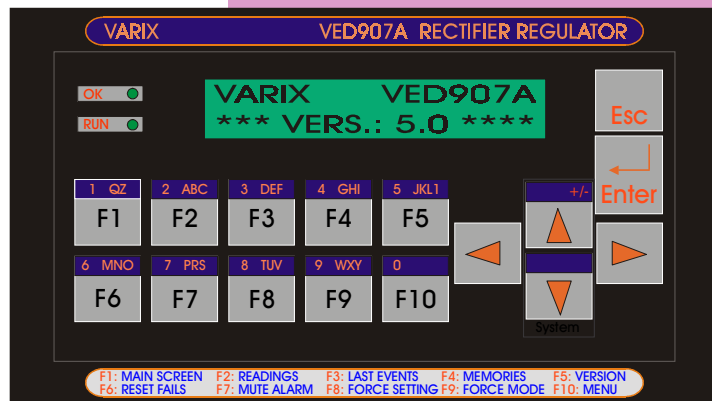


Bulletin

259AE

VED907 MODBUS CONTROL



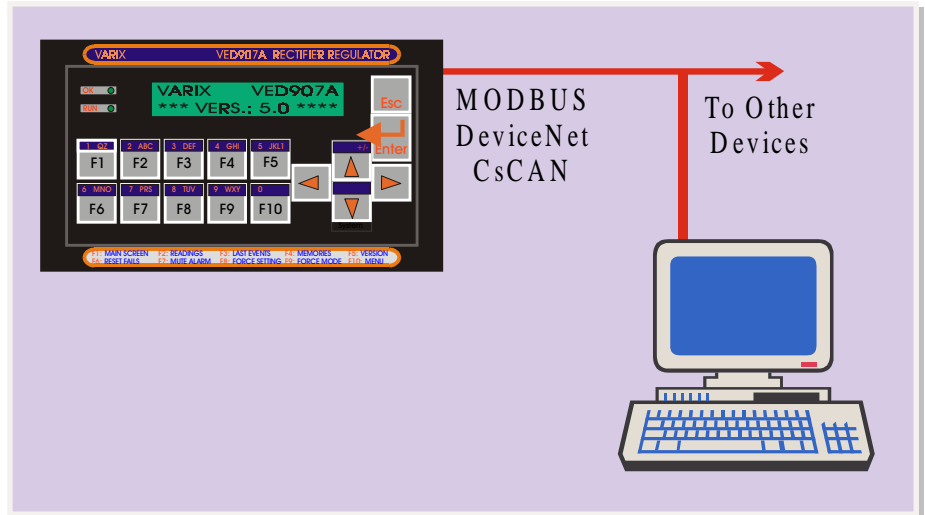
VED907A Rectifier Systems

Bulletin 259AE

RECTIVAR



VED907A + Transformer



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 ASCII printable 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.

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 equipment 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 that 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 VED907A 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 (Toggle).

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 VED907. 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 **lower 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 **second 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 VED907

The alteration of parameters, depending on the case, can be immediately effective after the alteration in the VED907 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 effective or "**Flagged**". if there is necessity of setting %R11.11 to be effective. Change all parameters you want first and then set %R11.11 to make the changes effective. The flag %R11.11 after being set at "1" is automatically reseted to "0" by the "firmware" after some mili-seconds.

Comm Port Buffering

The VED907A 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 VED907 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 VED907 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 need 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 its 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 VED907

The VED907 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 VED907 Signal CTS, available on the DB-9 connector, Pin 8. When the VED907 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 VED907 supports these control signals.

Network Connector (Optional) - This connector is MALE, and has the following pin assignments:

PinNumber	Function
1	Voltage -
2	CAN_L
3	Shield
4	CAN_H
5	Voltage +

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

PinNumber	Function	Direction	Relative to VED907	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)

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 (1100000000000101). 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	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
Polynommmial		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 VED907, 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 numbers are often called floating point. In VED907, 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:

.000000004567 = 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 VED907 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 known 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 VED907 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 00000000000000100 value (65540 decimal) is loaded into register %R43, %R43 contains 00000000000000100 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 VED907 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.

Some 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 / DATA START 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 or 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	DATA COILHO	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	DATA VALUE 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	DATA VALUE 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	00	00
DATA OUTPUTREGH.O.40110	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 flaws. The options for whole the flaws are: “None”, “Alarm”, “Inhibit”, “Trip”, “Both” and “Forces Open Loop.” **See Buletin 249A for more details.**

For each adress set as follow:

- 0 = None
- 1 = Alarm
- 2 = Inhibit
- 3 = Trip
- 4 = Both
- 5 = Forces Open Loop

1.01/A: Under Current Action. Load current under the programmed limit.

%R101 8 bit Immediate

1.01/B: Over Current Action. Load current over the programmed limit.

%R102 8 bit Immediate

1.02/A: Under Voltage Action. Load voltage under the programmed limit.

%R103 8bit Immediate

1.02/B: Over Voltage Action. Load voltage over the programmed limit.

%R104 8bit Immediate

1.03/A: Under Power Action. Load power under the programmed limit.

%R105 8 bit Immediate

1.03/B: Over Power Action. Load power over the programmed limit.

%R106 8 bit Immediate

1.04/A: Under Line Voltage Action. Line voltage under the programmed limit.

%R107 8 bit Immediate

1.04/B: Over Line Voltage Action. Line voltage over the programmed limit.

%R108 8 bit Immediate

1.05/A: Over Lag Action. Transformer power Factor too inductive, over the programmed limit.

%R109 8 bit Immediate

1.05/B: Over Lead Action. Transformer power Factor too Capacitive, over the programmed limit.

%R110 8 bit Immediate

1.06: Over Temperature Action. Over temperature on rectifier heater exchange.

%R111 8 bit Immediate

1.07: External Fail Action. Detects the closure of a normal open contact from a PLC or external protection relays to signalizes fails like open fuses, phase loss and many others.

%R113 8 bit Immediate

1.08: Inadvertent Inergization: The DRR detects if there is tension on load without enabling the rectifier.

%R114 8 bit Immediate

1.09: Lost of Control: DRR accomplishes an auto-check continually and it activates this fail if there is a discrepancy above 10% among the values of “Final Set Point” and “Process Value” in the input of the virtual error amplifier (PID).

%R115 8 bit Immediate

1.10/A: Under Line Current Action. Transformer primary current under the programmed limit.

%R116 8 bit Immediate

1.10/B: Over Line Current Action. Transformer primary current over the programmed limit.

%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: Constant Amper
- 2: Constant Voltage
- 4: Constante KW
- 8: Constant Voltage with Droop
- 16: Constante KW with Droop
- 32: Constante voltage with compound
- 64 : Constante KW with compound

Address:

%R123 8 bit Flagged

2.02A: Start Ramp.

Values: 0: Disable 1: Enable Immediate

2.02B: Stop Ramp.

Values: 0: Disable 1: Enable Immediate

%R152.15 1 bit Immediate

2.03: Output Parameter:

Values:

- 0: None
- 1: Setting Scale
- 2: Load Current
- 4: Line Voltage
- 8: Line Current
- 16: Power Factor
- 32: Line KVA
- 64: Line KVAR
- 128: Load Power
- 256: Set after Range
- 512: Set Final
- 1024: Control Value
- 2048: Load Voltage

Address:

%R124 16 bit Immediate

2.04: Up Down Start.

Values:

- 1: 0% of Range
- 2: 50% of Range
- 4: 100% of Range
- 8: Last Value
- 16: Nominal

Address:

%R125 8 bit Immediate

2.05: Polarization Flag (No, Yes).

Values 0: No 1: Yes

%R151.1 1 bit Immediate

2.06/A: Under Current Limit Flag.

Values: 0: No 1: Yes

%R151.3 8 bit Immediate

2.06/B: Over Current Limit Flag.

Values: 0: No 1: Yes

%R151.4 8 bit Immediate

2.07: Adjuste Range.

Values: 0: 0 to 100% 1: +/- 10% Immediate

%R151.8 8 bit Immediate

2.08: Operation Mode.

Values: 0: Automatic 1: Manual Flagged

%R128 8 bit Immediate

2.09A: Fast Loop.

Values: 0: Enable 1: Disable Immediate

%R152.12 1 bit Immediate

2.09B: Slow Loop.

Values: 0: Enable 1: Disable Immediate

%R152.7 1 bit Immediate

2.10: Time Program Flag.

Values: 0: No 1: Yes Immediate

%R152.1 1 bit Immediate

2.11: Mute Mode.

Values: 0: Manual 1: Automatic Immediate

%R152.9 1 bit Immediate

2.12: Reset Mode.

Values: 0: Manual 1: Automatic Immediate

%R152.10 1 bit Immediate

2.13: Setting Modes.**Values:**

- 0: Up/Down Only
- 1: Keyboard Only
- 2: Up/Down + Keyboard
- 3: Pot/0-5Volt Only
- 4: 0 to 20 mA Only
- 5: Pot/0-5V + 0 to 20 mA
- 6: Up/Down + 0 to 20 mA
- 7: Keyboard + 0 to 20 mA
- 8: Pot/0-5V + Up/Down
- 9: Pot/0-5V + Up/Down + 0 to 20 mA

Address:

%R703 8 bit Flagged

2.14: Power Factor Sensing Mode.

Values: 0: Internal 1: External
%R152.8 1 bit Flagged

2.15: Manual to Auto Transfer**Values:**

- 1: Mantain Process Value
- 2: Setting = 0%
- 4: Setting = 50%
- 8: Setting = 100%
- 16: Setting = Nominal

Address:

%R131 8 bit Flagged

03: Delays.

In this sub-item the delays for fail Detection of each flaw, 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. These times serve as filter for spurious conditions, avoiding detection of fails for normal process transients.

3.1/A: Load Under Current Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R178 16 bit Immediate

3.1/B: Load Over Current Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R179 16 bit Immediate

3.2/A: Load Under Voltage Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R180 16 bit Immediate

3.2/B: Load Over Voltage Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R181 16 bit Immediate

3.3/A: Under Power Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R182 16 bit Immediate

3.3/B: Over Power Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R183 16 bit Immediate

3.4/A: Line Under Voltage Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R184 16 bit Immediate

3.4/B: Line Over Voltage Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R185 16 bit Immediate

3.5/A: Over Lag Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R186 16 bit Immediate

3.5/B: Over Lead Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R187 16 bit Immediate

3.6: Over Temperature Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R188 16 bit Immediate

3.7: External Fail Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R190 16 bit Immediate

3.8: Inadvertent Energization Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R191 16 bit Immediate

3.9: Lost of Control Delay (0.1 to 1000 Seconds).

Values: 1 to 10000
%R192 16 bit Immediate

3.10: Line Under Current Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R193 16 bit Immediate

3.11: Line Over Current Delay (0,1 to 1000 Seconds).

Values: 1 to 10000
%R194 16 bit Immediate

3.12: Auto Mute Time (0.1 to 1000 Seconds).

Values: 1 to 10000
%R196 16 bit Immediate

3.13: Auto Reset Time (0.1 to 1000 Seconds).

Values: 1 to 10000
%R197 16 bit Immediate

3.14: Trip Pulse Time (0.1 to 1000 Seconds or infinite (infinite = 0)).

Values: 1 to 10000
%R199 16 bit Immediate

04: Parameters:

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: Under Current Trip.

Values: 0% to 80.00%
%R235 Float. Point-32Bit Flagged

04.1/B: Over Current Trip.

Values: 50% to 180.00%
%R237 Float. Point-32Bit Flagged

04.2/A: Under Voltage Trip.

Values: 0% to 80.00%
%R239 Float. Point-32Bit Flagged

04.2/B: Over Voltage Trip.

Values: 50% to 180.00%
%R241 Float. Point-32Bit Flagged

04.3/A: Under Power Trip.

Values: 0% to 80.00%
%R203 Float. Point-32Bit Flagged

04.3/B: Over Power Trip.

Values: 50% to 200.00%
%R207 Float. Point-32Bit Flagged

04.4/A: Under Line Voltage Trip.

Values: 50% to 100.00%
%R205 Float. Point-32Bit Flagged

04.4/B: Over Line Voltage Trip.

Values: 80% to 150.00%
%R209 Float. Point-32Bit Flagged

04.5/A: Over Lag Trip.

Values: 0.01 to 0.99
%R211 Float. Point-32Bit Immediate

04.5/B: Over Lead Trip.

Values: 0.01 to 0.99
%R215 Float. Point-32Bit Immediate

04.6/A: Under Line Current Trip.

Values: 10% to 80.00%
%R213 Float. Point-32Bit Flagged

04.6/B: Over Line Current Trip.

Values: 80% to 150.00%
%R217 Float. Point-32Bit Flagged

04.7/A: Under Current Limit.

Values: 10% to 80.00%
%R247 Float. Point-32Bit Flagged

04.7/B: Over Current Limit.

Values: 80% to 150.00%
%R249 Float. Point-32Bit Flagged

04.8/A: Start Ramp (.0001 a 30000 Units by seconds).

Values: 0 to 30000
%R225 Float. Point-32Bit Immediate

04.8/B: Stop Ramp (.0001 a 30000 Unidades por segundo).

Values: 0 to 30000
%R227 Float. Point-32Bit Immediate

04.9: Droop Rate (0% a 10%).

Values: 0% to 10%

%R219 Float. Point-32Bit Immediate

04.10: Compound Rate (10% a 90%).

Values: 10% to 90%

%R221 Float. Point-32Bit Immediate

04.11: Polarization Level (0.01% a 5%).

Values: 0.01% to 5.00%

%R223 Float. Point-32Bit Flagged

04.12: P.F. Polarity (Normal / Inverse).

Values: 0: Normal 1: Inverse

%R152.16 1 bit Flagged

05: PID Calibration.

In this sub-item are settled the parameters relative to the digital error amplifier with PID action (Proportional, Integral and Derivative).

05.01: Dead Band + .

Values: 0 to 1000

%R302 16Bit Immediate

05.02: Dead Band - .

Values: 0 to 1000

%R303 16Bit Immediate

05.03: Proportional Gain.

Values: 0 to 32767

%R304 16Bit Immediate

05.04: Derivative Gain.

Values: 0 to 32767

%R304 16Bit Immediate

05.05: Integral Rate - (0 a 32767 r/S).

Integral. Define a correção lenta, com tempo longo e tendendo a erro zero ou máximo de correção

05.06: Minimum Slew Time.

Values: 0 to 1000

%R310 16Bit Immediate

05.07: Derivative Term - (Error = PV-SP) ou PV = Process Value).

Values: 0: Error 1: PV (Process Value)

%R311.11 1Bit Immediate

Menu 06: Filter Calibration.

In this sub-item are settled the filter depth in four reading inputs.

05.01/A: Filter 1. Rectifier voltage.

Values: 0 to 100

%R361 16Bit Flagged

05.01/B: Filter 1. Line Voltage, Line Current, Power Factor and frequency.

Values: 0 to 100

%R363 16Bit Flagged

05.01/C: Filter 1. Rectifier Current.

Values: 0 to 100

%R365 16Bit Flagged

05.01/D: Filter 1. Setting 0 to 5VCC / Potentiometer.

Values: 0 to 100

%R367 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.

07.01A: Line Voltage Scale. (1 = No correction).

Values: -10.00 to 10.00

%R951 Float. Point-32Bit Immediate

07.01B: Line Voltage Zero. (0 = No correction).

Values: -99.99 to 99.99

%R971 Float. Point-32Bit Immediate

07.02A: Line Current Scale. (1 = No correction).

Values: -10.00 to 10.00

%R953 Float. Point-32Bit Immediate

07.02B: Line Current Zero. (0 = No correction).

Values: -99.99 to 99.99

%R973 Float. Point-32Bit Immediate

07.03A: Rectifier Current Scale. (1 = 0 correction).

Values: -10.00 to 10.00

%R955 Float. Point-32Bit Immediate

07.03B: Rectifier Current Zero. (0 = No correction).

Values: -99.99 to 99.99

%R975 Float. Point-32Bit Immediate

07.04A: Rectifier Current Scale. (1 = No correction).

Values: -10.00 to 10.00

%R961 Float. Point-32Bit Immediate

07.04B: Rectifier Current Zero. (0 = No correction).

Values: -99.99 to 99.99

%R977 Float. Point-32Bit Immediate

07.05: Setting Pot/0-5V Calibration (1 = No correction).

Values: -10.00 to 10.00

%R959 Float. Point-32Bit Immediate

07.06: Scale Shift Calibration(0 = No correction).

Values: -50.00% to 50.00%

%R957 Float. Point-32Bit Flagged

10: Set Nominal.

In this sub-item should be settled the nominal values of the system to be controlled.

10.01/A: Nominal Transformer KVA.

Values: 0.001 to 1000000.00

%R401 Float. Point-32Bit Flagged

10.01/B: Nominal Line Voltage.

Values: 100.00 to 50000.00

%R403 Float. Point-32Bit Flagged

10.01/C: Nominal Secondary Voltage.

Values: 10.00 to 9999.00

%R407 Float. Point-32Bit Flagged

10.02/A: Nominal Line Current (10 a 99999). Corrente nominal de linha ou primário do transformador de retificação.

Values: 10 to 99999

%R409 Float. Point-32Bit Flagged

10.02/B: Nominal Line Frequency.

Values: 50 to 400

%R419 Float. Point-32Bit Flagged

10.02/C: Nominal Power Factor.

Values: 0.10 to 1.00

%R421 Float. Point-32Bit Flagged

10.03/A: Line Current Transformer (A / 1A).

Values: 1 to 99999

%R411 Float. Point-32Bit Flagged

10.03/B: Line Voltage Transformer (V / 115 V).

Values: 1 to 50000

%R405 Float. Point-32Bit Flagged

10.04/A: Maximum Rectifier Current.

Values: 1.0 to 999999.0

%R415 Float. Point-32Bit Flagged

10.04/B: Transducer Rate (A / 5V).

Values: 1.0 to 999999.0

%R417 Float. Point-32Bit Flagged

10.05/A: Maximum VCC. (Vsec x 1,3).

Values: 1.0 to 10000.0

%R425 Float. Point-32Bit Flagged

10.05/B: Transducer Rate (VCC/10V).

Values: 1.0 to 10000.0

%R423 Float. Point-32Bit Flagged

10.06/A: Nominal VCC (1 a 10000 V).

Values: 1.0 to 10000.0

%R427 Float. Point-32Bit Flagged

10.06/B: Nominal ACC.

Values: 1.0 to 999999.0

%R429 Float. Point-32Bit Flagged

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 nothing 1: Start

%T120 1Bit Immediate

14: Set Time Program.

in this sub-item can be settled the segments timing and rates for time program.

14.01/A: Seg 1 Hour

Values: 0 to 24

%R1311 16 Bit Immediate

14.01/B: Seg 1 Minute

Values: 0 to 59

%R1312 16 Bit Immediate

14.01/C: Seg 1 Setting

Values: 0 to 100.0%

%R1313 Float.Point-32bit Immediate

14.01/D: Seg 1 Rate

Values: 0 to 100.00%

%R1315 Float.Point-32bit Immediate

14.02/A: Seg 2 Hour

Values: 0 to 24

%R1321 16 Bit Immediate

14.02/B: Seg 2 Minute

Values: 0 to 59

%R1322 16 Bit Immediate

14.02/C: Seg 2 Setting

Values: 0 to 100.0%

%R1323 Float.Point-32bit Immediate

14.02/D: Seg 1 Rate

Values: 0 to 100.00%

%R1325 Float.Point-32bit Immediate

14.03/A: Seg 3 Hour

Values: 0 to 24

%R1331 16 Bit Immediate

14.03/B: Seg 3 Minute

Values: 0 to 59

%R1332 16 Bit Immediate

14.03/C: Seg 3 Setting

Values: 0 to 100.0%

%R1333 Float.Point-32bit Immediate

14.03/D: Seg 3 Rate

Values: 0 to 100.00%

%R1335 Float.Point-32bit Immediate

14.04/A: Seg 4 Hour

Values: 0 to 24

%R1341 16 Bit Immediate

14.04/B: Seg 4 Minute

Values: 0 to 59

%R1342 16 Bit Immediate

14.04/C: Seg 4 Setting

Values: 0 to 100.0%

%R1343 Float.Point-32bit Immediate

14.04/D: Seg 4 Rate

Values: 0 to 100.00%

%R1345 Float.Point-32bit Immediate

14.05/A: Seg 5 Hour

Values: 0 to 24

%R1351 16 Bit Immediate

14.05/B: Seg 5 Minute

Values: 0 to 59

%R1352 16 Bit Immediate

14.05/C: Seg 5 Setting

Values: 0 to 100.0%

%R1353 Float.Point-32bit Immediate

14.05/D: Seg 5 Rate

Values: 0 to 100.00%

%R1355 Float.Point-32bit Immediate

14.06/A: Seg 6 Hour

Values: 0 to 24

%R1361 16 Bit Immediate

14.06/B: Seg 6 Minute

Values: 0 to 59

%R1362 16 Bit Immediate

14.06/C: Seg 6 Setting

Values: 0 to 100.0%

%R1363 Float.Point-32bit Immediate

14.06/D: Seg 6 Rate

Values: 0 to 100.00%

%R1365 Float.Point-32bit Immediate

14.07/A: Seg 7 Hour

Values: 0 to 24

%R1371 16 Bit Immediate

14.07/B: Seg 7 Minute

Values: 0 to 59

%R1372 16 Bit Immediate

14.07/C: Seg 7 Setting

Values: 0 to 100.0%

%R1373 Float.Point-32bit Immediate

14.07/D: Seg 7 Rate

Values: 0 to 100.00%

%R1375 Float.Point-32bit Immediate

14.08/A: Seg 8 Hour

Values: 0 to 24

%R1381 16 Bit Immediate

14.08/B: Seg 8 Minute

Values: 0 to 59

%R1382 16 Bit Immediate

14.08/C: Seg 8 Setting

Values: 0 to 100.0%

%R1383 Float.Point-32bit Immediate

14.08/D: Seg 8 Rate

Values: 0 to 100.00%

%R1385 Float.Point-32bit Immediate

15: Set Clock.**15.01/A: New Year**

Values: 2000 to 2100

%R0655 16 bit Immediate

15.01/B: New Month

Values: 1 to 12

%R0654 16 bit Immediate

15.01/C: New Day

Values: 1 to 31

%R0653 16 bit Immediate

15.01/D: New Hour

Values: 1 to 24

%R0652 16 bit Immediate

15.01/E: 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.

F02: PID Bias.

Values: -32000 to 32000

%R307 16 Bit Immediate

F03/A: Upper Clamp mode.

Values: 0: Fixed 1: Automatic

%R152.13 16 Bit Immediate

F03/B: Upper Clamp Level.

Values: 0 to 32000

%R308 16 Bit Immediate

F04/A: Lower Clamp mode.

Values: 0: Fixed 1: Automatic

%R152.14 16 Bit Immediate

F04/B: Lower Clamp Level.

Values: 0 to 32000

%R309 16 Bit Immediate

F09: Analog Output 2 Calibration (1 = no change).

Values: 0 to 10.00

%R911 Float.Point-32 Bit Flagged

F14: Frequency Reading Calibration (1 = no change).

Values: 0 to 10.00

%R931 Float.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 VED907 for eventual reset of the events memorized.

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 Start Hour.

%R1002 16 Bit Read Only

L02/B: Last Start Minute.

%R1003 16 Bit Read Only

L02/C: Last Start Day.

%R1004 16 Bit Read Only

L02/D: Last Start Month.

%R1005 16 Bit Read Only

L02/E: Last Start Year.

%R1006 16 Bit Read Only

L03/A: Last Stop Hour.

%R1007 16 Bit Read Only

L03/B: Last Stop Minute.

%R1008 16 Bit Read Only

L03/C: Last Stop Day.
 %R1009 16 Bit Read Only
L03/D: Last Stop Month.
 %R1010 16 Bit Read Only
L03/E: Last Stop Year.
 %R1011 16 Bit Read Only

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

L05/A: Total Operating Hour.
 %R1025 16 Bit Read Only
L05/B: Total Operating 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: Event:

- 0: None
- 1: Undercurrent
- 2: Overcurrent
- 3: Undervoltage
- 4: Overvoltage
- 5: Load Underpower
- 6: Load Overpower
- 7: Line Undervoltage
- 8: Line Overvoltage
- 9: Line Overlapping
- 10: Line Overleading
- 11: Line Undercurrent
- 12: Rectifier Overtemperature
- 13: External Fail
- 14: Inadvertent Energization
- 15: Lost of Control
- 16: Line Overcurrent
- 17: Not Used
- 18: Normal Enable
- 19: Normal Disable

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: Load Current.

%R827 Float.Poin -32 Bit Read Only

L19/B: Load Voltage.

%R821 Float.Poin -32 Bit Read Only

L20/A: Line Current.

%R823 Float.Poin -32 Bit Read Only

L20/B: Line Voltage.

%R825 Float.Poin -32 Bit Read Only

L21/A: Line KVA.

%R833 Float.Poin -32 Bit Read Only

L21/B: Line KVAR.

%R835 Float.Poin -32 Bit Read Only

L22/A: Line P.F. Angle.

%R153.5 1 bit Read Only

Values: 0: Lag 1: Lead

L22/A: Line Power Factor.

%R829 Float.Poin -32 Bit Read Only

L22/B: Load KW.

%R839 Float.Poin -32 Bit Read Only

L23: Line Frequency.

%R831 Float.Poin -32 Bit Read Only

L24/A: Actual State.

Values: 1:Inhib. 2:Polarized 3:Operating.

%R701 16 Bit Read Only

L24/B: Ramp State.

Values: 1:Inhib. 2:Polarized 3:Operating.

%R132 16 Bit Read Only

L24/C: Limit Status.Values: 0:No. 1:Limiting Upper Current
2:Limiting Lower Current.

%R157 8 Bit Read Only

L24/D: Final Setting (0 to 100%).

%R709 Float.Point - 32 Bit Read Only

L24/E: Set Value.

%R705 Float.Point - 32 Bit Read Only

L23/F: Set Type.

Values: 1:Amper 2: Voltage 3: KW

%R133 16 Bit Read Only

L25/A: Operating Mode.

Values: 0:Automatic 1:Man_Open_Loop

%R702 8 Bit Read Only

L25/B: Protection Status.

Values: 1:Normal 2:Alarm

3:Failed 4:Muted

%R129 8 Bit Read Only

L25/C: Souce After Force.

Values:

0: Up/Down Only

1: Keyboard Only

2: Up/Down + Keyboard

3: Pot/0-5Volt Only

4: 0 to 20 mA Only

5: Pot/0-5V + 0 to 20 mA

6: Up/Down + 0 to 20 mA

7: Keyboard + 0 to 20 mA

8: Pot/0-5V + Up/Down

9: Pot/0-5V + Up/Down + 0 to 20 mA

%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 activate 1: Activate

L26/A: Load Undercurrent Flag.

%R20.1 1 Bit Read Only

L26/B: Load Overcurrent Flag.

%R20.2 1 Bit Read Only

L26/C: Load Undervoltage Flag.

%R20.3 1 Bit Read Only

L26/D: Load Overvoltage Flag.

%R20.4 1 Bit Read Only

L26/E: Load Underpower Flag.

%R20.5 1 Bit Read Only

L26/F: Load Overpower Flag.

%R20.6 1 Bit Read Only

L26/G: Line Undervoltage Flag.

%R20.7 1 Bit Read Only

L26/H: Line Overvoltage Flag.

%R20.8 1 Bit Read Only

L26/I: Line Overlag Flag.

%R20.9 1 Bit Read Only

L26/J: Line Overlead Flag.

%R20.10 1Bit Read Only

L26/K: Line Undercurrent Flag.

%R20.11 1Bit Read Only

L26/L: Rectifier Overtemperat. Flag.

%R20.12 1Bit Read Only

L26/M: External Fail Flag.

%R20.13 1Bit Read Only

L26/N: Inadvertent Energization Flag.

%R20.14 1Bit Read Only

L26/O: Lost Control Flag.

%R20.15 1Bit Read Only

L26/P: Line Overcurent Flag.

%R20.16 1Bit Read Only

L26/Q: First Fail Occured.

Values:

0: None

1: Undercurrent

2: Overcurrent

3: Undervoltage

4: Overvoltage

5: Load Underpower

6: Load Overpower

7: Line Undervoltage

8: Line Overvoltage

9: Line Overlag

10: Line Overlead

11: Line Undercurrent

12: Rectifier Overtemperature

13: External Fail

14: Inadvertent Energization

15: Lost of Control

16: Line Overcurrent

%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 For-
ced (Keyboard)

%R155.8 1 Bit Read Only

L27/C: Droop Status.Values: 0:Droop Inactive 1:Droop Acti-
ve

%R155.6 1 Bit Read Only

L27/D: Compound Status.Values: 0:Compound Inactive 1:Com-
pound Active

%R155.4 1 Bit Read Only

L27/E: Limiting by Clamping.

Values: 0:Not Clamping 1:Clamping

%R155.5 1 Bit Read Only

L26/F: Limiting by Undercurrent.

Values: 0:Not Limiting 1:Limiting

%R155.2 1 Bit Read Only

L27/G: Limiting by Overcurrent.

Values: 0:Not Limiting 1:Limiting

%R155.1 1 Bit Read Only

Mute Alarm Command via Modbus:Force the input register once (temporari-
ly) to "1" for muting or force K7.**C01/A: Mute Alarm Command**Values: 0: No Mute 1: Mute Alarm
Immediate

%I.5

or

%R1.5 1Bit

or

%K7

Reset Fails Command via Modbus:Force the input register twice (temporari-
ly) to "1" for reset (The first force will
mute the alarm and second one will reset)
or force K6. To reset command to be efec-
tive any fail should be active in the mo-
ment.**C01/A: Mute Alarm Command**Values: 0: No Reset 1: Reset Fails
Immediate

%I.5

or

%R1.5 1Bit

or

%K6

Varix Brazil:

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

Varix Electronics USA:

10001 NW 50 Th Street - Bldg. 102-A
Fort Lauderdale - Florida - 33351
Tool Free: 1-800-238 6696 - Phone: (954) 572 5535 - Fax: (954) 572 0331



- STATICS EXCITERS FOR GENERATORS.
- STATICS EXCITERS FOR SYNCHRONOUS MOTORS.
 - AVRS - AUTOMATIC VOLTAGE REGULATORS.
 - SOFT STARTERS FOR MOTORS.
 - POWER CONTROLLERS.
 - SSCs - SOLID STATE CONTACTORS.
 - CONTROLLED RECTIFIERS UP TO 150.000 A.
 - CHOPPERS FOR MOTORS.
 - PROTECTION RELAYS FOR EXCITATION SYSTEMS.
- DIGITAL RELAYS FOR CIRCUIT BREAKERS SUPERVISION.
 - TRANSMITTERS AND TRANSDUCERS.
- SYNCHRONISM CONTROL BOX FOR SYNCHRONOUS MOTORS.
 - CROWBAR FOR SYNCHRONOUS MOTORS.
 - CROWBAR FOR TRANSIENT PROTECTION.
 - SPECIAL EQUIPMENTS.