Technical Brief
AN260 Rev A4

Remote Autonomous Zone Nodes (RAZN)
MODBUS Technology and Registers

By John Sonnenberg *Raveon Technologies Corp*

There are many different versions of the RAZN. This Application note shows how MODBUS works with the Remote Autonomous Zone Nodes (RAZN) and this shows the list of Registers SCADA communciations can access.

Overview

The Remote Autonomous Zone Note (**RAZN**) is setup in a plastic enclosure, with communication ports and DC power input on the front of it. IO Terminals are on the upper side of the RAZN enclosure. Most use one or two 12-pin IO terminals. If there is a radio modem inside the RAZN, the antenna connector is on the back side.

In the Remote Autonomous Zone Note (**RAZN**), there is a CPU board that connects to serial ports, data radios, and an upper IO board. The CPU board does the RTU functions, and Autonomous features.

Communication Methods to and from a RAZN

Ethernet Connect an Ethernet Cable or Wi-Fi adaptor to the RAZN's 10/100mbps Ethernet connector. This is a *Terminal Server* with 1-3 TCP/IP ports for 1-3 simultaneous client connections.

RS-485 Serial Connect an RS-485 serial cable to a RAZN or dozens of RAZNs that share this differential serial communications port connection.

Narrow Band RF The RAZN can have Raveon's RV-M6, or RV-M8 data radio modem installed inside for ultra long range RF data 1-50 miles in VHF or UHF RF bands.

LoRa RF The RAZN can have Raveon's RV-M50 LoRa data radio modem installed inside for long range license-free RF data 1-10 miles.

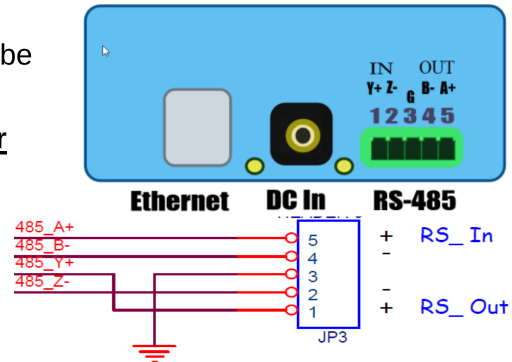
RS-232 Serial Connect an RS232 serial cable to a RAZN. This is an optional feature on all versions of the RV-N55 RAZN.

Wi-Fi The Ethernet connection can connect to a Wi-Fi modem to use Wi-Fi.

Raveon is always interested in adding features and options our customers need, and we are willing to adding any "linearizing software" for the thermocouple. And if you would like some additional Thermistor sensor to factor temperatures to your thermocouple, please contact Raveon customer support.

Or if you'd like the RAZN to wirelessly mimic the voltage from the thermocouple over a long distance, please contact Raveon customer support because this high accuracy 24bit ADC can wirelessly send voltage information to wherever you want it.

Communication Default Interface Settings



The **RAZN** has front panel interfaces are Ethernet and **RS-485**.
DC input power of 12V is on the front panel, **DC In**. Commands can be used to set/change baud rates, IP address, and port numbers.

RS-485 Serial Port (S1) **{SCADA RTU and User Interface for commands}**

Front panel RS-485 defaults to 9600 baud. Data in gets evaluated as a MODBUS message to this RAZN.

You can get into the *Command Mode* using a terminal emulator by typing +++, or sending in, +++ Enter commands you want to read or execute, and after 60 seconds of no commands, it will auto-exits the command mode.

E1 Ethernet TCP/IP Terminal Port1 (502). **{SCADA Communications}**

MODBUS TCP messages are processed, all the time, by this RAZN being communicated by this TCP/IP port. Responses will also come back out this port. The 10.10.10.209 IP address and Terminal port numbers can be configured in Command mode on the RS-485 interface.

E2 Ethernet TCP/IP Terminal Port2 (503). **{Communicate with the Internal Radio}**

On this Port2, all data is sent/transmitted over the air on the internal data radio inside the RAZN. When the radio receives data over the air, it comes out this terminal Port2. Use this to communicate wirelessly to remote devices, or just monitor wireless communications.

You can also enter +++ to go into the command mode and run commands on Terminal Port2.

To read your IP address info from the RS-485 port commands, enter **SHOW** to see all. IPADDR is the IP address command to set or read. **IPGATEWAY**, **IPMASK**, and **TCPP** are the main Ethernet confirmation commands.

SCADA Commands in the RAZN

The following commands in the RAZN are powerful SCADA and telemetry features that can be executed by issuing manual commands in the radio's command mode. They also can be executed using **WMX** messages local or remote.

Primary Commands to Setup your SCADA communication interfaces:

Command	Commands Description	Parameters	Default
SCDAPRO	**SCADAPRO x** Set the SCADA protocol to process data as. 0=none, 1=Modbus RTU, 2=Modbus TCP. 4=DNP3 The protocol flagged as used in SCDAPRO variable is the one used within the RAZN.	0 = NONE 1 = RTU 2 = TCP 4 = DNP3	2 (MODBUS TCP)
SLAVE	**SLAVE xxx** Sets Slave ID Address, **xxx** in decimal format. 1-255 are valid SCADA slave addresses. Used as Unit Identifier in MODBUS RTU and TCP protocols.	Xxx: 1-255 0=ignore	4
IOPIN	**IOPIN XX M** Set the GPIO bits state on the GPIO interface panel to inputs or outputs. XX parameter are the hexadecimal representation of the pins being configured. M is the mode for the XX pins. Mode M values: **A**: Digital TTL Input, **B**: Digital TTL Output. **C**: Open Drain MOSFET output, **D**: Power switched output. **E**:Voltage Input **F**:4-20mA Current Input **P**: Dual Pin Input	XX=Hex 00-FF M=(A,B,C,D,E)	

	S: Switch Input with pull-up res. Only use this to configure pins that are GPIO configurable.		
DEFOUT	**DEFOUT xx** Set the default output bit state used for power-up and re-booting. Xx is a hexadecimal variable, for all bits. 0 is the factory default unless specified in a particular product.	Xx: 0-FF	0
BCADD	**BCADD xxx** Sets additional broadcast Slave Address, **xxx** in decimal format. 1-255 are valid additional SCADA listen addresses to listen for a broadcast message. 0 is also a broadcast address, independent of this BCADD additional address.	Xxx: 1-255	

Ethernet Configuration Commands

The following commands are used to configure the Ethernet interface.

Command	Command Description	Parameters	Factory Default
IPADDR[2,3]	**Internet Protocol Address.** Get or set the unit IP Address in dotted quad decimal[1] notation.	<aaa.bbb.ccc.ddd>	10.10.10.209
IPMASK[2,3]	**Internet Protocol Address Mask.** Get or set the unit IP Address Mask dotted quad decimal[1] notation.	<aaa.bbb.ccc.ddd>	255.255.255.0
IPGATEWAY	**Internet Protocol Gateway Address.** Get or set the unit IP Gateway Address in dotted quad decimal[1] notation.	<aaa.bbb.ccc.ddd>	192.168.26.1
TCPP	**Gets/Sets the TCP server port number** to use for telnet server sessions with a client that connects. **TCPP x y** where **x** equals the telnet session and y equals the server port number. Max number of TCP/IP connections is 3.	1-65525	1 502 2 503 3 504

RAZN's MODBUS SCADA Commands

Modbus is a registered trademark of MODICON, Inc.

MODBUS Protocol Information

Big-Endian

Modbus is specified as big-endian, which means the most significant value is at the lowest address. With a read of a 16-bit (single register) value, the 1st byte returned is the MSB (most significant byte) and the 2nd byte returned is the LSB (least significant byte).

MODBUS TCP Commands

Modbus is a registered trademark of MODICON, Inc.

Messages sent from devices to the RAZN via the Ethernet port utilize SCADA and Telemetry protocols such as MODBUS TCP are often called "Telegrams". If an external Ethernet interface is used that converts the Ethernet message to RS232 or RS485, the serial ports on the RAZN can be configured to accept MODBUS TCP commands also. The SCDAPRO command sets the protocol to use in this product.

Default TCP/IP terminal port configuration is:

Port1 (502) is primarily for SCADA Communications. Port1 (503) is for communication to the data radio modem in the RAZN.

MODBUS TCP Functions used in the RAZN Products

The following MODBUS functions are incorporated into the RAZN.

Function Code (Decimal)	**Action**	**Traditional Description**	**Additional Information**
01	Read	Read Coil status of Outputs	IO0 through IO15 are register numbers 1-16 (0x001 – 0x0010).
02	Read	Read Input status of digital inputs	IO0 through IO15 are register numbers 1-16 (0x001 – 0x0010).
03	Read	Read Holding registers	03 can read registers that are used to hold configuration information, set points, and various parameters.
04	Read	Read Input Registers	Read the binary contents of input registers. Refer to specifications of the slave device for address and content.
05	Write	Force single coil	Broadcast and Group Addresses can be used.
06	Write	Preset Single Register	Presets a value into a single register. Some can be written, some cannot.
15	Write	Force Multiple Coils	Write to set the state of digital outputs. 1-16 bits can be forced with this commands.
16	Write	Preset multiple registers	Presets values into registers. Broadcast and Group Addresses not usable.

MODBUS TCP Message Format

MODBUS TCP is protocol designed to communicate over network connections that use the TCP protocol. The structure of a MODBUS TCP/IP message sent to a device is:

Byte 0:	transaction identifier – copied by server – usually 0
Byte 1:	transaction identifier – copied by server – usually 0
Byte 2:	protocol identifier = 0
Byte 3:	protocol identifier = 0
Byte 4:	length field (upper byte) = 0 (since all messages are smaller than 256) Number of bytes starting at byte 6
Byte 5:	length field (lower byte) = number of bytes following
Byte 6:	PLC unit identifier (previously ‘slave address’)
Byte 7:	MODBUS function code
Byte 8 and others:	Data as needed. On requests, This is the address in the Modbus map.

The transaction identifier is for transaction pairing; the MODBUS server copies the request transaction identifier in the response. Slave device must return the transaction identifiers received from the master.

MODBUS TCP Examples With Different Function Codes

Here is a list of many Modbus messages and responses in TCP Ethernet interface.

Example: (Coil status of Outputs) FC:1

Using function code 1 to read output status. The status of outputs 0-3 is shown as the byte value 0F hex, or binary 0000 1111.

The example request data is shown here in hex. 0005h is the Transaction code. 0006h is the message length. 14h is the Slave ID 20, 01h is the function, 0001h is IO# 0. 0001h is 1 which is the number of registers to read.

MODBUS TCP *Request*
08 08 00 00 00 06 14 01 00 01 00 04

Response to Request Example.
08 08 00 00 00 04 14 01 02 00 0F   1-2 bytes are returned, and the bits in them are the status

To request all 16 input bit status, send: 08 08 00 00 00 06 14 01 00 01 00 10
Response to Request Example.   08 08 00 00 00 05 14 01 02 30 00   Bits 10 and 11 were set.

Example: (Read Input Status)   FunctionCode:2

Using function code 2 to read input status in binary format. The status of outputs 0-3 is shown as a byte value of hex 0F, or binary 00001111.
The example request data is shown here in hex. 0808h is the Transaction code. 0006h is the message length. 14h is the Slave ID 20, 01h is the function, 0009h is IO# 8. 0008h is the number of registers to read: IO#8-15.

MODBUS TCP *Request*
08 08 00 00 00 06 14 01 00 09 00 08

Response to Request Example.
08 08 00 00 00 04 14 01 01 0F   1 bytes are returned, and the 8 are the status of IO# 8-15

Example: (Read Input Voltage)   FC:3

Using function code 3 (read holding Register) send the TCP command to the RAZN you want to read a register. The voltage register base number is 6400, so to read **IO#** 10, the register to read is 6410. It is a floating point 4 byte numeric response.

The example request data is shown here in hex. 0005h is the Transaction code. 0006h is the message length. 16h is the Slave ID 22, 03h is the function, 190Ah is register 6510. 0001h is 1 which is the number of registers to read.

MODBUS TCP *Request*
00 05 00 00 00 06 16 03 19 0A 00 01

Response to Request Example. 4 bytes, 4099C28Fh is 4.46V integer.
00 05 00 00 00 07 16 03 04 40 99 C2 8F

Example: (Read Output Status) FC:3

Using function code 3 (read Register 1001) All 16 bits in Reg. 1001 show status of IO# 0-15.

The example is shown in hex.   0005h is the Transaction code. 0006h is the message length. 04h is the Slave ID, 03E9h is register 1001. 0001h is 1the # of registers to read.

MODBUS TCP *Request*
00 05 00 00 00 06 04 03 03 E9 00 01

Response to Request.
00 05 00 00 00 07 04 03 04 03 E9 00 70   IO# bits set are: 4, 5, and 6   (70 hex)

Example: (Read INPUT Registers) FC:4

The same structure as function 03. The function code and accessible data are different. Using function code 4 to read 2 input status The message specifies the coil reference to be written. Coils are addressed starting at zero-coil 1 is addressed as 0. IO#0:Coil1, IO#1:Coil2, IO#2:Coil3, IO#3:Coil4, IO#4:Coil5, IO#5:Coil6, IO#6:Coil7….

The example is shown in hex. 0003h is Transaction code. 0006h is data length. 14h is the Slave ID 20, 00 05h is Input number for terminal IO# 4. 0001 is asking for One input register.

MODBUS TCP *Write Single Coil*

00 03 00 00 00 06 14 04 00 05 00 01

Response Format.

0003 0000 0006 14 04 02 FF00 the FF00 shows that IO#5 is ON. 0000 would show it off.

Example: (Force Coil) FC:5

Using function code 5 to (Set IO#5 ON). The message specifies the coil reference to be written. Coils are addressed starting at zero-coil 1 is addressed as 0. IO#0:Coil1, IO#1:Coil2, IO#2:Coil3, IO#3:Coil4, IO#4:Coil5, IO#5:Coil6, IO#6:Coil7….

The example is shown in hex. 0005h is Transaction code. 0006h is data length. 04h is the Slave ID, 00 05h is coil number for terminal IO# 4. FF00h specifies Turn it ON. 0000 would run it off.

MODBUS TCP *Write Single Coil*

00 03 00 00 00 06 04 05 00 05 FF 00

Response Format.

0003 0000 0006 04 05 0005 FF00 the FF00 shows that IO#5 is ON.

Error Response Example

0003 0000 0003 04 81 01 Response if an unavailable function was requested.

MODBUS TCP Error Codes:

Using exception codes in error messages.

01	ILLEGAL FUNCTION	The illegal function error is reported back by a Modbus server when either it does not support the function at all, or does not support that function code on the requested registers.
02	ILLEGAL DATA ADDRESS	The illegal data address error occurs if the server rejects the combination of starting register and length used. One possibility, is a mistake in your program on the starting register number.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave. This indicates a fault in the structure of remainder of a complex request, such as that the implied length is incorrect.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave attempted to perform the requested action.
11	COM Port Error	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Example: (Preset Multiple Registers) FC:16 Pulse a Solenoid IO Terminal

Using function code 16 (write 2 Registers 2700 and 2701) All 16 bits in Registers. This example shows how to talk to 2 registers that will specify which IO pins to pulse, and how many mS the pulse duration should be. This MOSBUS command can pulse a Solenoid.

The example is shown in hex.   0005h is the Transaction code. 0006h is the message length. 2Ch is the Slave ID 44. Register 27000 Is A8Ch. 0002h is 2 the # of registers to preset. IO# 4 and IO# 6 in bit mask register 2700 are 0050h to pulse. The 450mS pulse duration is 01C2h.

MODBUS TCP *Preset Multiple Registers*
00 05 00 00 00 06 2C 10 0A 8C 00 02 00 50 01 C2

Response format is the same as sent.
00 05 00 00 00 06 2C 10 0A 8C 00 02 00 50 01 C2

Communication Rates

Communication rates are very different for the different types of communication protocols and communication terminals.

Hire are estimated communication rates for SCADA communications using MODBUS communications.

When a Ethernet TCP/IP connection is used to communicate to the RAZN, here is the estimated time in mS that the RAZN gets and processes the data, and returns the response.

Ethernet TCP/IP Data Communications				
		Bytes Sent	Time (mS) RAZN gets data and processes	Time (mS) Response data gets back
		12	**6**	**8**
		18	**6**	**9**

When an RS-485 serial port connection is used to communicate to the RAZN, here is the estimated time in mS that the RAZN gets and processes the data, and returns the response.
9600 baud is the default baud rate in the RAZN RS-485

RS-485 Data Communications				
	RS-485 Baud	Bytes Sent	Time (mS) RAZN gets data and processes	Time (mS) Response data gets back
	2400	12	**44**	**84**
	2400	18	**64**	**124**
	9600	12	**14**	**24**
	9600	18	**19**	**34**
	19200	12	**9**	**14**
	19200	18	**12**	**19**

When an RAZN uses a data radio to communicate wirelessly, here is the estimated time in mS that the RAZN gets and processes the data, and returns the response.

RF Data Modem Radio Communications				
	Over the Air (OTA) Baud	Bytes Sent	Time (mS) RAZN gets data and processes	Time (mS) Response data gets back
	4800	12	**56**	**108**

	4800	18	**68**	**133**
	9600	12	**36**	**68**
	9600	18	**43**	**83**
	12800	12	**31**	**58**
	12800	18	**37**	**70**

MODBUS RTU Commands

Modbus is a registered trademark of MODICON, Inc.

Messages sent from devices that utilize SCADA and Telemetry protocols such as MODBUS are often called "Telegrams".

RTU Command Header

RTU commands have a header with 4 bytes containing the following variables.

Byte 0:	Slave Address. The ID of the RTU/SLAVE device
Byte 1:	Function Code for the message. Industry standard codes are used.
Byte 2 and 3:	Starting Register Address, 2 byte MSB and LSB. Start register is a 0 offset from the default register value assigned to each Function.
Additional bytes	Additional message varies by the type of function code and number of registers accessed.

MODBUS RTU Functions in the RAZN

The following MODBUS RTU Functions are incorporated into the RAZN.

Function Code (Decimal)	**Action**	**Traditional Description**	**Additional Information**
01	Read	Read Coil status	IO0 through IO15 are register numbers 1-16 (0x001 – 0x0010).
02	Read	Read Input status	Similar function as 01, except returns efficient data in bits instead of a 2 byte response.
03	Read	Holding registers	03 can read registers that are used to hold configuration information, set points, and various parameters
05	Write	Force single coil	Broadcast and Group Addresses can be used.
06	Write	Preset Single Register	Presets a value into a single register. Some can be written, some cannot.
08	R/W	Loop Back Message	Provides a series of tests for checking communication with client. Client echo's the message right back. Broadcast and Group Addresses not usable.
16	Write	Preset multiple registers	Presets values into registers.
81	Error	Error Happened	The next byte has the Error code in it if the response notifies an Error.

MODBUS RTU Examples With Defferent Funtion Codes

(02) Read Discrete Inputs (Input Bit status loaded into Bytes).

To read the Input Status of input bits, a command uses function code 02 to send the read request to the remote device. This function 02 s used to read contiguous status of discrete inputs in a remote device. The request specifies the starting address (the address of the first input specified), and the number of inputs to report.

So for example, the 02 function telegrams to read 3 coils IO0-IO2 on device 150(0x96) and receive the information is shown here in hexadecimal data format:

Byte #	Query Field Name	Example
Byte 0	Slave Address. The ID of the RTU device	96
Byte 1	Function Code for this message.	02
Byte 2	Starting Register Address MSB	00
Byte 3	Starting Register Address LSB	01
Byte 4	Number of coils MSB	00
Byte 5	Number of coils LSB	03
Byte 6	*Error Check Low Byte*	75
Byte 7	*Error Check High Byte*	2C

If the returned input quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros.

All bits in a remote device are assigned an address, and all a sequential addresses. The response will be 1 or more bytes. Bit 0 of the first response byte is the bit status of the specified address. Bit 1 is the second bit…

The structure of the response is:

Byte 0:	Slave Address. The ID of the RTU device
Byte 1:	Function Code for this message. 02
Byte 2:	Byte Count (N)
N Bytes:	The response Data bytes containing the requested input bit status.
2 CRC Bytes	*Error Check*

So for example, the telegram to read 3 bits, IO0-IO2 on device 150(0x96) is:

Request: 96 02 00 01 00 03 75 2C 68 CA
Response: 96 02 01 05 4D FF (it reported coil 1 and 3 are on) 0x05 = 00000101

Each status bits requested is contained in one bit of the data field. The least significant bit of the first data byte contains the status of the starting addressed status bit. Each successive status bit corresponds to the next significant bit in the data field. The unused bits in the last data byte are set to logical zeros.

(06) Preset Single Register (Set Registers, settings, configuration Info).

To reset the state of internal registers and write to the registers and sensors, a command uses function code 06 to send the value to update the register in remote device. See Raveon's lists of registers called IOREG and VARS. These are the registers that may be able to be written with function 06 if the register is writable.

If this command is used to set the status of an output bit register, such as register 1001 which holds the status of output bits, all output bits will get set with this command. Even if they are digital outputs, open collector, or DC output, all output bits will be set with register 1001. Use caution when presetting registers like this.

Two write to a single register, message to the RTU slave is:

Byte 0:	Slave Address. The ID of the RTU device
Byte 1:	Function Code for this message. 06
Byte 2:	Register number MSB
Byte 3:	Register number LSB
Byte 4:	Data to store, MSB
Byte 5:	Data to store, LSB
Byte 6:	*Error Check Low Byte (LSB)*
Byte 7:	*Error Check High Byte (MSB)*

The structure of the 03 READ HOLDING REGISTERS response is:

Byte 0:	Slave Address. The ID of the RTU device
Byte 1:	Function Code for this message. 06
Byte 2:	Byte Count (**N**) being returned
Byte 3:	Register data read, MSB
Bytes 4:	Register data read, LSB
Byte 4+N:	Byte #5 (Optional based on data number **N**)
Byte 5+N:	Byte #6 (Optional based on data number **N**)
Byte 6+N:	*Error Check Low Byte (LSB)*
Byte 7+N:	*Error Check High Byte (MSB)*

So for example, the telegram to read bit #2 (Coil address 2) on device 3 is:

Request: 19 06 03 E9 00 03 1B A3
Response: 19 06 03 E9 00 03 1B A3 (it reported coil 2 is on)

Register List

This is a list of the registers the RAZN has within it, that can be accessed, read, or set. Manu registers are used to indicate the status of IO pins. Some registers indicate the status of particular bits. Some registers hold the voltage level of analog voltage inputs.

This is Raveon's register map of register address for features, IO bits, and internal registers in the RAZN and most other SCADA products. It lists the register addresses to use for accessing these registers with Modbus commands.

In the Read/Write columns, R means read, W means Write, R/W means both Read and writeable. N means the register is implemented, but may not have been fully tested by our team. If the space is blank, it means it is not implemented, and will probably throw an error if it is accessed with Modbus protocol commands.

Address (Decimal)	**Description**	**Data type**	**More Info**	**Read/ Write**
1	GPIO *OUTPUT* bit. Base is IO#0 bit	U16	These are the main GPIO bits on the front of the product Address 1-16 are bits 0-15. These can be used for Switched Power, Digital Inputs, Digital outputs, Button inputs, Optical Isolated digital Inputs,…	R/W
2	IO#1 bit GPIO	U16		R/W
3	IO#2 bit GPIO	U16		R/W
4-64	IO# bits 4-64	U16		R/W
100	GPIO *INPUT* Bit, base is IO0. GPIO port must be set in input mode to read this bit.	U16		R
101	IO1 bit GPIO if set to Input mode.	U16		R
102	IO2 bit GPIO if set to input mode.	U16		R
103-115	IO bits 4-16 if set to input mode.	U16		R
1000	Merged IO status register, with both the input and output states for all 16 IO pins in one Register.	U16	The combined digital status INs/OUTs for telemetry as SCADA products. Merges all IO such as digital in, out, Open Drain outs, and switched voltage outputs.	R
1001	All Output Bits status register for all 16 IO pins in one Register.	U16	This register holds the bit state of output pins 0-15 within this one register. GPIO ports IO0-IO2 are bits 0, 1, and 2 in this register. Writing can change all output bits used by this register.	R/W
1002	All Input Bits status register for all 16 IO pins in one Register.	U16	This register holds the state of input bits 0-15 pins within the one register. GPIO ports IO0-IO2 are bits 0, 1, and 2 in this register.	R
1003	Input Polarity Register	U16	IO pin inputs can have polarity inverted. 16 bits reflect polarity inversion. 0=none, 1=inverted.	R
2200	*Up Counter* registers for IO0 - IO15 bits when in input mode. Each bit is one register. IO0 uses the first register, IO1 uses the next…	U16	The *Up Count* register ticks up every time an <u>Input bit</u> state changes from 0 to 1. Write commands can be used to reset the counter. Use **2600** to periodically reset this counter to 0 of you want.	R/W
2300	Up Timer registers for IO0 - IO15 bits times how the bit has been up (1). IO0 uses the first register, IO1 uses the next…	U32	The Up Timer register counts the time in mS a bit is on (1). Write commands can be used to reset the timer. Time is in milliseconds. (mS). Returns 4 bytes (32bits) per register.	R/W
2400	Down Timer Registers for IO0 - IO15 bits times how long the bit has been down (0). IO0 uses the first register, IO1 uses the next…	U32	The Down Timer register counts the time a bit is off (0). Write commands can be used to reset the timer. Time is in milliseconds (mS). Returns 4 bytes (32bits) per register. If the bit is 1, the Down Timer Register is reset to 0.	R/W
2500	*Reset Interval* It is the number of <u>seconds</u> how often the Pulse Counter is reset to 0.	U16	Registers **2200** and **2600** are reset at this register's time. If set to 0, then 2200 Up Count does not reset at all. His is one register used for all IOs.	R/W
2530	*Previous Up Counts when reset.* The number from **2200** when reset.	U16	When the **2200** registers are reset to 0 in the time shown on **2500**, it is stored here. If 2500 was set to 1 minute, this shows the pulses in the previous minute.	R/W
2600	*ON Time Counter* It is the number of Seconds an IO pin is ON or was On.	U16	When On, it ticks up, when OFF it shows last on-time. Register 2600-2615 are for IO# terminals 0-15.	R/W

2700	Bit Mask of outputs to Pulse On. It pulses these On, in mS specified in 2701. This reg. sets and shows the pins that are pulsed on.	U16	IO# terminal ports are in this bit mask. Bit 1 is IO#0. IO#0 - IO#15 are this 16 bits to tag as should be pulsed ON. Register 2710 is the number of mS on-time to puts these ON. Register 2700 and 2701 must be written at the same time.	R/W
2701	Time is in mS of the bits specified in this reg. 2700 ref specifies which outputs to pulse on. 0, the pulsed outputs are OFF, not on.	U16	Number of mS to pulse all the IO terminals tagged in the register 2700 bit mask. And they will stay on for this many milliseconds (mS). When the time is over, the bits specified in 2700 register will be turned off. When set more than 0, Pulse starts and this ticks down 1 every 1mS. When. You can read it also.	R/W
2900	IO Error log. Bits 0-15 are IO# 0-15	U16	If any error occurs on an IO terminal pin, it is set in this bit mask. IO$0 is bit 0 IO#5 is bit 5.. These are all combined from error logs 2902-2903	R/W
2901	IO Under Current status log. Bits 0-15 are IO# 0-15 Bit state: 0=OK, 1=Error.	U16	If an under-current error occurs on an IO terminal pin. If the error goes away, the IO bit is cleared in this log. IO pins that use this are **SD** and **DSD**.	R/W
2902	IO Over Current status log. Bits 0-15 are IO# 0-15 Bit state 0=OK, Bit state 1=Over Power.	U16	If an over-power current draw occurs on an IO terminal pin, it is set in this bit mask. Over Power is: Over Current. IO pins that use this are **SD** and **DSD**. These are not cleared even if the error goes away. You can clear them	R/W
2903	IO Under Current status log. Bits 0-15 are IO# 0-15 Bit state: 0=OK, 1=Error.	U16	If an under-current error occurs on an IO terminal pin. If the error goes away, the IO bit is cleared. IO pins that use this are **SD** and **DSD**.	R/W
2904	Low current threshold in mA for solenoid driver current monitoring	U16	IOs that trigger this are tagged in2903 register. Command **STSDL** can set or read this also.	R/W
2905	High/Max current threshold in mA for solenoid driver over-current monitoring.	U16	IOs that trigger this are tagged in2902 register. Command **STSDM** can set or read this also.	R/W
3300	Up Timer registers for IO0 - IO15 bits times how many seconds the bit has been up (1). IO0 uses the first register, IO1 uses the next…	U16	The Up Timer register counts the time a bit is on (1). Write commands can be used to reset the timer. Time is in Seconds. Register 3300-3315 are for IO# terminals 0-15. These are from the Register 2600-2615 mS registers converted to Seconds here.	R/W
3400	Down Timer registers for IO0 - IO15 bits times how man seconds the bit has been down (0). IO#0 uses the first register, IO#1 uses the next…	U16	The Down Timer register counts the time a bit is off (0). Write commands can be used to reset the timer. Time is in Seconds. These are from the Register 2400-2415 mS registers converted to Seconds here.	R/W
3450	Solenoid Time Pulse Duration mS…	U16	3450 is IO# 0. 3450-3465 are registers for IO numbers 0-15.	R/W
3480	Solenoid IN/OFF Time Duration Seconds	U16	3480 is IO# 0. 3480-3485 are registers for IO numbers 0-15.	R/W
5000	Input Voltage to device, mV	U16	Voltage in millivolts of the DC input power to the product/modem.	R
5001	Device Current Draw	U16	Current draw of the product in mA. Devices without current sensors return 0, or throw exception	N
5002	Device temperature	INT16	The temperature of the product, in degrees C. The register is a polarized 16 bit number.	R
5003	GPIO State register	U16	The combined digital status INs/OUTs for telemetry and SCADA products. This returns a merged status of both input bits and output bits.	R

5004	GPIO Digital Input config	U16	The bits in this register are GPIO IO# bits that are configured as Digital Inputs	R
5005	GPIO Digital Output config	U16	The GPIO bits IO# that are configured as Digital Output bits	R
5006	GPIO Open Drain config ***RD***	U16	The GPIO bits IO# that are configured as Open Drain output functions such as Relay drivers.	R
5007	GPIO DC Switched config ***SV***	U16	The GPIO IO# terminal ports that are configured as AV/DC switched output functions.	R
5011	GPIO State register ***DSD***	U16	The GPIO IO# terminal ports that are configured as Dual Pin Solenoid Driver.	R
5012	GPIO State register ***SD***	U16	The GPIO IO# terminal ports that are configured as three (3) Pin Solenoid Driver.	R
5013	GPIO State register ***S BI***	U32	The GPIO IO# terminal ports that have Input Bits that have Pull-up resistors enabled. 4 Bytes in this register: Days, Hours, Minutes, Seconds. Bytes are dddd hhhh mmm ssss	R
	System Information			
5030	Power ON timer	U32	How long has this product been turned ON. The 4 bytes in this register dd,hh,mm,ss Top dd is days, hh is hours, mm is minutes, ss is seconds.	R
	Optional Registers in some		Raveon SCADA products.	
5200	Accelerometer, X	I16	The G force in X axis of the device. Option in some products. millG -90 to +90	R
5201	Accelerometer, Y	I16	The G force in Y axis of the device. Option in some products. millG -90 to +90	R
5202	Accelerometer, Z	I16	The G force in Z axis of the device. Option in some products. millG -90 to +90	R
5203	Accelerometer, Motion	U16	The value of the motion.	R
5204	Accelerometer, Motion Yes/No	U16	The motion. 0=no, 1=Yes	R
5205	Accelerometer, No Motion Time	U32	The number of mS with no motion.	R
5206	Accelerometer, Motion Time	U32	The number of mS with motion occurring.	R
5250	Momentary Bits monitored ON time and auto-reset to 0.	U16	The bits array indicating which Output ports are monitored.	R
5251	Momentary time in seconds used to time the momentary bits.	U16	The number of seconds the Momentary bits can only be on.	R
5252	Momentary reset counter	U16	The number of times a momentary bit has been reset to 0/off. Each bit each time increments this register counter.	R
6000	Analog input, ADC value. 6000-6015. Register 6000 is input 0, 6001 is input 1	U16	The ADC's register value. Stored last time it was read.	R
6040	Analog inputs in High resolution differential ADCs. IO*0 –IO15 terminals is 6040-6055	I32	The ADC's register value. Stored last time it was read. Typically 24+ bits. Signed value	R
6060	Temperature sensors on the ADS input are computed here.	I16	The calculated temperature from the ADS inputs and the specified sensor. 16 bit signed value.	R
6100	Text Output Register for	TXT	The limit to the number of bytes sent to this register is	W

	sending a text message out the radio modem's serial port.		246 bytes.	
6400	Analog input voltage value. 6000-6015. Register 6400 is input 0, 6401 is input 1.	F24	The measured voltage of this input, based on calibration of the RAZN.	R
6500	Input current sensor value. Register 6500 is input 0, 6510 is input IN# 10	U16	The measured input current value in mA (milliamps). (7.784mA = 7784)	R
6540	4-20mA 16 bit integer current sensor value. Register 6540 is input 0, 6550 is input IN# 10	I16	16 bit signed value. The measured 4-20mA input current value is full range of this register. 4mA=0, 20mA is 7FFF.	R
6600	SCADA Inbound command reception count.	U16	Tick up each time a SCADA command comes in.	R
6601	SCADA Inbound message timer.	U16	Number of seconds since a message came in.	R
6700	Set GPIO/FIO output states on various bits 0-7	U16	Upper 8 bits specify the bits to manage. Lower 8 bits set the desired state to set for the specified bits. Hex example: 3130 will set bits 5&4 and clear bit 0.	W
6701	Set GPIO/FIO output states on various bits 8-15	U16	Upper 8 bits specify the bits to manage. Lower 8 bits set the desired state to set for the specified bits. Hex example: 3130 will set bits 13&12 and clear bit 9.	W
6800	Current sensor value last measured at active time.	U16	The measured current value in mA (milliamps). (7.784mA = 7784) IO#0 –IO#15 terminals: 6800-6815	R
6840	Current Sensor MAX value ever read on this IO.	U16	The measured current value in mA (milliamps). (7.784mA = 7784) IO#0 –IO#15 terminals: 6840-6855	R
7000	MIMIC mode bit designators	U16	Bits 0-15 designate which **IO#** bits are used in the MIMIC mode. This are the bits that have their state send out automatically.	R
7001	MIMIC mode bit states	U16	The bits in this register are the digital state of the **IO#** terminals used in the MIMIC mode.	R
7003	MIMIC mode bit status	U16	The status returned from the remote MIMIC in this register are the digital state of the **IO#** terminals used in the MIMIC mode.	R
7004	Time in Seconds since the last MIMIC slave sent a status message.	U16	When this device powers up, it is set to 65000. It is rest to 0 when a slave sends as status message to this device, and ticks up every second. Max value is 65000.	R
12000	SCADA Product Version.	U32	Returns a U16 code for the type of product it is. Upper by "M" for modems. Lower byte: 01=GPIO 3 bit.	N
12001	Products Serial Number.	U32	The electronic serial number inside this product. Each product has a unique electronic serial number.	N
12002	Product Thing code.	U16	Each RAZN version has a type code. This is the code after the model number dash: N55**-1,** N55**-3A,** N55**-3B…**	N
12003	SCADA Product Version.	U32	Returns a U16 code for the type of product it is. Upper by "M" for modems. Lower byte: 01=GPIO 3 bit.	N
12004	Data Logging size.	U16	Maximum Data Log size in this product	R
13000	COMM_INFO **S1** Communication safety	U16	RS485 serial port communication safety information,	N

	Information.			
13010	COMM_INFO **S2** Communication safety Information.	U16	RS232 serial port communication safety information,	N
13020	COMM_INFO **R1** Communication safety Information.	U16	UWORC internal RF modem communication safety information,	N
13030	COMM_INFO **E1** Communication safety Information.	U16	TCP/IP interface #1 communication safety information,	N
13040	COMM_INFO **E2** Communication safety Information.	U16	TCP/IP interface #2 communication safety information,	N
13050	COMM_INFO **E3** Communication safety Information.	U16	TCP/IP interface #3 communication safety information,	N
15000	Data Log	Bytes	The bytes for the Data Logger. Log1 is 15000. Log 2 is 15001, Log 1000 is 15999.	
56000	Firmware Version information.	U32	Returns 4 bytes. Text bytes such as B17_	N

Register Data Types

U16	16 Bit unsigned Integer
U32	32 bit unsigned Integer
I16	Signed 16 bit Integer
I32	Signed 32 bit Integer
TXT	Text, 8 bit characters in an array.
F32	32 bit Floating Point number.

Copyright, Notices, and Trademarks

Raveon Technologies Corporation

2320 Cousteau Court

Vista, CA 92081

sales@raveontech.com

760-444-5995