reg no. 0, (0h) data HI	0F
Reg no. 0, (0h) data LO	A0
Reg no. 1, (1h) data HI	00
Reg no. 1, (1h) data LO	3C
Reg no. 2, (2h) data HI	00
Reg no. 2, (2h) data LO	9B
CRC LO	20
CRC HI	34

Reading Input Registers

Read Input Registers

Read the contents of analogue read-only information.
EXAMPLE
Request the value of the 30011 Modbus - Nº 10. Supose It is 452.0.
It has a long representation. 2 registers are used (30011 high word and 30012 low word) 452.0, unit 0.1 - 4520 (000011A8h).

Request message.

Field name	Hex value
Slave address	01
Function	04
Start address HI	00
Start address LO	0A
Number of Registers HI	00
Number of Registers LO	02
CRC LO	51
CRC HI	C9

Response message.

Field name	Hex value
Slave address	01
Function	04
Byte count	04
Reg no. 10 (0Ah) data HI	00
Reg no. 10 (0Ah) data LO	00
Reg no. 11 (0Bh) data HI	11
Reg no. 11 (0Bh) data LO	A8
CRC LO	F6
CRC HI	6A

Forcing Single Coil (Bit)

Force Single Coil (Bit)

Set the status of one changeable digital parameter.
EXAMPLE
Set one command to ON. This will cause the some kind of action.
Modbus no = 1 - address LO 1 (01h)
Run = 1 - 0 Data HI = 255 (0FFh), Data LO = 00 (00h)

Request message.

Field name	Hex value
Slave address	01
Function	05
Start address HI	00
Start address LO	01
Data HI	FF
Data LO	00
CRC LO	DD
CRC HI	FA

Response message.

Field name	Hex value
Slave address	01
Function	05
Start address HI	00
Start address LO	01
Data HI	FF
Data LO	00
CRC LO	DD
CRC HI	FA

Forcing Single Register

Force Single Register

Set the value of one analogue changeable parameter.
EXAMPLE
Set the register 40014 (Modbus 13) to 12.5 sec. = (125 /10)
Modbus no 13 -> address LO (0Dh) 12.5s, unit 0.1s - 125 (7Dh)

Request message.

Field name	Hex value
Slave address	01
Function	06
Start address HI	00
Start address LO	0D
Data HI	00
Data LO	7D
CRC LO	D8
CRC HI	28

Response message.

Field name	Hex value
Slave address	01
Function	06
Start address HI	00
Start address LO	0D
Data HI	00
Data LO	7D
CRC LO	D8
CRC HI	28

Forcing Multiple Coil (Bits)

Force Multiple Coil

Set the status of multiple digital changeable parameters.
EXAMPLE
Set one flag to ON and other to ON. This will cause some actions or change parameters. Coil no. = 0-1 Reset -> 1 // Run = 1 ->- 00000011 (03h)

Request message.

Field name	Hex value
Slave address	01
Function	0F
Start address HI	00
Start address LO	00
Number of Coils HI	00
Number of Coils LO	02
Byte count	01
Coil no. 0-1 status (0000 0011B)	03
CRC LO	9E
CRC HI	96

Response message.

Field name	Hex value
Slave address	01
Function	0F
Start address HI	00
Start address LO	00
Number of Coils HI	00
Number of Coils LO	02
CRC LO	D4
CRC HI	0A

Forcing Multiple Register

Force Multiple Register

Set the contents of multiple changeable analogue parameters.
EXAMPLE
Set the register 40018 (Modbus Nº 17) to 25.0 (250 / 10)
and 40019 (Modbus Nº 18) to 55.
25.0, unit 0.1 -> - 250 (00FAh) // 55, unit 1% -> 55 (0037h)

Request message.

Field name	**Hex value**
Slave address	01
Function	10
Start address HI	00
Start address LO	11
Number of Registers HI	00
Number of Registers LO	02
Byte count	04
Data HI reg 17 (11h)	00
Data LO reg 17 (11h)	FA
Data HI reg 18 (12h)	00
Data LO reg 18 (12h)	37
CRC LO	52
CRC HI	88

Response message.

Field name	**Hex value**
Slave address	01
Function	10
Start address HI	00
Start address LO	11
Number of Registers HI	00
Number of Registers LO	02
CRC LO	11
CRC HI	CD

Forcing / Reading Multiple Register

Force/Read Multiple Register

Set and read the contents of multiple analogue changeable parameters in the same message.

EXAMPLE

Set one parameter to 2 (40022 = Modbus Nº 21) and other to 1 (40023 = Modus Nº 22) and read others two. They are 1450 and 17000.

1450, unit 1 -> 1450 (05AAh)

17000, unit 1-> 17000 (4268h)

Request message.

Field name		**Hex value**
Slave address		01
Function		17
Start read address HI		00
Start read address LO		03
Number of read Regs HI	00	
Number of read Regs LO	02	
Start write address HI		00
Start write address LO		15
Number of write Regs HI	00	
Number of write Regs LO		02
Byte count		04
Data HI Reg 21 (15h)		00
Data LO Reg 21 (15h)		02
Data HI Reg 22 (16h)		00
Data LO Reg 22 (16h)		01
CRC LO		62
CRC HI		77

Response message.

Field name	**Hex value**
Slave address	01
Function	17
Byte count	04
Reg no. 3, (3h) data HI	05
Reg no. 3, (3h) data LO	AA
Reg no. 4, (4h) data HI	42
Reg no. 4, (4h) data LO	68
CRC LO	E8
CRC HI	85

Kinds of Error

Errors, exception codes

Two kinds of errors are possible:
• Transmission errors.
• Operation errors.

Transmission errors

Transmission errors are:
• Frame error (stop bit error).
• Parity error (if parity is used).
• CRC error.
• No message at all.

These errors are caused by i.e. electrical interference from machinery or damage to the communication channel (cables, contact, I/O ports etc.). This unit will not act on or answer the master when a transmission error occurs. (Same result as if a non-existing slave is addressed). The master will eventually cause a time-out condition.

Operation errors

If no transmission error is detected in the master query, the message is examined. If an illegal function code, data address or data value is detected, the message is not acted upon but an answer with an exception code is sent back to the master. This unit can also send back an exception code when a set (force) function message is received during some busy operation states.
Bit 8 (most significant bit) in the function code byte is set to a '1' in the exception response message. Example with an illegal data address when reading an input register.

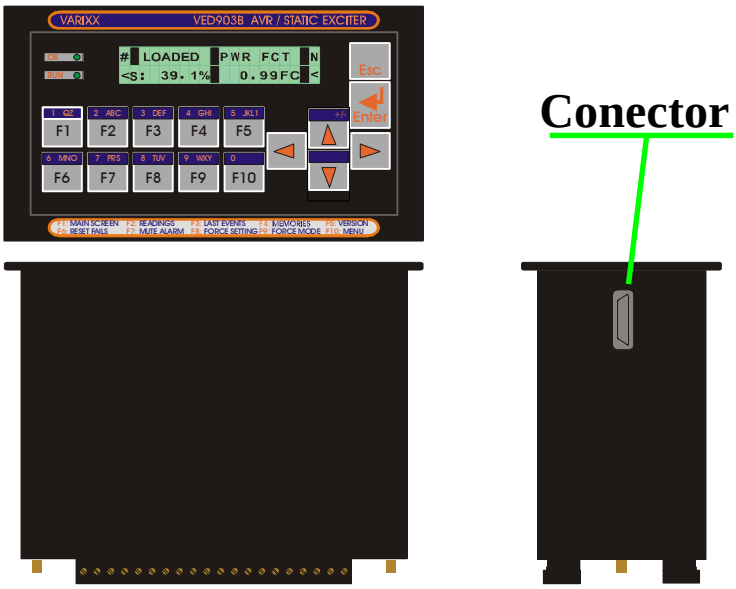
Exception response message.

Field name	Hex value
Slave address	01
Function	84
Exception code	02
CRC LO	C2
CRC HI	C1

Exception codes.

Exc. code	Name	Description
01	Illegal function	This unit doesn't support the function code.
02	Illegal data address	The data address is not within its boundaries.
03	Illegal data value	The data value is not within it's boundaries.
06	Busy	The unit is unable to perform the request at this time. Retry later.

RS232 Conector Localization



PORTA DE COMUNICAÇÃO RS232C:

STANDARD 9 PINOS (DB9). OPTIONALLY VARIXX CAN SUPPLY A RS232C TO RS485 CONVERTER MODULE.

Serial Port - This connector is FEMALE and has the following pin assignments:

PinNumber	Function	DirectionRelative to VED903	Description
1	DCD	Out	Data Carrier Detect(always ON)
2	RXD	Out	Receive Data
3	TXD	In	Transmit Data
4	DTR	In	Data Terminal Ready
5	GND	–	Ground return
6	DSR	Out	Data Set ready
7	RTS	In	Request To Send
8	CTS	Out	Clear To Send
9	RI	Out	Ring Indicator (Always OFF)

RS232 to RS485 Converter.

Case of use a RS232 / RS485 Converter:
A RS485 is adequate to a multi-droop network.

In the multi-droop Modbus network is necessary a termination with resistors in the last slave unit of the network cable at two or 4 wires.

These resistors increases the noise immunity and its non placement commits the reliability. The connection cables of the network should be of good quality and shielded with the shield must be connected to the ground in just one point. In networks using two wires cable in short distances it is acceptable use of twisted pair's use.

There are two board versions: one of them with DIP switchs to close the termination resistors and the other without DIP switchs or resistors. In the version without the termination resistors is the user's responsibility the placement of the resistors directly in the connector cable of the last slave device of the network.

The value of these resistors must have a adequate ohm value and it should be connected among the lines A / Ground and B / Ground.

ASCII Framing

Framing in ASCII Transmission mode is accomplished by the use of the unique colon, (:), character to indicate the beginning of frame and carriage return/line feed, (CRLF), to delineate end of frame. The line feed character also serves as a synchronizing character which indicates that the transmitting station is ready to receive an immediate reply.

BEGIN FRAME	ADDRESS	FUNCTION	DATA	ERROR CHECK	EOF	READY TO RECEIVE
:	2-CHAR 16-BIT	2-CHAR 16-BIT	N X 4-CHAR N X 16-BITS	2-CHAR 16-BITS	CR	LF

RTU Framing

Frame synchronization can be maintained in RTU transmission mode only by simulating a synchronous message. The receiving device monitors the elapsed time between receipt of characters. If three and one-half character times elapse without a new character or completion of the frame, then the device flushes the frame and assumes that the next byte received will be an address.

T1,T2,T3	ADDRESS	FUNCTION	DATA	CHECK	T1,T2,T3
	8-BITS	8-BITS	N X 8-BITS	16-BITS	

Address Field

The address field immediately follows the beginning of frame and consists of 8-bits, (RTU), or 2 characters, (ASCII). These bits indicate the user assigned address of the slave device that is to receive the message sent by the attached master.
Each slave must be assigned a unique address and only the addressed slave will respond to a query that contains its address. When the slave sends a response, the slave address informs the master which slave is communicating. In a broadcast message, an address of 0 is used. All slaves interpret this as an instruction to read and take action on the message, but not to issue a response message.

Function Field

The Function Code field tells the addressed slave what function to perform. MODBUS function codes are specifically designed for interacting on MODBUS industrial communications system. The high order bit in this field is set by the slave device to indicate an exception condition in the response message. If no exceptions exist, the high-order bit is maintained as zero in the response message.

Data Field

The data field contains information needed by the slave to perform the specific function or it contains data collected by the slave in response to a query. This information may be values, address references, or limits. For example, the function code tells the slave to read a holding register, and the data field is needed to indicate which register to start at and how many to read. The imbedded address and data information varies with the type and capacity of the equipmment associated with the slave.

Error Check Field

This field allows the master and slave devices to check a message for errors in transmission. Sometimes, because of electrical noise or other interference, a message may be changed slightly while its on its way from one device to another. The error checking assures hat the slave or master does not react to messages that have changed during transmission. This increases the safety and the efficiency of the MODBUS system.
The error check field uses a Longitudinal Redundancy Check, (LRC), in the ASCII mode of transmission, and a CRC-16 check in the RTU mode.

Exception Responses

Programming or operation errors are those involving illegal data in a message, no response from the slave to its interface unit, or difficulty in communicating with a slave. These errors result in an exception response from either the master computer software or the slave, depending on the type of error. The exception response codes are listed below. When a slave detects one of these errors, it sends a response message to the master consisting of the slave address, function code, error code, and error check fields. To indicate that the response is a notification of an error, the high-order bit of the function code is set to one.

CODE	NAME	MEANING
01	ILLEGAL FUNCTION	The message function received is not an allowable action for the addressed slave.
02	ILLEGAL DATA ADDRESS	The address referenced in the data field is not an allowable address for the addressed slave device.
03	ILLEGAL DATA VALUE	The value referenced in the data field is not allowable in the addressed slave location.
04	FAILURE IN ASSOCIATED DEVICE	The slave's PC has failed to respond to a message or an abortive error occurred.
05	ACKNOWLEDGE	The slave PLC has accepted and is processing the long duration program command.
06	BUSY, REJECTED MESSAGE	The message was received without error, but the PLC is engaged in processing a long duration program command.
07	NAK-NEGATIVE ACKNOWLEDGMENT	The PROGRAM function just requested could not be performed.

Using Modbus Slave Communications

Overview:
The VED903B allow the serial port to act as a Modbus/RTU slave. The Modbus function supports both ASCII and RTU modes (RTU configured by factory, ASCII under request) of operation across a range of baud rates and protocol frames. Also supported is port activity status, an in-activity timer and support for call-on-exception operation.
Basic Operation:
Before the Modbus function accepts messages, the Modbus must be activated pressing F5 (keyboard) for more than 3 seconds (Toogle).
Inactivity Timer:
The Modbus function contains a timer that is reset on the reception of a valid message addressed to this function. Should communications cease between the master and this function, that timer expires which sets an Inactivity timeout bit in the status word. Once communications is reestablished, both the timer and the Inactivity timeout bit in the status word are reset. Setting the timeout value to zero (at menu 13) disables this feature.
Report-on-exception:
Report-on-exception is a method of immediately informing the master that the slave has important information pending. This method is typically used in applications where modems are used as the communication channel, and the slaves are polled for data between long intervals. Once the connection is established, the master and slave require some cooperative functionality on determining the address of the slave calling. Since this functionality is not a standardized or a part of the Modbus protocol, the Modbus function contains two alternate methods such that the one most appropriate for the master is selected.
The first method involves the slave responding to the non-standard Modbus request **Get Slave Address**, which is broadcasted by the master after the connection is established. Since this is just a response to a Modbus request, this method does not require that Exception Messaging be enabled. **This is method used by VED903**. Use of this method with a third party master can require that master to be modified to support this command. The Modbus request and response frames are presented below:

Request:

ADDR	FUNC	
0	65 (41H)	
Response:		
ADDR	FUNC	DATA
(SLAVE ADDR)	65(41H)	(SLAVE ADDR)

The second method involves the slave sending an unsolicited response (Exception Message) to the master once the connection is established (available only by request). The specific byte pattern used for the Exception Message depends on that supported by the master. When sent, the appropriate header and checksums are inserted automatically by the Modbus function. The Byte Count acts as the trigger that starts the transmission of the response. When the Byte Count transitions from zero to a specific number, that number of bytes are sent. Once transmitted, the Modbus function responds to master requests as expected.

Master Mapping:
To access a memory point or memory flag over Modbus, the master must be configured as to the point's type and offset. This is usually accomplished with one of two methods. The **first method** uses the traditional addressing scheme where the **high digit** represents the **point's type** and the l**ower digits** represent the **point's offset** (starting with point 1). Since only four types can be represented in this manner, the Modbus function packs several data tables into a single point type array.

The **Traditional RTU Reference** column below specifies the **starting address of each table**. The s**econd method** requires the master to be configured with the specific Modbus command and offset. The supported Modbus commands and the associated offset are also illustrated below.

Reference	Maximum Range	Traditional Modbus Reference	Modbus Command(s)	ModbusOffset
%I1	2048	10001	Read Input Status (2)	00000
%IG1	256	13001		03000
%S1	256	14001		04000
%K1	256	15001		05000
%Q1	2048	00001	Read Flag Status (1)Force Flag (5)Force Multiple Flags (15)	00000
%M1	2048	03001		03000
%T1	2048	06001		06000
%QG1	256	09001		90000
%AI1	512	30001	Read Input Register (4)	00000
%AIG1	32	33001		03000
%SR1	32	34001		04000
%AQ1	512	40001	Read Holding Register (3)Load Register (6)Load Multiple Registers (16)	00000
%R1	2048	43001		03000
%AQG1	32	46001		06000

Parameter Alteration in VED903

The alteration of parameters, depending on the case, can be immediately effective after the alteration in the VED903 menu or it can only be effective after the exit of the menu if the alteration goes by the keyboard, or setting the flag %R11.11 (On) if the alteration goes by Modbus. This is made for reasons of synchronizing or gain of processing speed.

For each parameter the alteration will be indicated **"Immediate"** if there is not necessity of setting %R11.11 to be efective or **"Flagged"**. If there is necessity of setting %R11.11 to be efective. Change all parameters you want first and than set %R11.11 to make the changes efective. The flag %R11.11 after being seted at "1" is automatically reseted to "0" by the "firmware" after some mili-seconds.

Comm Port Buffering

The VED903 firmware maintains a Transmit Buffer and a Receive Buffer. When a Send or Receive task is performed, data is transferred between the appropriate buffer and the program's registers.

For a Comm Port Transmit element, the TX Count word contains the number of characters moved from the program registers to the transmit buffer. This number can be less than the requested number if the comm port buffer is full.

For a Comm Port Receive element, the RX Count word contains the number of characters moved from the receive buffer in the program area. This number can be less than the requested number if the comm port buffer contains fewer characters than requested.

Serial Port

The serial port physically present on the VED903 unit is referred to as COMM1.

Handshaking

Handshaking is a method whereby the destination end of a transmission can control how much and when data is sent to it.

NOTE: For purposes of this discussion, source end is defined as the unit which is transmitting data. Destination end is defined as the unit which actually receives the data.

Handshaking is configured on VED903 menu. There are five (5) possible types of handshaking:

NONE -- There is no handshaking. The source unit sends as many data bytes as it can as fast as possible for a given the baud rate. No consideration is given to the capabilities of the destination end.

XON/XOFF -- (Also called software handshaking) The destination end keeps track of how many characters it has received and the size of its internal buffers. If the buffer gets full or the unit is otherwise unable to receive further characters, it must transmit the XOFF (transmit off) character. The source end must then stop transmitting data until a subsequent XON character is sent by the destination end.

Because there is some heavy software overhead involved, the timing of transmissions is variable. The destination must first determine that it is full and then transmit the XOFF signal. The source end must read the XOFF signal and react to it. In the mean time, several additional data bytes can be sent. It is up to the destination end to ensure that it sends the XOFF signal soon enough that the buffer is not overrun.

The XON and XOFF characters are predefined by the ASCII character set. XON is 11 hex or 17 decimal. XOFF is 13 hex or 19 decimal. The XON/XOFF handshaking is most often used where only ASCII values are being sent. XON/XOFF can not be easily used where binary data is involved, because the XON/XOFF codes are also valid binary codes.

Note that XON/XOFF handshaking usually implies a full duplex (both ends may transmit simultaneously) communications channel as the destination end needs to transmit the XOFF characters at any time (including in the middle of a transmission from the source end).

The advantage to XON/XOFF handshaking is that it can be implemented using an easy and cheap three-wire (TX/RX/Common) cable.

HARDWARE -- Also called RTS/CTS handshaking. Hardware handshaking requires extra signals be sent between the two units, thus this is more expensive to implement due to the increased numbers of wires in the interconnecting cables.

In operation, the destination end determines that it is empty, and activates its CTS (Clear To Send) signal. In response, the source end sends data so long as the CTS signal remains active.

Many devices have the RTS/CTS signals wired directly into the hardware. Thus, an inactive CTS signals from the destination end can instantly shut down the source end. These hardware operations can be very fast because no software control is necessary in this case. Also, this manner of handshaking can be used regardless of the nature of the data being transmitted, ASCII coded or binary.

Multi-Drop Full Duplex -- In a full-duplex multi-drop situation, all available units are wired in parallel. For receiver circuitry this is no problem so long as the load on the network is not excessive. All units have their receivers enabled at all times.

Each message sent through the system is somehow identified by giving it a drop address. All units will receive all messages. All

units check the drop address against their own address, and only the unit with the matching address responds.
When a unit determines that it has something to transmit it turns on its transmitter, sends the necessary data packet, and then disables its transmitter.
Full Duplex Multi-drop is usually found in multi-master or peer to peer systems, where all units have a more or less equal chance of needing to transmit a message. Often, the units needs to verify that the message they sent are sent correctly so the receiver is left on at all times.
The advantage to this system is that many units can be connected to a simple three-wire (RX/RX/Common) cable. The drawback to this system is increased firmware and software complexity.
Multi-Drop Half Duplex -- Multi-Drop Half-Duplex operation is identical to Full-Duplex except that the transmitting unit's receiver is disabled when the unit is transmitting.
All units maintain their transmitters disabled and receivers enabled at all times except when they need to transmit. Usually, protocols dictate that only the unit matching the drop address can transmit. This unit turns on its transmitter, turns off its receiver, sends the necessary data packet, and then disables its transmitter and enables its receiver.
Half Duplex Multi-drop is usually found in Master/Slave systems where one unit is designated a Master and all other units are Slaves. The Master transmits a message to one Slave, and then disable it's transmitter. All Slaves hear the message, but only the Slave with the matching "drop address" will turn on its transmitter and respond.

Using RS-485 with the VED903
The VED903 does not provide RS-485 compatible signals. It is necessary to purchase and install a third-party RS-232 to RS-485 converter.
In this mode, transmitter control is VED903 Signal CTS, available on the DB-9 connector, Pin 8. When the VED903 asserts this signal, the converter enables its transmitting section.

How to Connect a MODBUS slave device.
The physical device characteristics of the particular slave device determines the communications parameters required for connection. MODBUS is a multidrop communications protocol, (software), but typical RS-232 serial connections are not. RS-232 is basically a point-to-point hardware protocol with the transmit line of one device connected to the receive line of another device. Various combinations of protocol converters and/or modems may be used to multidrop RS-232 data links. Additionally, some serial cards may be configured to support 20 mA current loop for multidrop operation.

If a single slave device is to be connected, standard RS-232 hardware may be used. Depending upon the requirements of the master device, certain control signals may be required. These are typically RTS/CTS, (pins 4&5), or DTR/DSR/DCD, (pins 6,8 & 20). The VED903 supports these control signals.

Error Detection.

There are two types of errors which may occur in a communications system: transmission errors and programming errors. The MODBUS system has specific methods for dealing with either type of error.
Communications errors usually consist of a changed bit or bits within a message. The most frequent cause of communications errors is noise: unwanted electrical signals in a communications channel. These signals occur because of electrical interference from machinery, damage to the communications channel, impulse noise, (spikes), etc. Communications errors are detected by character framing, a parity check, and a redundancy check.
When the character framing, parity, or redundancy checks detect a communications error, processing of the message stops. A slave will not act on or respond to the message. (The same occurs if a non-existent slave address is used.)
When a communications error occurs, the message is unreliable. The slave cannot know for sure if this message was intended for it. So the CPU might be answering a message which was not its message to begin with. It is essential to program the MODBUS Master to assume a communications error has occurred if there is no response in a reasonable time. The length of this time depends upon the baud rate, type of message, and scan time of the slave. Once this time is determined, the master may be programmed to automatically retransmit the message.
The MODBUS system provides several levels of error checking to assure the quality of the data transmission. To detect multibit errors where the parity has not changed, the system uses redundancy checks: Cyclical Redundancy Check, (CRC), for the RTU mode and Longitudinal Redundancy Check, (LRC), for the ASCII mode.
CRC-16 Cyclic Redundancy Check
The CRC-16 error check sequence is implemented as described in the following paragraphs.
The message, (data bits only, disregarding start/stop and parity bits), is considered as one continuous binary number whose most significant bit, (MSB), is transmitted first. The message is pre-multiplied by X**16, (shifted left 16 bits), then divided by X**16 + X**15 + X**2 + 1 expressed as a binary number (11000000000000101). The integer quotient digits are ignored and the 16-bit remainder (initialized to all ones at the start to avoid the case where all zeroes being an accepted message), is appended to the message, (MSB first), as the two CRC check bytes. The resulting message including the CRC, when divided by the same polynomial (X**16 + X**15 + X**2 + 1), at the receiver will give a zero remainder if no errors have occurred. (The receiving unit recalculates the CRC and compares it to the transmitted CRC). All arithmetic is performed modulo two, (no carries). An example of the CRC-16 error check for message HEX 0207, (address 2, function 7 or a status request to slave number 2) follows:
The device used to serialize the data for transmission will send the conventional LSB or right-most bit of each character first. In generating the CRC, the first bit transmitted is defined as the MSB of the dividend. For convenience then, and since there are no carries used in arithmetic, let's assume while computing the CRC that the MSB is on the right. To be consistent, the bit order of the generating polynomial must be reversed. The MSB of the polynomial is dropped since it affects only the quotient and not the remainder. This yields 1010 0000 0000 0001, (HEX A001). Note that this reversal of the bit order will have no effect whatever on the interpretation or the bit order of characters external to the CRC calculations.

The step by step procedure to form the CRC-16 is as follows:

1. Load a 16-bit register with all 1's.
2. Exclusive OR the first 8-bit byte with the high order byte of the 16-bit register, putting the result in the 16-bit register.
3. Shift the 16-bit register one bit to the right.
4a. If the bit shifted out to the right is one, exclusive OR the generating polynomial 1010 0000 0000 0001 with the 16-bit register.
4b. If the bit shifted out to the right is zero; return to step 3.
5. Repeat steps 3 and 4 until 8 shifts have been performed.
6. Exclusive OR the next 8-bit byte with the 16-bit register.
7. Repeat step 3 through 6 until all bytes of the message have been exclusive OR'rd with the 16-bit register and shifted 8 times.
8. The contents of the 16-bit register are the 2 byte CRC error check and is added to the message most significant bits first.

16-BIT REGISTER	MSB				Flag
(Exclusive OR)		1111	1111	1111	1111
02			0000	0010	
1111	1111	1111	1101		
Shift 1	0111	1111	1111	1110	1
Polynomial		1010	0000	0000	0001
1101	1111	1111	1111		
Shift 2	0110	1111	1111	1111	1
Polynomial		1010	0000	0000	0001
1100	1111	1111	1110		
Shift 3	0110	0111	1111	1111	0
Shift 4	0011	0011	1111	1111	1
Polynomial		1010	0000	0000	0001
1001	0011	1111	1110		
Shift 5	0100	1001	1111	1111	0
Shift 6	0010	0100	1111	1111	1
Polynomial		1010	0000	0000	0001
1000	0100	1111	1110		
Shift 7	0100	0010	0111	1111	0
Shift 8	0010	0001	0011	1111	1
Polynomial		1010	0000	0000	0001
1000	0001	0011	1110		
07			0000	0111	
1000	0001	0011	1001		
Shift 1	0100	0000	1001	1100	1
Polynomial		1010	0000	0000	0001
1110	0000	1001	1101		
Shift 2	0111	0000	0100	1110	1
Polynomial		1010	0000	0000	0001
1101	0000	0010	1111		
Shift 3	0110	1000	0010	0111	1
Polynommial		1010	0000	0000	0001
1100	1000	0010	0110		
Shift 4	0110	0100	0001	0011	0
Shift 5	0011	0010	0000	1001	1
Polynomial		1010	0000	0000	0001
1001	0010	0000	1000		
Shift 6	0100	1001	0000	0100	0
Shift 7	0010	0100	1000	0010	0
Shift 8	0001	0010	0100	0001	0
HEX 12			HEX 41		

TRANSMITTED MESSAGE WITH CRC-16
(MESSAGE SHIFTED TO RIGHT TO TRANSMIT)

12		41		07		02	
0001	0010	0100	0001	0000	0111	0000	0010

LRC (Longitudinal Redundancy Check)
The error check sequence for the ASCII mode is LRC. The error check is an 8-bit binary number represented and transmitted as two ASCII hexadecimal (hex) characters. The error check is produced by converting the hex characters to binary, adding the binary characters without wraparound carry, and two's complementing the result. At the received end the LRC is recalculated and compared to the sent LRC. The colon, CR, LF, and any imbedded non-ASCII hex characters are ignored in calculating the LRC.

Address 02		0000 0010
Function 01		0000 0001
Start Add H.O.	00	0000 0000
Start Add L.O.	00	0000 0000
Quantity of Pts	00	0000 0000
	08	0000 1000
Sum		0000 1011
1's complement		1111 0100
+1		0000 0001
Error Check F5 2's complement		1111 0101

Data Types

In VED903, data may be stored or used in a variety of different formats. The format used depends on how the information is to be interpreted. Typical interpretations are binary bit patterns, unsigned numbers, signed numbers, floating point values, and strings.

Type Name Description

BOOL Boolean A single bit. It can contain only the values '0' or '1'.
BYTE Byte A string of 8 consecutive bits. Byte values are used where the value of the data is not as important as the bit patterns (shifts and rotates).
WORD Word A string of 16 consecutive bits. Word values are used where the value of the data is not as important as the bit patterns (shifts and rotates).
DWORD Double Word A string of 32 consecutive bits. DWORD values are used where the value of the data is not as important as the bit patterns (shifts and rotates).
INT Integer A 16-bit signed value. Integers are used where the value of the data is expected to be in the range of -32,768 to +32,767
SINT Short Integer An 8-bit signed value. Short Integers are used where the value of the data is expected to be in the range of -128 to +127.
DINT Double Integer A 32-bit signed value. Double Integers are used where the value of the data is expected to be in the range of -2,147,483,648 to +2,147,483,647.
UINT Unsigned Integer A 16-bit unsigned value. Unsigned Integers are used where the value of the data is expected to be in the range of -0 (zero) to 65,535.
USINT Unsigned Short Integer An 8-bit unsigned value. Unsigned Short Integers are used where the value of the data is expected to be in the range of 0 (zero) to 255
UDINT Unsigned Double Integer A 32-bit unsigned value. Unsigned Double Integers are used where the value of the data is expected to be in the range of 0 (zero) to 4,294,967,296.
REAL Floating Point A 32-bit value. Values are stored and operated on in IEEE single precision (six digit) format. Values range from -3.40282E+38 to +3.40282E+38.
STRING String A variable-length succession of characters. Each character is represented by one byte.
The bits in word registers may be used as Boolean values. In this case, Bit Offset Addressing is used to specify the Register Type, Offset, and Bit Offset for the required bit.
Using Boolean registers to represent Real numbers is usually ineffective.

STORAGE ORDER
32-bit values (DWORD, DINT, UDINT) occupy 32 consecutive bits of data, or two (2 consecutive 16-bit registers. For example, if a DINT is defined at Register %R43, the 32-bit value is contained in %R43 and %R44.

For 32-bit values, data is stored Low Order Word first. For example, if a DINT is defined at Register %R43 and contains the value "65540", (0000000000000001 0000000000000100) register %R43 will contain "4" and %R44 will contain "1".

Byte values (such as STRINGS) are stored High Order Byte first. For example, to store the string "31" in register %R43, store the HEX value 3133 (decimal 12595).

Real Numbers
A number which contains an explicit decimal point is known as a REAL or Floating Point number. The numbers are termed "real" because they reflect the real value of a measurement (to the accuracy of the system) in whole units and fractional parts of units without artificial truncation to some less-precise format such as integers.
The location of the decimal point (thus determining the number of whole units and fractional parts) is contained with the number itself. Since for any given real number the decimal point can be in a different position, real number are often called floating point. In VED903, the terms real and floating point are used interchangeably.

FORMAT
Real numbers are usually input and displayed as a six digit field:

3.12159 654321

If the number is too large or too small to be represented using only six digits, the number is displayed as a six-digit field plus an exponent:
1.03647e+12 9.73157e-22

For display purposes, the format consists of a six-digit value with floating decimal point, and an optional exponent. If the number to be displayed can be displayed in six digits or less, there is no exponent:

+3.14159 -654321 12 .001357 -.000032

The sign, '+' or '-', is optional. If the sign is not included, then '+' is assumed.
Numbers with more decimal places are displayed using Scientific Notation. This displays a six-digit number with decimal point and an exponent. The exponent part is indicated by the letter 'E' or 'e', the sign of the exponent ('+' or '-') and a two-digit number that is the exponent. For example:

.0000000004567 = 4.567e-10 3143286945 = 3.14329e+09

Note that in the second example some precision is lost, because there are only six significant digits possible.

Internally, floating point numbers are stored in single-precision 32-bit IEEE format. This format uses a 23-bit mantissa (the value portion), an 8-bit exponent, and a single sign bit.

It is important to note that 32 bits are required for storage. In the VED903 this requires two (2) consecutive 16-bit word registers, presumably %R.

RANGE
Given the single precision 32-bit IEEE format, acceptable values range from +/-3.40282E+38 (a very small fractional number) to +/-3.40282E+38 (a very large integer number).

SIGNIFICANT DIGITS
The real number format supports six (6) significant digits. When more than six (6) significant digits are displayed, only the first six can be counted on for accuracy.

3.14159265 = 3.14159 2535.00000045 = 2535

ENTERING FLOATING POINT VALUES
All floating numbers must adhere to the above format.
If an exponent is included, the mantissa (value) portion must also contain a decimal point. Note that if the entered format is other than x.yyy, the decimal point is moved and the exponent adjusted accordingly:

123.456e+3 = 123456 [The actual value can be displayed with six digits and no exponent]

143.643E-12 = 1.43643E-10 [Decimal point is moved and exponent adjusted]

A decimal point must be included to reduce any ambiguities. For example, 123e10 should be entered as 123.0e10, or better still 1.23e12.

Neither the mantissa nor the exponent may contain spaces.
"123 45e-12" and "4.3256e -23" will not be interpreted correctly because of the embedded spaces.

Both the mantissa and the exponent may contain a sign, + or -; i.e.:
"-1.3245e+12" or "4.243e-8". if the sign is missing then the associated part is assumed to be positive, **"1.2345e10".**

ERRORS

OVERFLOW is the most common error. This occurs when the result of a real number operation is greater than +3.40282E+38 or less than -3.40282E-38. For example, the equation
1.2345E-20 * 2.3456E-20
certainly causes this problem.

INFINITY
In case of an overflow result, power flow through the offending element is OFF, and the resulting value is set to Positive Infinity (if the value is greater than +3.40282E+38) or Negative Infinity (if the value is less than -3.40282E+38).

NOT A NUMBER (NAN)
If an infinity result is passed through to other calculations, the result can be undefined. This is know as Not a Number (NAN).
In the case of a NAN result, power flow through the offending element is OFF.
If a NAN result is passed through to another element, it feeds through to successive elements.

Register Types

Controllers offer a wide variety of Register Types. In most cases, the controller treats register types as if they were memory locations.
The following is a list of register types implemented in the VED903 and available for user´s.

%AI Analog Input
16-bit input registers used to gather analog input data such as voltages, temperatures, and speed settings coming from an attached device.
%AQ Analog Output
16-bit output registers used to send analog information such a voltages, levels or speed settings to an attached device.
%I Digital Input
Single-bit input registers. Typically, an external switch is connected to the registers.
%K Key Bit
Single-bit flags used to give the programmer direct access to any front panel keys appearing on a unit. Only the OCS series has keypads.
%Q Digital Output
Single-bit output registers. Typically, these bits are connected to an actuator, indicator light or alarm annunciator.
%R General Purpose Register
Retentive 16-bit registers.
%S System Bit
Single-bit bit coils predefined for system use.
%SR System Register
16-bit registers predefined for system use.
%T Temporary Bit
Non-retentive single-bit registers.

Bit-Mapped Addressing of 32-bit Registers

Bit-mapped addressing of 32-bit registers is not allowed. Bit offset values range from 1 to 16.
In order to access all 32 bits in a double register it is necessary to address the upper word of the register separately. Storage is such that the lower word is stored in the first (base) register, and the upper word is stored in the next consecutive register.
For example. if the 32-bit binary 0000000000000001 0000000000000100 value (65540 decimal) is loaded into register %R43, %R43 contains 0000000000000100 and %R44 contains 0000000000000001. Therefore, to check Bit 17 of the DWORD stored at %R43, one must check Bit 1 of %R44, addressed as %R44.1.

Numbering Base
In VED903 all offsets begin with 1 (one). 0 (zero) is not valid for register offset nor bit offset addressing.
Register offsets are thus in the range of 1 to X, where X is the maximum number of register in this model. For example, if the selected type has 2048 %R registers, they are addressed as %R01 through %R2048.

Bit Offsets are in the range of 1 to 16.

Groups of Boolean registers can be accessed as a 16-bit register. In this case, though, the Bit offset must lie on a 16-bit boundary, 1, 17, 33, etc.

Other Examples of Modbus Communication

READ OUTPUT STATUS (FUNCTION CODE 01)
This function allows the user to obtain the ON/OFF status of logic coils used to control discrete outputs from the addressed slave. In addition to the slave address and function fields, the message requires that the information field contain the initial coil address to be read, (Starting Address), and the number of locations that will be interrogated to obtain status data.

The following is an example of a message to Read Output Status Coils 20-56 from slave device number 17.

ADDR	FUNC	DATASTART PT HO	DATA START PT LO	DATA # OF PTS HO	DATA # OF PTS LO	ERROR CHECK FIELD
11	01	00	13	00	25	B6

An example response to Read Output Status is shown below. The data is packed one bit for each coil. The response includes the slave address, function code, quantity of data characters, and error checking. Data will be packed with one bit with one bit for each coil, (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follow. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeroes at the high end. The quantity of data characters is always specified as the quantity of RTU characters, i.e., the number is the same whether RTU or ASCII is used.

ADDR	FUNC	BYTE COUNT	DATA COIL STATUS20-27	DATA COIL STATUS28-35
11	01	05	CD	6B

DATA COIL STAT.36-43	DATA COIL STAT.44-51	DATA COIL STAT.52-56	ERROR CHECK FIELD
B2	0E	1B	D6

The status of coils 20-27 is shown as CD(HEX) = 1100 1101(Binary). Reading left to right, this shows that coils 27,26,23,22, and 20 are all on. The other coil data bytes are decoded similarly.

READ INPUT REGISTERS (FUNCTION CODE 04)

Function Code 04 obtains the contents of the controllers input registers. These locations receive their vales from devices connected to the I/O structure and can only be referenced, not altered from within the controller nor via MODBUS.
The example below requests the contents of register 30009 in slave number 17.

ADDR	FUNC	DATA START PT HO	DATA START PT LO	DATA # OF REGS HO	DATA # OF REGS LO	ERROR CHECK FIELD
11	04	00	08	00	01	E2

In the response message, the contents of register 30009 is decimal value 0.

ADDR	FUNC	BYTE COUNT	DATA INPUT REG HO 30009	DATA INPUT REG LO 30009	ERROR CHECK FIELD
11	04	02	00	00	E9

FORCE SINGLE COIL (FUNCTION CODE 05)

This message forces a single coil either On of OFF. Any coil that exists within the controller can be forced to either state, (ON or OFF). Coils are numbered from zero (i.e. coil 1 is address 0000, coil 2 is address 0001, etc.). The data value 65,280, (FF00 HEX) will set the coil ON and the value zero will turn it off. All other values are illegal and will not effect the coil. The use of slave address 00, (Broadcast mode), will force all attached slaves to modify the desired coil.
The example below requests slave number 17 to turn coil number 0173 ON.

ADDR	FUNC	DATACOILHO	DATA COILLO	DATA # ON/OFF	DATA	ERROR CHECK FIELD
11	05	00	AC	FF	00	3F

The normal response to the command request is to retransmit the message as received, after the coil state has been altered.

ADDR	FUNC	DATA COILHO	DATA COILLO	DATA # ON/OFF	DATA	ERROR CHECK FIELD
11	05	00	AC	FF	00	3F

PRESET SINGLE REGISTER (FUNCTION CODE 06)

Function 06 allows the user to modify the contents of a holding register. Any holding register that exists within the controller can have its contents changed by this message. The values are provided in binary up to the maximum capacity of the controller. Unused high-order bits must be set to zero. When used with slave address 00, all slave controllers will load the specified register with the contents specified.

ADDR / FUNC / DATA REGHO / DATA REGLO / DATA VALUEHO / DATAVALUE LO / ERROR CHECK FIELD						
11	06	00	87	03	9E	C1

The normal response to a preset single register request is to retransmit the query message after the register has been altered.

ADDR / FUNC / DATA REGHO / DATA REGLO / DATA VALUEHO / DATAVALUE LO / ERROR CHECK FIELD						
11	06	00	87	03	9E	C1

READ INPUT STATUS (FUNCTION CODE 02)

This function allows the user to obtain the ON/OFF status of discrete inputs in the addressed slave.In addition to the slave address and function code fields, this message requires that the information field contain the initial input address to be read, (Starting Address) and the number of locations that will be interrogated to obtain the status data.
The following is an example of a message to Read Input Status Coils 10197-10218 from slave device number 17.

ADDR / FUNC / DATA START PT HO / DATA START PT LO / DATA # OF PTS HO / DATA # OF PTS LO / ERROR CHECK FIELD						
11	02	00	C4	00	16	13

An example response to Read Input Status is shown below. The data is packed one bit for each coil. The response includes the slave address, function code, quantity of data characters, and error checking. Data will be packed with one bit with one bit for each coil, (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follow. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeroes at the high end. The quantity of data characters is always specified as the quantity of RTU characters, i.e., the number is the same whether RTU or ASCII is used.

ADDR / FUNC / BYTE COUNT / DATA DISCRETE INPUT10197-10204 / DATA DISCRETE INPUT10205-10212				
11	02	03	AC	DB

DATA DISCRETE INPUT10213-10218 / ERROR CHECK FIELD		
35	2E	LRC

The status of inputs 10197-10204 is shown as AC (HEX) = 1010 1100 (Binary). Reading left to right, this shows that inputs 10204, 10202, 10200 and 10099 are all on. The other input data bytes are decoded similarly.

READ OUTPUT REGISTERS (FUNCTION CODE 03)

Read Output Registers allows the user to obtain the binary contents of holding registers in the addressed slave.
These registers can store the numerical values of associated timers and counters which can be driven to external devices.
The following example reads registers 40108 through 40110 from slave number 17.

ADDR / FUNC / DATA START PT HO / DATA START PT LO / DATA # OF REGS HO / DATA # OF REGS LO / ERROR CHECK FIELD						
11	03	00	6B	00	03	7E

The addresses slave responds with its address and the function code, followed by the information field. The information field contains 2 bytes describing the quantity of data bytes to be returned. The contents of the registers requested (DATA), are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, low order bits.
In the example below, the registers 40108-40110 have the decimal contents 555, 0, and 100 respectively.

ADDR / FUNC / BYTE COUNT / DATA OUTPUTREGH.O.40108 / DATA OUTPUTREGL.O.40108				
11	03	06	02	2B

DATA OUTPUTREGH.O.40109 / DATA OUTPUTREGL.O.40109 / DATA OUTPUTREGH.O.40110		
00	00	00

DATA OUTPUTREGL.O.40110 / ERROR CHECK FIELD	
64	55

Modbus Programming

01: Actions: In this sub-item can be programmed the actions that will be taken in each one of the possible fail. The options for whole the fails are: “None”, “Alarm”, “Inhibit”, “Trip”, “Both”, “Force Field Current” and “Forces Open Loop.” **See Buletin 246C for more details.**
For each adress set as follow:
1 = None
2 = Alarm
4 = Inhibit
8 = Trip
16 = Both
32 = Forces Field Current
64 = Force Open Loop

1.01/A: Under Voltage Action.
%R101 8 bit Immediate
1.01/B: Over Voltage Action.
%R102 8 bit Immediate

1.02/A: Under Frequency Action. Line frequncy under the programmed limit.
%R103 8bit Immediate
1.02/B: Over Frequency Action. Line frequency over the programmed limit.
%R104 8bit Immediate

1.03/A: Under Excitation Action. Load current under the programmed limit.
%R105 8 bit Immediate
1.03/B: Over Excitation Action. Load current over the programmed limit.
%R106 8 bit Immediate

1.04/A: Line Over Lead Action.
%R107 8 bit Immediate
1.04/B: Line Over Lag Action.
%R108 8 bit Immediate

1.05/A: Line Under Current Action. Line current, under the programmed limit.
%R109 8 bit Immediate
1.05/B: Line Over Current Action.
%R110 8 bit Immediate

1.06A: Line Under Power Action.
%R111 8 bit Immediate
1.06B: Line Over Power Action
%R112 8 bit Immediate
1.07A: Motoring Action.
1.07B: Motoring.
%R113 8 bit Immediate
1.07B: Inadvertent Energization Action.
%R116 8 bit Immediate

1.08A: Exciter Over Temperature.
%R114 8 bit Immediate
1.08B: Field Over Temperature.
%R118 8 bit Immediate

1.09: External Fail Action.
%R115 8 bit Immediate

1.10/A: Field Loss (Voltage) Action.
%R099 8 bit Immediate
1.10/B: Field Short Circuit Ation.
%R119 8 bit Immediate

1.11: Lost of Control Action.
%R117 8 bit Immediate

02 - Modes: In this sub-item can be programmed the Operation modes, Settling modes, Regulation modes and others:

02.1: Regulation Mode:
Values:
1: Voltage Constant
2: Power Factor Constant
4: KVAR Constante
8: Volts/Hz Constant (U/F)
16: Volt/PF = Compound PF
32: Volt/VAR = Compound VAR
64 : Volt/PF/Hz = U/F + Compound PF
128: Volt/VAR/Hz = U/F + Compound VAR
Address:
%R123 8 bit Flagged

2.02A/B: Manual to Auto Transfer / Auto to Man Transfer
Values:
1: Mantain Process Value
2: Setting = 0%
4: Setting = 50%
8: Setting = 100%
16: Setting = Nominal
32: Not Change Setpoint
64: Preset
Addresses:
%R131 8 bit Flagged
%R132 8 bit Flagged

2.03: Output Parameter:
Values:
0: None
1: Setting Scale
2: Excitation Current
4: Line Voltage
8: Line Current
16: Power Factor
32: Line KVA
64: Line KVAR
128: Line Frequency
256: Set after Range
512: Set Final
1024: Control Value
2048: Field Voltage
4096: Field Temperature
Address:
%R124 16 bit Immediate

2.04: Up Down Start Mode **Values:**
1: 0% of Range
2: 50% of Range
4: 100% of Range
8: Last Value
16: Nominal
32: Preset Scale
Address:
%R125 8 bit Immediate

2.05A: Soft Start **Values:**
0: Disable
1: Slow
2: Medium
4: Fast
%R154 8 bit Flagged
2.05B: Force Channel
Values: **0**: CH1 (PSP) **1**: CH2(SSP)
%R136 8 bit Immediate

2.06/A: Under Excitation Limit Flag.
Values: **0**: No **1**: Yes
%R151.3 8 bit Immediate
2.06/B: Over Excitation Limit Flag.
Values: **0**: No **1**: Yes
%R151.4 8 bit Immediate

2.07/A: Lead Limit Flag.
Values: **0**: No **1**: Yes
%R151.5 1 bit Immediate
2.07/B: Lag Limit Flag.
Values: **0**: No **1**: Yes
%R151.6 1 bit Immediate

2.08: Adjuste Range.
Values: **0**: +/- 20% **1**: +/- 100%
%R151.8 1 bit Immediate

2.09: Operation Mode.
Values: **0**: Automatic
1: Manual Field Current
2: Manual Open Loop
%R128 8 bit Flagged

2.10A: PSP to SSP
Values: **0**: No Chg **1**:Preset
%R138 8 bit Immediate
2.10B: SSP to PSP
Values: **0**: No Chg **1**:Preset
%R139 8 bit Immediate

2.11A: Volts/Hz Start Point
Values: **0**: Disable **1**: Enable
%R151.9 1 bit Immediate

2.11B: Line Current Compensation.
Values: 0: Disable **1**: Enable
%R151.11 1 bit Immediate

2.12A: FCX Modes. **Values:**
1: Delay
2: Switch
4: Switch + Delay
8: Load
16: Switch + Load
32: Delay +Load
64: Delay +Load + Switch
Address:
%R133 8 bit Flagged
2.12B: Power Mode.
Values: 0: Generator **1**: Auxiliary
%R152.7 1 bit Immediate

2.13A: Mute Mode.
Values: 0: Manual / **1**: Auto / **2**: Locked
%R159 8 bit Immediate
2.13B: Reset Mode.
Values: 0: Manual / **1**: Auto / **2**: Locked
%R160 8 bit Immediate

2.14: Setting Modes. **Values:**
0: Up/Down Only
1: Keyboard Only
2: Up/Down + Keyboard
3: Set 1 = Pot / 0 - 5Volt/0-20 mA
Address:
%R703 8 bit Flagged

2.15A: Power Factor Sensing Mode.
Values: 0: Internal **1**: External
%R152.8 1 bit Immediate
2.15B: Channel.
Values: 0: Single **1**: Double
%R773.1 1 bit Immediate

2.16A/B/C: Digital Output 1/2/3.
1: Start
2: FAR
4: FCX
8: Alarm
16: Trip
Address:
%R134 8 bit Immediate
%R135 8 bit Immediate
%R137 8 bit Immediate

2.17: Control Input. **Values:**
1: Mute/Reset
2: Force Open Loop
4: Force Field Current
8: Follow Field Voltage
16: Internal Hot Backup
32: External Hot Backup
64: Follow Set Point
128: Enable PSS
Address:
%R156 16 bit Flagged

2.18A: Analog Output Scale Min.
Values: Min: 0 **Max**: 90
%R913 Float. Point-32Bit Immediate
2.18B: Analog Output Scale Max.
Values: Min: 10 **Max**: 100
%R915 Float. Point-32Bit Immediate

2.19A: VAR Limit.
Values: 0: No **1:** Yes
%R151.13 1 Bit Immediate
2.19B: Load Angle Limit.
Values: 0: No **1:** Yes
%R151.14 1 Bit Immediate

2.20A: PSS Enable.
Values: 0: No **1:** Yes
%R151.15 1 Bit Immediate
2.20B: PSS Signal Polarity.
Values: 0: Normal **1:** Inverse
%R151.16 1 Bit Immediate
2.20C: PSS Signal Compound %.
Values: Min: 0 **Max**: 100
%R233 Float. Point-32Bit Immediate

03: Delays.
In this sub-item are the delays for fail detection of each fail, delay for "Auto-Mute" and delay for Auto-Reset, Time of Trip Pulse and others are settled. Notice that value settled are multiplied by 0,1 seconds. For example for 1 second is due to set the value = 10.
3.1/A: Line Under Voltage Delay (0,1 to 1000 Seconds). Values: 1 to 10000
%R178 16 bit Immediate
3.1/B: Line Over Voltage Delay (0,1 to 1000 Seconds). Values: 1 to 10000
%R179 16 bit Immediate

3.2/A: Line Under Frequency Delay (0,1 to 1000 Seconds).
Values: 1 to 10000
%R180 16 bit Immediate
3.2/B: Line Over Frequency Delay (0,1 to 1000 Seconds.
Values: 1 to 10000
%R181 16 bit Immediate

3.3/A: Under Excitation Delay (0,1 to 1000 Seconds). Values: 1 to 10000
%R182 16 bit Immediate
3.3/B: Over Excitation Delay (0,1 to 1000 Seconds.
Values: 1 to 10000
%R183 16 bit Immediate

3.4/A: Line Over Lead Delay (0,1 to 1000 Seconds. **Values:** 1 to 10000
%R184 16 bit Immediate
3.4/B: Line Over Lag Delay (0,1 to 1000 Seconds. **Values:** 1 to 10000
%R185 16 bit Immediate

3.5/A: Line Under Current Delay (0,1 to 1000 Seconds. **Values:** 1 to 10000
%R186 16 bit Immediate
3.5/B: Line Over Current Delay (0,1 to 1000 Seconds. **Values:** 1 to 10000
%R187 16 bit Immediate

3.6A: Line Under Power Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R188 16 bit Immediate
3.6B: Line Over Power Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R189 16 bit Immediate

3.7: FCX Delay (0,1 to 1000 Seconds).
Values: 1 to 10000
%R190 16 bit Immediate

3.8A: Exciter Over Temperature Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R191 16 bit Immediate
3.8B: Field Over Temperature Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R195 16 bit Immediate

3.9A: External Fail Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R192 16 bit Immediate
3.9B: Lost of Control Delay (0.1 to 1000 Seconds). **Values:** 1 to 10000
%R194 16 bit Immediate

3.10A: Inadvertent Energization Delay (0,1 to 1000 Seconds). **Values:** 1 to 10000
%R193 16 bit Immediate
3.10B: Motoring Delay (0.1 to 1000 Seconds). **Values:** 1 to 10000
%R198 16 bit Immediate

3.11A: Auto Mute Time (0.1 to 1000 Seconds).Values: 1 to 10000
%R196 16 bit Immediate
3.11B: Auto Reset Time (0.1 to 1000 Seconds). **Values:** 1 to 10000
%R197 16 bit Immediate

3.12: Trip Pulse Time (0.1 to 1000 Seconds or infinite (infinite = 0)).
Values: 0 to 10000
%R199 16 bit Immediate

3.13A: Field Loss Time (0.1 to 1000 Seconds or infinite (infinite = 0)).
Values: 0 to 10000
%R200 16 bit Immediate
3.13B: Follow Time (0.1 to 100 Seconds).
Values: 0 to 1000
%R177 16 bit Immediate
3.14: Over Excit Boost Time (0.1 to 100 secondes). **Values:** 0 to 1000
%R177 16 bit Immediate

04: Limts / Rates:
In this sub-item the parameters for the Limitation functions and Trip protections, the Droop rate, Compound Rate, Load Current Compensation Rate and polarity of the Power Factor reading are settled.
04.1/A: Line Under Voltage Trip.
Values: 0% to 100.00%
%R235 Float. Point-32Bit Flagged
04.1/B: Line Over Voltage Trip.
Values: 100% to 180.00%
%R237 Float. Point-32Bit Flagged
04.2/A: Line Under Frequency Trip.
Values: 10% to 100.00%
%R239 Float. Point-32Bit Flagged
04.2/B: Line Over Frequency Trip.
Values: 100% to 200.00%
%R241 Float. Point-32Bit Flagged

04.3/A: Under Excitation Limit.
Values: 0% to 100.00%
%R203 Float. Point-32Bit Flagged
04.3/B: Over Excitation Limit.
Values: 70% to 300.00%
%R207 Float. Point-32Bit Flagged

04.4/A: Under Excitation Trip.
Values: 0% to 100.00%
%R205 Float. Point-32Bit Flagged
04.4/B: Over Excitation Trip.
Values: 70% to 300.00%
%R209 Float. Point-32Bit Flagged

04.5/A: Over Lag Limit.
Values: 0.0 to 0.99
%R211 Float. Point-32Bit Immediate
04.5/B: Over Lead Limit.
Values: 0.0 to 0.99
%R215 Float. Point-32Bit Immediate

04.6/A: Over Lead Trip.
Values: 0.0 to 0.99
%R213 Float. Point-32Bit Immediate
04.6/B: Over Lag Trip.
Values: 0.0 to 0.99
%R217 Float. Point-32Bit Immediate

04.7/A: Line Under Current Trip.
Values: 0% to 100.00%
%R243 Float. Point-32Bit Flagged
04.7/B: Line Over Current Trip.
Values: 70% to 200.00%
%R245 Float. Point-32Bit Flagged

04.8/A: Under Power Trip.
Values: 0% to 100.00%
%R247 Float. Point-32Bit Flagged
04.8/B: Over Power Trip.
Values: 70% to 200.00%
%R249 Float. Point-32Bit Flagged

04.9A: Power Factor Read Point.
Values: 0 to 100% of Line Current
%R201 Float. Point-32Bit Immediate
04.9B: Build Up Point.
Values: 0 to 100% of Line Voltage
%R251 Float. Point-32Bit Immediate

04.10/A: Droop Rate. **Values:** 0 to 10%
%R219 Float. Point-32Bit Immediate
04.10/B: Compound Rate.
Values: 0% to 10%
%R221 Float. Point-32Bit Immediate

04.11A,B,C,D: Limiting PID.
A)P Gain (0 to 9999) x 0.1
%R344 16 Bit Integer Immediate
B)D Time (0 to 9999) x 0.1
%R345 16 Bit Integer Immediate
C)I Rate (0 to 9999) x 0.1
%R346 16 Bit Integer Immediate
D)Slew Time (0 to 999) x 0.1
%R310 16 Bit Integer Immediate

04.12/A: Line Current Compens. Rate.
Values: 0.0% to 15.00%
%R223 Float. Point-32Bit Flagged
04.12/B: V/Hz Operation Point.
Values: 0.0% to 15.00%
%R229 Float. Point-32Bit Flagged

04.13/A: P.F. Polarity (Normal / Inverse).
Values: 0: Normal **1**: Inverse
%R152.16 1 bit Flagged
04.13/B: Max Field Temperature.
Values: 25ºC to 200ºC
%R227 Float. Point-32Bit Flagged

04.14/A: VAR limit %.
Values: 0 to 100 %
%R257 Float. Point-32Bit Flagged
04.14/B: Polar Angle Limit.
Values: 0 to 90
%R433 Float. Point-32Bit Flagged

04.15: Follow Span (0 to 20%).
Values: 0 to 20.00
%R231 Float. Point-32Bit Immediate

05: PID Calibration.
In this sub-item are settled the parameters relative to the digital error amplifier with PID action.

05.01/A: Dead Band +/- PID 1.
Values: 0 to 1000
%R302 16Bit Immediate
05.01/B: Dead Band +/- PID 2.
Values: 0 to 1000
%R322 16Bit Immediate

05.02/A: Derivative Sensitivity PID 1.
Values: 0: Normal **1:** Reduced
%R311.14 1Bit Immediate
05.02/B: Integral Clamping PID 1.
Values: 0: No **1:** Yes
%R311.13 1Bit Immediate
05.02/C: Derivative Sensitivity PID 2.
Values: 0: Normal **1:** Reduced
%R331.14 1Bit Immediate
05.02/D: Integral Clamping PID 2.
Values: 0: No **1:** Yes
%R331.13 1Bit Immediate

05.03/A: Proportional Gain PID 1.
Values: 0 to 32767 (x 0.01)
%R304 16Bit Immediate
05.03/B: Proportional Gain PID 2.
Values: 0 to 32767 (x 0.01)
%R324 16Bit Immediate

05.04/A: Derivative Time PID 1.
Values: 0 to 32767 (x 0.01 S)
%R305 16Bit Immediate
05.04/BA: Derivative Time PID 2.
Values: 0 to 32767 (x 0.01 S)
%R325 16Bit Immediate

05.05/A: Integral Rate PID 1
Values: (0 a 32767 r/1000 S).
%R379 16Bit Flagged
05.05/B: Integral Rate PID 2
Values: (0 a 32767 r/1000 S).
%R389 16Bit Flagged

05.06/A: Minimum Slew Time PID 1.
Values: 0 to 1000 S
%R310 16Bit Immediate
05.06/A: Minimum Slew Time PID 2.
Values: 0 to 1000 S
%R330 16Bit Immediate

05.07/A: Derivative Term PID 1 (Error = PV-SP) ou PV = Process Value).
Values: 0: Error **1**: PV (Process Value)
%R311.11 1Bit Immediate
05.07/B: Derivative Term PID 2 (Error = PV-SP) ou PV = Process Value).
Values: 0: Error **1**: PV (Process Value)
%R331.11 1Bit Immediate

05.08/A: Clamping Low PID 1.
Values: 0 to 32000
%R309 16 Bit Immediate
05.08/B: Integral HighPID 1.
Values: 0 to 32000
%R308 16 Bit Immediate
05.08/C: Clamping Low PID 2.
Values: 0 to 32000
%R329 16 Bit Immediate
05.08/D: Clamping High PID 2.
Values: 0 to 32000
%R328 16 Bit Immediate

Menu 06: Filter Calibration.
In this sub-item are settled the filter depth in four reading inputs and two outputs.
06.01/A/B/C/D/E/F: Filter 1 to 6.
Values: 0 to 99
%R361 16Bit Flagged

%R363 16Bit Flagged
%R365 16Bit Flagged
%R367 16Bit Flagged
%R368 16Bit Flagged
%R369 16Bit Flagged

07: Custom Calibration.
In this sub-item can be made digital calibrations for the several readings, compensating the errors of Transducers, Voltage Transformers and Current Transformers used by the user.

07.01/A: Line Voltage Zero. (**0** = No correction). **Values:** -1000.00 to 1000.00
%R971 Float. Point-32Bit Immediate
07.01/B: Line Voltage Scale. (**1** = No correction). **Values:** 0 to 9.9999
%R951 Float. Point-32Bit Immediate

07.02A: Line Current Zero. (**0** = No correction). **Values:** -1000.00 to 1000.00
%R973 Float. Point-32Bit Immediate
07.02B: Line Current Scale. (**1** = No correction). **Values:** 0 to 9.9999
%R953 Float. Point-32Bit Immediate

07.03/A: Field Current Zero. (**0** = o correction). **Values:** -1000.00 to 1000.00
%R975 Float. Point-32Bit Immediate
07.03/B: Field Current Scale. (**1** = No correction). **Values:** 0 to 9.9999
%R955 Float. Point-32Bit Immediate

07.04/A: Field Voltage Zero. (**0** = No correction). **Values:** -1000.00 to 1000.00
%R977 Float. Point-32Bit Immediate
07.04/B: Field Voltage Scale. (**1** = No correction). **Values:** 0 to 9.9999
%R961 Float.Point-32Bit Immediate
07.05:Set 1- Pot/0-5V Calibration (**1** = No correction)**. Values:** -10.00 to 10.00
%R959 Float. Point-32Bit Immediate

07.06/A: Scale Shift PID 1 (**0** = No correction)**. Values:** -50.00% to 50.00%
%R957 Float. Point-32Bit Flagged
07.06/B: Scale Shift PID 2 (**0** = No correction)**. Values:** -50.00% to 50.00%
%R967 Float. Point-32Bit Flagged

10: Set Nominal.
In this sub-item should be settled the nominal values of the system.
10.01/A: Nominal Generator KVA.
Values: 0.1 to 9999999.0
%R401 Float.Point-32Bit Flagged
10.01/B: Nominal Line Frequency.
Values: 1 to 999
%R419 Float.Point-32Bit Flagged
10.01/C: Nominal Generator Power Factor. Values: 0.10 to 1.00
%R421 Float.Point-32Bit Flagged

10.02/A: Nominal Line Voltage.
Values: 10 to 99999
%R403 Float.Point-32Bit Flagged
10.02/B: Nominal Sensing Transformer Input Voltage. **Values:** 10 to 99999
%R405 Float.Point-32Bit Flagged
10.02/C: Nominal Sensing Transformer Output Voltage. **Values:** 10 to 256
%R407 Float.Point-32Bit Flagged

10.03/A: Nominal Line Current.
Values: 10 to 99999
%R409 Float.Point-32Bit Flagged
10.03/B: Line Current Transformer Input. Values: 10 to 99999
%R411 Float.Point-32Bit Flagged
10.03/C: Line Current Transformer Output. **Values:** 0.01 to 10.00
%R413 Float.Point-32Bit Flagged

10.04/A: Nominal Field Current.
Values: 1.00 to 9999.00
%R415 Float.Point-32Bit Flagged
10.04/B: Field Current Transducer Rate (A / 5V).
Values: 0.01 to 999.99
%R417 Float.Point-32Bit Flagged

10.05/A: Excitation Transformer.
Values: 5 to 2500
%R423 Float.Point-32Bit Flagged
10.05/A: Nominal Field Voltage.
Values: 1 to 2000
%R425 Float.Point-32Bit Flagged
10.05/B:Field Voltage Transducer Rate (VCC/10V).
Values: 5 to 2500
%R427 Float.Point-32Bit Flagged
10.06/A:Field Resistance (Ohm).
Values: 0 to 10000
%R435 Float.Point-32Bit Flagged
10.06/B:Field Synch. Reactance (Ohm).
Values: 0 to 1000
%R431 Float.Point-32Bit Flagged
10.07:Field Current w/ No Load (A).
Values: 1 to 10000 A
%R429 Float.Point-32Bit Flagged

11: Preset Primary Set Point (PID 1) And Secondary Set Point (PID 2)
11/A: PSP Primary Set Point. Values: 00.00% to 100.00%
%R761 Float. Point-32Bit Imediate
11/B: SSP Secondary Set Point. Values: -00.00% to 100.00%
%R763 Float. Point-32Bit Imediate

12: PID Auto Tune.
In this sub-item the PID Auto Tune function can be started. The value of T120 register returns to zero after auto-tune.
Values: 0: do notting 1: Start
%T120 1Bit Immediate

14: Set Clock.
14.01/A: New Year. Values: 2000 to 2100
%R0655 16 bit Immediate
14.01/B: New Month. Values: 1 to 12
%R0654 16 bit Immediate
14.01/C: New Day. Values: 1 to 31
%R0653 16 bit Immediate
14.01/D: New Hour. Values: 1 to 24
%R0652 16 bit Immediate
14.01/B: New Minute. Values: 0 to 59
%R0651 16 bit Immediate

Special functions
These functions are for factory calibration and they are not documented in the manual. In special case the user can use it via Modbus, but carefully. The items numbers are not sequential to combine with the real factory items.

V01/A: Polar Angle Limit Mode.
Values: 0: W/ EXC (pu) 1: W/ PF
%R151.10 1 Bit Immediate
V01/B: Polar Angle Limit Calibration
Values: 0 to 10
%R1371 FloatPoit - 32Bit Immediate

V02/A: PID 1 Bias. Values: -32000 to 32000
%R307 16 Bit Immediate
F02/B: PID 2 Bias. Values: -32000 to 32000
%R327 16 Bit Immediate
F02/C: PID 3 Bias. Values: -32000 to 32000
%R347 16 Bit Immediate

V03/A: PID 1 Upper/Lower Clamp Mode.
Values: 0: Manual **1**: Automatic
%R152.13 1 Bit Immediate
F03/B: PID 1 Upper/Lower Clamp Mode.
%R152.14 1 Bit Immediate

V04: Bridge Type.
Values: 0: IGBT (PWM) **1**: Tirystor
%R152.1 1 Bit Immediate

V14: Power Factor Reading Calibration (1 = no change)**. Values:** 0 to 10.00
%R929 Float.Point-32 Bit Flagged

V15: Frequency Reading Calibration (1 = no change)**. Values:** 0 to 10.00
%R931 Folat.Point-32 Bit Flagged

Readings:
These functions are only for reading of the values and events. The user should not modify through Modbus. Use the menu in the VED903 for eventual reset of the events memorized.

F3/01: Last Event / 1º Fail
%R1001 16 Bit Read Only

L01/A: Last Event Hour.
%R1012 16 Bit Read Only
L01/B: Last Event Minute.
%R1013 16 Bit Read Only
L01/C: Last Event Day.
%R1014 16 Bit Read Only
L01/D: Last Event Month.
%R1015 16 Bit Read Only
L01/E: Last Event Year.
%R1016 16 Bit Read Only

L02/A: Last Excitation Hour.
%R1002 16 Bit Read Only
L02/B: Last Excitation Minute.
%R1003 16 Bit Read Only
L02/C: Last Excitation Day.
%R1004 16 Bit Read Only
L02/D: Last Excitation Month.
%R1005 16 Bit Read Only
L02/E: Last Excitation Year.
%R1006 16 Bit Read Only

L03/A: Last De-excitation Hour.
%R1007 16 Bit Read Only
L03/B: Last De-excitation Minute.
%R1008 16 Bit Read Only
L03/C: Last De-excitation Day.
%R1009 16 Bit Read Only
L03/D: Last De-excitation Month.
%R1010 16 Bit Read Only
L03/E: Last De-excitation Year.
%R1011 16 Bit Read Only

L04: Total Excitation Times.
%R1022 16 Bit Read Only

L05/A: Total Excited Hours.
%R1025 16 Bit Read Only
L05/B: Total Excited Minute.
%R1027 16 Bit Read Only

L06/A: Total Life Hour.
%R1029 16 Bit Read Only
L06/B: Total Life Minute.
%R1031 16 Bit Read Only

Table for values from L07 to L18
Value:
0: None
1: Undervoltage
2: Overvoltage
3: Underfrequency
4: Overfrequency
5: Lield Undercurrent
6: Field Overcurrent
7: Overleading
8: Overlaging
9: Line Undercurrent
10: Line Overcurrent
11: Line Underpower
12: Line Overpower
13: Motoring / Inadv. Energization
14: Exciter Overtemperature
15: External Fail
16: Field Loss
17: Lost Control
18: Field Overtemperature
19: Field Short Circuit

L07/A: Memory 1 Event (See Table).
%R571 16 Bit Read Only
L07/B: Memory 1 Hour.
%R501 16 Bit Read Only
L07/C: Memory 1 Minute.
%R502 16 Bit Read Only
L07/D: Memory 1 Day.
%R503 16 Bit Read Only
L07/E: Memory 1 Month.
%R504 16 Bit Read Only
L07/F: Memory 1 Year.
%R505 16 Bit Read Only

L08/A: Memory 2 Event (See Table).
%R572 16 Bit Read Only
L08/B: Memory 2 Hour.
%R506 16 Bit Read Only
L08/C: Memory 2 Minute.
%R507 16 Bit Read Only
L08/D: Memory 2 Day.
%R508 16 Bit Read Only
L08/E: Memory 2 Month.
%R509 16 Bit Read Only
L08/F: Memory 2 Year.
%R510 16 Bit Read Only

L09/A: Memory 3 Event (See Table).
%R573 16 Bit Read Only
L09/B: Memory 3 Hour.
%R511 16 Bit Read Only
L09/C: Memory 3 Minute.
%R512 16 Bit Read Only
L09/D: Memory 3 Day.
%R513 16 Bit Read Only
L09/E: Memory 3 Month.
%R514 16 Bit Read Only
L09/F: Memory 3 Year.
%R515 16 Bit Read Only

L10/A: Memory 4 Event (See Table).
%R574 16 Bit Read Only
L10/B: Memory 4 Hour.
%R516 16 Bit Read Only
L10/C: Memory 4 Minute.
%R517 16 Bit Read Only
L10/D: Memory 4 Day.
%R518 16 Bit Read Only
L10/E: Memory 4 Month.
%R519 16 Bit Read Only
L10/F: Memory 4 Year.
%R520 16 Bit Read Only

L11/A: Memory 5 Event (See Table).
%R575 16 Bit Read Only
L11/B: Memory 5 Hour.
%R521 16 Bit Read Only
L11/C: Memory 5 Minute.
%R522 16 Bit Read Only
L11/D: Memory 5 Day.
%R523 16 Bit Read Only
L11/E: Memory 5 Month.
%R524 16 Bit Read Only
L11/F: Memory 5 Year.
%R525 16 Bit Read Only

L12/A: Memory 6 Event (See Table).
%R576 16 Bit Read Only
L12/B: Memory 6 Hour.
%R526 16 Bit Read Only
L12/C: Memory 6 Minute.
%R527 16 Bit Read Only
L12/D: Memory 6 Day.
%R528 16 Bit Read Only
L12/E: Memory 6 Month.
%R529 16 Bit Read Only
L12/F: Memory 6 Year.
%R530 16 Bit Read Only

L13/A: Memory 7 Event (See Table).
%R577 16 Bit Read Only
L13/B: Memory 7 Hour.
%R531 16 Bit Read Only
L13/C: Memory 7 Minute.
%R532 16 Bit Read Only
L13/D: Memory 7 Day.
%R533 16 Bit Read Only
L13/E: Memory 7 Month.
%R534 16 Bit Read Only
L13/F: Memory 7 Year.
%R535 16 Bit Read Only

L14/A: Memory 8 Event (See Table).
%R578 16 Bit Read Only
L14/B: Memory 8 Hour.
%R536 16 Bit Read Only
L14/C: Memory 8 Minute.
%R537 16 Bit Read Only
L14/D: Memory 8 Day.
%R538 16 Bit Read Only
L14/E: Memory 8 Month.
%R539 16 Bit Read Only
L14/F: Memory 8 Year.
%R540 16 Bit Read Only

L15/A: Memory 9 Event (See Table).
%R579 16 Bit Read Only
L15/B: Memory 9 Hour.
%R541 16 Bit Read Only
L15/C: Memory 9 Minute.
%R542 16 Bit Read Only
L15/D: Memory 9 Day.
%R543 16 Bit Read Only
L15/E: Memory 9 Month.
%R544 16 Bit Read Only
L15/F: Memory 9 Year.
%R545 16 Bit Read Only

L16/A: Memory 10 Event (See Table).
%R580 16 Bit Read Only
L16/B: Memory 10 Hour.

%R546 16 Bit Read Only
L16/C: Memory 10 Minute.
%R547 16 Bit Read Only
L16/D: Memory 10 Day.
%R548 16 Bit Read Only
L16/E: Memory 10 Month.
%R549 16 Bit Read Only
L16/F: Memory 10 Year.
%R550 16 Bit Read Only

L17/A: Memory 11 Event (See Table).
%R581 16 Bit Read Only
L17/B: Memory 11 Hour.
%R551 16 Bit Read Only
L17/C: Memory 11 Minute.
%R552 16 Bit Read Only
L17/D: Memory 11 Day.
%R553 16 Bit Read Only
L17/E: Memory 11 Month.
%R554 16 Bit Read Only
L17/F: Memory 11 Year.
%R555 16 Bit Read Only

L18/A: Memory 12 Event (See Table).
%R582 16 Bit Read Only
L18/B: Memory 12 Hour.
%R556 16 Bit Read Only
L18/C: Memory 12 Minute.
%R557 16 Bit Read Only
L18/D: Memory 12 Day.
%R558 16 Bit Read Only
L18/E: Memory 12 Month.
%R559 16 Bit Read Only
L18/F: Memory 12 Year.
%R560 16 Bit Read Only

L19/A: Line Voltage.
%R863 Float.Poin -32 Bit Read Only
L19/B: Line Current.
%R825 Float.Poin -32 Bit Read Only
L19/C: Line Frequency.
%R831 Float.Poin -32 Bit Read Only

L20/A: Line KVA / MVA.
%R865 Float.Poin -32 Bit Read Only
L20/B: KAV / MVA Factor(for L20A).
%R153.10 1Bit Read Only
0= KVA 1= MVA
L20/C: Line KVAR / MVAR.
%R867 Float.Poin -32 Bit Read Only
L20/D: KAV / MVA Factor(for L20A).
%R153.11 1Bit Read Only
0= KVAR 1= MVAR
L20/E: Quadrant = Generator / Motor.
%R153.4 1Bit Read Only
0= Generator 1= Motor

L21/A: Line Power Factor Value.
%R829 Float.Poin -32 Bit Read Only
L21/B: Line Power Factor Angle.
%R775 8 bit Read Only
Values: **0**: Lag **1**: Lead

L21/C: Line KW /MW.
%R869 Float.Poin -32 Bit Read Only

L21/D: KW / MW Factor(for L21C).
%R153.12 1Bit Read Only
0= KW 1= MW

L22: Field Current.
%R827 Float.Poin -32 Bit Read Only
L22: Field Voltage.
%R821 Float.Poin -32 Bit Read Only
L22: Field Temperature.
%R857 Float.Poin -32 Bit Read Only

L23/A: Actual State.
Values: 1:Standby. **2:**Buikdup **3:**Soft Starting **5:** Fail **6:** Alarm **7:**Excited **8:** Loaded.
%R701 16 Bit Read Only
L23/B: Main variable.
Values: 1:Volt. **2:**Power Factor **3:**KVAR **4**:MVAR. **5:**Volt/PF **6:**Volt/VAR **7**:Open Loop. **8**:V/Hz **9:**V/PF/HZ **10:**V/VAR/Hz **11:**Field Amp.
%R143 8 Bit Read Only
L23/C: Force Status.
Values: 0: Normal **1:**Forced
%R130.3 8 Bit Read Only
L23/D: Seting Limit Status.
Values: 0: Normal **1:**Forced
%R771 8 Bit Read Only
L23/E: Forced Setting Value.
%R781 Real/Floating Point Read Only
L23/F: Actual Variable.
%R777 Real/Floating Point Read Only

L24/A: Limit Status.
Values: 0:No. **1:**Limiting Upper Current **2:**Limiting Lower Current.
%R157 8 Bit Read Only
L24/B: Final Setting (0 to 100%).
%R709 Float.Point - 32 Bit Read Only
L24/C: Set Value.
%R705 Float.Point - 32 Bit Read Only
L24/D: Set Type.
Values: 1:Amper **2**: Voltage **3**: KW
%R143 16 Bit Read Only
L24/E: Main Variable Complement.
Values: 0:Lag. **1:**Lead **3:**Forced **4**:Volt **5:**Field Current **6:**PF **7**: Amper.
%R779 8 Bit Read Only
L24/F: Process Limit Status.
0:No. **1:**Limiting Lower **2:**Limiting Upper.
%R158 16 Bit Read Only

L25/A: Operating Status.
1:Normal **2:**Alarm Active **2:**Fail Active **3:**Mutted.
%R129 8 Bit Read Only
L25/B: Actual Operating Mode.
0:Automatic **1**:Man_Open_Loop 3 **3:**Man_Field Current
%R702 8 Bit Read Only

L25/C: Souce After Force. **Values:**
0: Up/Down Only
1: Keyboard Only
2: Up/Down + Keyboard
3: Set 1= Pot/0-5Volt Only
4: Set 2 = 0 to 20 mA Only
5: Set 1 + Set 2
6: Up/Down + Set 2
7: Keyboard + Set 2
8: Set 1 + Up/Down
9: Set1 + Up/Down + Set 2
%R717 8Bit Read Only

The next readings (L26/A to L26/P) inform the active or nom reseted fails. After the reset command, in case there is not any active fail, all the registers should have the value " 0 ".
Values for all the registers:
0: Don't active **1**: Active

L26/A: Load Undervoltge Flag.
%R20.1 1 Bit Read Only
L26/B: Load Overvoltage Flag.
%R20.2 1 Bit Read Only
L26/C: Under/Over Frequency Flag.
%R30.4 1 Bit Read Only
L26/D: Under Excitation Flag.
%R20.5 1 Bit Read Only
L26/E: Over excitation Flag.
%R20.6 1 Bit Read Only
L26/F: Over Lead Angle Flag.
%R20.7 1 Bit Read Only
L26/G: Over Lag Angle Flag.
%R20.8 1 Bit Read Only
L26/H: Line Undercurrent Flag.
%R20.9 1 Bit Read Only
L26/I: Line Overcurrent Flag.
%R20.10 1 Bit Read Only
L26/J: Line Underpower Flag.
%R20.11 1Bit Read Only
L26/K: Line Overpower Flag.
%R20.12 1Bit Read Only
L26/L: Motoring Flag.
%R20.13 1Bit Read Only
L26/M: Exciter Overtemperature.
%R20.14 1Bit Read Only
L26/N: External Fail Flag.
%R20.15 1Bit Read Only
L26/O: Inadvertent Energization.
%R20.16 1Bit Read Only
L26/P: Lost Control Flag.
%R30.1 1Bit Read Only
L26/Q: Field Overtemperature Flag.
%R30.2 1Bit Read Only

L26/Q: First Fail Ocurred.
Values:
0: None
1: Line Undervoltage
2: Line Overvoltage
3: Line Undefrequency
4: Line Overfrequency
5: Field Undercurrent

6: Field Overcurrent
7: Overlead
8: Overlag
9: Line Undercurrent
10: Line Overcurrent
11: Line Underpower
12: Line Overpower
13: Motoring / Inadv. Energization
14: Exciter Overtemperature
15: External Fail
16: Field Loss
%R1050 16 Bit Read Only

L27/A: Force Mode Status.
Values: 0: Mode Normal **1:**Mode Forced
%R155.7 1 Bit Read Only
L27/B: Force Setting Status.
Values: 0: Setting Normal **1:**Setting Forced (Keyboard)
%R155.8 1 Bit Read Only
L27/C: Droop Status.
0:Droop Inactive **1:**Droop Active
%R155.6 1 Bit Read Only
L27/D: Limiting By Lag Status.
Values: 0: No **1:**Limiting
%R155.1 1 Bit Read Only
L27/E: Limiting by Lead Status.
0:No **1:**Limiting
%R155.2 1 Bit Read Only
L26/F: Limiting by Overexcitation.
0:No **1:**Limiting
%R155.3 1 Bit Read Only
L27/G: Limiting by Underexcitation.
0:No **1:**Limiting
%R155.4 1 Bit Read Only
L27/H: Clamping at Minimum Status.
0:No **1:**Clamping
%R155.9 1 Bit Read Only
L27/I: Clamping at Maximun Status.
0:No **1:**Clamping
%R155.10 1 Bit Read Only

Mute Alarm Command via Modbus:
Force the input register once (temporarilly) to "1" or force K7for muting.
C01/A: Mute Alarm Command
Values: 0: No Mute **1**: Mute Alarm
%I.5 Immediate or
%R1.5 1Bit Immediate or
%K7 Immediate

Reset Fails Command via Modbus:
Force the input register twice (temporarilly) to "1" for reset (The first force will mute the alarm and second one will reset) or force K6.
C01/A: Mute Alarm Command
Values: 0: No Reset **1**: Reset Fails
%I.5 Immediate or
%R1.5 1Bit Immediate or
%K6 Immediate

NOTES

NOTES

Varixx Industria Eletrônica

Rua Phelipe Zaidan Maluf 1501 - Distrito Industrial Unileste
Piracicaba - SP - CEP13.422.190 - Phone: (55) (19) 3424.4000 - Fax: (55) (19) 3424.4001
www.varixx.com.br e-mail: info@varixx.com.br

- Excitatrizes Estáticas para Geradores.
- Excitatrizes Estáticas para Motores Síncronos.
- AVRs -Automatic Voltage Regulators.
- Soft Starters para Motores.
- Soft Starters de Média Tensão.
- Acionamento, excitação e proteção para média tensão.
- Relês de Proteção.
- Controladores de Potência.
- CESs - Contatores de Estado Sólido.
- Retificadores Controlados de alta corrente.
- Choppers para Motores.
- Transmissores e Transdutores.
- Control Box para Motores Síncronos.
- Crowbar para Motores Síncronos.
- Crowbar para Proteção contra Transiêntes.
- Equipamentos Especiais.