

# Scully Rack Controller System

# **Intellitrol**

## Modbus RTU Protocol Specification

Written by  
Gary Cadman  
Stephen F. Tougas  
Robert D. Houk

9/1/2005

**CONFIDENTIAL**

## Revision History

Revision	Who	Date	Changes Made
2.3	SFT	03/27/95	<ul style="list-style-type: none"><li>Added Write Multiple Register command 10 hex &amp; UNIX date registers</li></ul>
2.4	SFT	04/20/95	<ul style="list-style-type: none"><li>Added Gary Cadman's, Frank Ski's, and Scott McCutcheon's comments. Changed Main Relay State and Backup Relay State registers, introduction, changed future response time to 100 milliseconds, changed maximum Intellitrol vehicle numbers to 5,000, and changed program load delay to 60 seconds. Added 32 vehicle bypass key list to vehicle list from vehicle 10,000 to 10,031, added compatibility modules to Truck Type State, added command size limitation, added Intellitrol LED's to Read Input Status, added force bits to load low and high Kernels, added force bits to erase the bypass key list, added logging of remote authorization on the Intellitrol, added examples for setting VIP and Intellitrol clocks, added log entry for recording erasing the bypass key list, and added registers to control error logging periods. Removed PERMIT LED and NONPERMIT LED from Read Output Status and fixed bit meanings, removed extra probe state registers, and removed backup functions for reading the S/N and date and time.</li></ul>
2.5	SFT	04/26/94	<ul style="list-style-type: none"><li>Fixed labeling, added explanation text to Bypass Activity register 48H, put conditions on the acquire state register, removed fail-safe monitoring from Probe State registers 50H through 57H, added Overall Truck State register 58H, added state to Probe State registers 50H through 57H and reworded states, corrected register function names (i.e. single/multiple), added Read Output Status comments, added and revised Read Input Status comments including adding Lo and Hi Kernel Running bits, changed program loading force bits, removed program load low and high Kernel force bits, changed boot-loading description, added S3, S5, and S7 records to the Motorola S-Record Format, replaced Kernel Checksum Error log entry and replaced with New Mode Forced, replaced RAM error log entry with Reset Occurred, changed name of Bypass log entry to VIP bypass, added bypass log entries for Overfill, Vapor, and Ground, added Read Scully Log function 49 hex, modified Changes From Revision 1.4 text, added initial Read Input Bits jumper settings, added Read Scully Log Element function code 49H to read the Scully Log.</li></ul>
2.6	RDH	11/15/95	<ul style="list-style-type: none"><li>Change all "Cybertrol" to Intellitrol; minor formatting changes and prettification; Relayout sections to "make it flow better"; Sectionize for page numbering (TOC in roman, etc.).</li></ul>
2.7	RDH	12/03/95	<ul style="list-style-type: none"><li>Massive revamp/update. New Modbus register layout. Add new custom Modbus commands: CRC Multiple Vehicles; Write Bypass Keys; Read Bypass Keys; Write Feature-Enable Password; Read NV/EEPROM Block; Write NV/EEPROM Block. Reflect Intellitrol Beta version 0.5.E3 / Release 1.0</li></ul>
2.8	MR	12/29/97	<ul style="list-style-type: none"><li>Added information to reflect changes in Version 1.05 Software</li></ul>
3.0	GRC	8/31/05	Revised for new Intellitrol CPU board.
3.1	HP	11/4/19	Updated for version 1.7.0

## Table of Contents

1.	Introduction and Overview .....	1
1.1.	Associated Documents .....	2
1.2.	Communications Interface .....	2
1.3.	Communications Response Time .....	2
1.4.	General Error Recovery Strategy .....	3
1.5.	Compatibility With Other Equipment .....	3
2.	Data Structures .....	5
2.1.	Vehicle List Data Structure .....	5
2.1.1.	Intellitrol Vehicle List Features and Operations .....	5
2.1.1.1.	Intellitrol Supported Features/Operations .....	6
2.2.	Bypass Key List Data Structure .....	6
2.3.	VIP Bypass Log Data Structure .....	6
2.3.1.	Supported Features and Operations .....	7
2.4.	Event Log Data Structure .....	7
2.4.1.	EEPROM Initialized [Event Type 01] .....	9
2.4.2.	System Reset [Event Type 02] .....	9
2.4.3.	Bypass Activity [Event Type 03] .....	9
2.4.4.	Hardware Error [Event Type 04] .....	10
2.4.4.1.	Firmware CRC-16 Fault [Hardware Error Subtype 00] .....	10
2.4.4.2.	Relay Fault [Hardware Error Subtype 01] .....	10
2.4.5.	Voltage Error [Event Type 05] .....	11
2.4.6.	Impact Sensor Tripped [Event Type 06] .....	11
2.4.7.	Overfill Info [Event Type 07] .....	11
2.4.8.	Maintenance Error [Event Type 08] .....	12
2.4.9.	Reset Info [Event Type 09] .....	12
2.4.10.	Overfill Info 2 [Event Type 0A] .....	12
2.5.	VIP and Intellitrol Force Bit Assignments .....	13
2.5.1.	Forced Shutdown [Force Code 00] .....	14
2.5.2.	Forced Recover [Force Code 02] .....	14
2.5.3.	Erase Vehicle List [Force Code 03] .....	14
2.5.4.	Erase Log [Force Code 04] .....	14
2.5.5.	Hardware Reset [Force Code 06] .....	14
2.5.6.	Force Overfill Bypass [Force Code 08] .....	14
2.5.7.	Force Ground Bypass [Force Code 09] .....	15
2.5.8.	Enable Ground Fault Detection [Force Code 0A] .....	15
2.5.9.	Enable Deadman Switch [Force Code 0F] .....	15
2.5.10.	Erase Bypass Key List [Force Code 12] .....	15
2.5.11.	Erase EEPROM [Force Code 13] .....	16
2.5.12.	Force VIP Bypass [Force Code 15] .....	16
2.5.13.	Enable VIP [Force Code 16] .....	16
2.6.	Input Status Bits .....	16
2.6.1.	Fault [Input Status Bit 00] .....	18
2.6.2.	Truck Present [Input Status Bit 01] .....	18
2.6.3.	Truck Talk [Input Status Bit 02] .....	18
2.6.4.	Truck in Vlist [Input Status Bit 03] .....	18
2.6.5.	Bypass [Input Status Bit 04] .....	19
2.6.6.	Idle [Input Status Bit 05] .....	19
2.6.7.	Permitting [Input Status Bit 06] .....	19
2.6.8.	Non Permissive [Input Status Bit 07] .....	19
2.6.9.	Debug Jumper [Input Status Bit 08] .....	19
2.6.10.	CH5 High Resistance [Input Status Bit 09] .....	19

# Intellitrol/VIP Modbus RTU Protocol Specification

---

2.6.11.	Deadman OK [Input Status Bit 12] .....	19
2.6.12.	Diode GND [Input Status Bit 13] .....	20
2.6.13.	Resistive GND [Input Status Bit 14] .....	20
2.6.14.	Intellicheck [Input Status Bit 15] .....	20
2.6.15.	Bad EEPROM [Input Status Bit 17] .....	20
2.6.16.	ADC Time-out [Input Status Bit 18] .....	20
2.6.17.	Shell CRC Error [Input Status Bit 19] .....	20
2.6.18.	Clock Error [Input Status Bit 20] .....	20
2.6.19.	Bad CPU [Input Status Bit 21] .....	21
2.6.20.	Truck [Input Status Bit 22] .....	21
2.6.21.	Kernel CRC Error [Input Status Bit 23] .....	21
2.6.22.	Voltage Error [Input Status Bit 24] .....	21
2.6.23.	Tim Data Line Fault [Input Status Bit 26] .....	21
2.6.24.	Ground Fault [Input Status Bit 28] .....	21
2.6.25.	Special Ops Mode [Input Status Bit 29] .....	21
2.6.26.	Shutdown [Input Status Bit 30] .....	22
2.6.27.	Relay Error [Input Status Bit 31] .....	22
2.7.	Output Status Bits .....	22
2.7.1.	Five-Wire-Optic Output Pulse [Output Status Bits 04 - 05] .....	23
2.7.2.	Five-Wire-Optic Echo Pulse [Output Status Bits 06 - 07] .....	23
2.7.3.	Vehicle Communications [Output Status Bits 08 - 09] .....	23
2.7.4.	S-Record Counter [Output Status Bits 16 - 31] .....	23
2.8.	16-Bit Control and Data Registers .....	24
2.8.1.	Shell Version [Register 005] .....	29
2.8.2.	Wait for TAS Delay [Register 008] .....	30
2.8.3.	Bypass Active Time [Register 009] .....	30
2.8.4.	Terminal Number [Register 00A] .....	30
2.8.5.	Modbus Minimum Response Delay [Register 00B] .....	30
2.8.6.	Shell Checksum [Register 00C] .....	31
2.8.7.	VIP Mode Control [Register 00E] .....	31
2.8.7.1.	VIP Mode Control Bit Assignments .....	31
2.8.8.	Unit Serial Number [Registers 020-023] .....	31
2.8.9.	Hardware Revision Level [Register 024] .....	32
2.8.10.	Config-A [Register 025] .....	32
2.8.11.	Config-B [Register 026] .....	32
2.8.12.	UNIX Date/Time [Registers 100-101] .....	33
2.8.13.	Event Elapsed Time [Registers 102-103] .....	34
2.8.14.	Status-A [Register 104] .....	34
2.8.15.	Status-B [Register 105] .....	35
2.8.16.	Truck Serial Number [Registers 10A-10C] .....	35
2.8.17.	Probe State [Registers 10D-114] .....	36
2.8.17.1.	Probe States .....	36
2.8.18.	Bypass State [Register 115] .....	37
2.8.19.	Bypass Serial Number [Registers 116-118] .....	37
2.8.20.	Bypass Time [Register 119] .....	38
2.8.21.	Non-Permit Reasons [Register 11A] .....	38
2.9.	TIM Builder Info .....	39
3.	Modbus Functions .....	46
3.1.	Read Output Status Function Code 01 .....	47
3.1.1.	Example Query Message Reading Status Bits 0 - 2 .....	47
3.1.2.	Example Response Message Reading Status Bits 0 - 2 .....	47
3.1.3.	Example Query Message Reading Status Bits 16 - 31 .....	47
3.2.	Read Input Status Bits Function Code 02 .....	47
3.2.1.	Example Query Message Reading Input Status Bits 0 - 15 .....	48
3.2.2.	Example Response Message Reading Input Status Bits 0 - 15 .....	48

# Intellitrol/VIP Modbus RTU Protocol Specification

---

3.3.	Read Multiple 16-Bit Registers Function Code 03 .....	48
3.3.1.	Example Query Message Read Registers 3 and 4 .....	48
3.3.2.	Example Response Message Read Registers 3 and 4 .....	48
3.4.	Force (Set) Single Bit Function Code 05 .....	49
3.4.1.	Example Query Message To Erase The Vehicle List .....	49
3.4.2.	Example Response Message To Erase The Vehicle List .....	49
3.5.	Write Single 16-Bit Register Function Code 06 .....	49
3.5.1.	Example Query Message Write Register 0E .....	50
3.5.2.	Example Response Message Write Register 0 .....	50
3.6.	Write Multiple 16-Bit Registers Function Code 10 Hex .....	50
3.6.1.	Example Query Message Write Registers 100 - 101 (Set Intellitrol Time) .....	50
3.6.2.	Example Response Message Read Registers 100 - 101 (Set Intellitrol Time) .....	50
3.7.	Write Single Vehicle Function Code 41 Hex .....	51
3.7.1.	Example Query Message .....	51
3.7.2.	Example Response Message .....	51
3.8.	Read Single Vehicle Function Code 42 Hex .....	51
3.8.1.	Example Query Message To Read Single Vehicle .....	51
3.8.2.	Example Response Message .....	52
3.8.3.	Example Response Message .....	52
3.9.	Write Multiple Vehicles Function Code 46 Hex .....	52
3.9.1.	Example Query Message .....	52
3.9.2.	Example Response Message .....	53
3.10.	Read Multiple Vehicles Function Code 47 Hex .....	53
3.10.1.	Example Query Message .....	53
3.10.2.	Example Response Message .....	53
3.11.	CRC Vehicle List Function Code 4A Hex .....	54
3.11.1.	Example Query Message .....	54
3.11.2.	Example Response Message .....	54
3.12.	Write Bypass Keys Function Code 4B Hex .....	54
3.13.	Read Bypass Keys Function Code 4C Hex .....	55
3.14.	Write "Enable-Features" Password Function Code 4D Hex .....	55
3.14.1.	Example Message Set Enable Password .....	55
3.14.2.	Example Message Set Enable Password Response .....	55
3.15.	Normal Modbus Message Format .....	55
4.	Modbus Exception Response Messages .....	57
4.1.	Example Exception Response Message .....	57
4.2.	Exception Response Codes .....	58
5.	CRC Generation .....	60
5.1.	"C" Implementation of CRC-16 .....	60
5.1.1.	Big'n'Fast "C" Implementation of CRC-16 .....	60
5.1.2.	Small'n'Slow "C" Implementation of CRC-16 .....	63

## 1. Introduction and Overview

This document describes the communications protocol supported by the Intellitrol and VIP rack controller units to support the exchange of control and status information with Terminal Automation System (TAS) software. The Intellitrol and the VIP units support an RS-485 half duplex multidrop serial communications line. The inter-unit or “network” protocol implemented is based on the Modicon, Inc. “Modbus™” industrial automation control protocol, extending the protocol semantics to better encompass peer-to-peer (computer system to computer system) interchange of data. This protocol allows multiple Intellitrol units, VIP units, and equipment from other manufacturers to be intermixed on and share the same communication line. The Modbus protocol relies on the bus master (TAS) polling bus slaves (Intellitrol and VIP units) to detect arrival, loading, and departure of trucks. System response time to truck transactions is limited by the polling rate of the TAS.

The Scully Intellitrol/VIP Communications Protocol is designed to comply with The Modicon Modbus Protocol Reference Guide (PI-MBUS 300 Rev E) of March 1993 where possible. Consult the Reference Guide for Modbus details not specified in this document. The Intellitrol and VIP rack controller units use Modicon defined Modbus functions 1 and 2 to read single bits, function 5 to set single bits, and functions 3 and 6 to read and write 16-bit integers. “User Defined” functions 41 through 4F (hex) are Scully extensions to the Modbus protocol to support the Intellitrol and VIP rack controller units. All Modbus protocol functions used by either the Intellitrol or VIP rack controller units are detailed in this document (see the *Data Structures* and *Modbus Functions* sections later in this document).

The Intellitrol has two microprocessors able to communicate on the Modbus. The backup processor can also communicate its status on the Modbus when requested. The main processor passes the message onto the backup processor. The Scully Intellitrol rack controller unit is an intrinsically safe overfill prevention system for monitoring loading of tanker trucks. Optionally, the Intellitrol may also include VIP (Vehicle Identification Prover), ground-proving safety, vapor-flow-sensing, and/or deadman-switch functionality. With the inclusion of the VIP (and/or ground) options, the Intellitrol rack controller unit is a self-contained complete replacement for, and superset of, the VIP rack controller unit (and/or the ST-47 Ground Prover rack controller unit) with included overfill prevention. The Intellitrol recognizes a tanker truck connecting to the terminal and allows it to load liquid petroleum products so long as it is correctly electrically grounded (if optionally configured for ground proving capability) and its overfill sensors are functioning properly and indicating “dry” (non-overfill). Additionally, if VIP functionality is incorporated, the truck must be properly identified by serial number. Trucks which are not configured properly or are dysfunctional can be bypassed by physically presenting a special electronic bypass key to the rack controller unit. The TAS may also bypass (as well as “unbypass”) the Intellitrol rack controller remotely, as well as exert explicit authorization control (in both a normal and override scheme) over the VIP subsystem. The deadman switch operation, if enabled, cannot be bypassed.

The Intellitrol overfill prevention system is a dual-processor active fail-safe architecture. The Intellitrol internally has two completely independent (insulated from each other) microprocessor subsystems with separate active probe-monitoring circuitry. The two microprocessors (“Main” and “Backup”) must *both* agree that *all* the truck’s compartment’s overfill probes indicate it is safe to “permit” before the Intellitrol rack controller unit will permit. Each microprocessor has its own permit relay; the two permit relays are wired in a series configuration. Each microprocessor must actively and continuously drive its permit relay for the relay to activate (close its contacts). Further, if both relays are determined to be shorted (or, in other words, the Intellitrol is “hot-wired” and cannot control the terminal system), the Intellitrol flashes a special Attention/Warning pattern to alert terminal personnel of the terminal lane’s unsafe condition. Each microprocessor has its own hardware “watchdog” circuit to further ensure that each microprocessor is operating properly.

Extensive dynamic status information on the internal state of the Intellitrol rack controller is available for TAS operations to monitor, log, and/or display. In addition, the Intellitrol rack controller maintains an internal

“Event Log” in which detected problems (and “interesting” events in general, such as system resets and bypass activity) are recorded. This Event Log is also accessible to TAS operations.

The Scully Vehicle Identification Prover (VIP) is a system to automatically identify and allow loading of road tankers based on date/time-sensitive information associated with a unique serial number stored in the tanker. The VIP can either be a subsystem within the Intellitrol rack controller unit or can be installed at a terminal as a separate standalone unit. Road tankers which use the VIP system are equipped with either a Scully Truck Identification Module (TIM) containing an electronic serial number that is laser-etched (and cannot be “reprogrammed” or changed) into the TIM's Read-Only Memory (ROM) at the time of manufacture, or a Scully IntelliCheck on-board control system which contains a similar electronic serial number. The rack controller interrogates the on-truck TIM through the Scully overfill prevention socket (pin 9 which is shared with the “ground bolt” in so-equipped trucks). The VIP unit contains a list of authorized serial numbers (the Vehicle List) downloaded from the Terminal Automation System (TAS) over the RS-485 Modbus serial line. If the road tanker's TIM number is in the vehicle list, the permit relay closes, and a green "authorized" LED on the front display panel of the rack controller unit illuminates after the "wait for TAS delay" has expired. Otherwise, the permit relay stays open, and a red "unauthorized" LED illuminates. The TAS must load the vehicle list into the rack controller unit and keep it current if the VIP system is to authorize road tankers without TAS control.

## 1.1. Associated Documents

1. Modicon Modbus Protocol Reference Guide (PI-MBUS 300 Rev E) March 1993.
2. Scully Universal VIP Software Requirements Specification Version 0.6 written by Gary Cadman #61287.
3. Scully Intellitrol Rack Controller Product Specification Revision B written by Art Shea June 9, 1994 #P-020-01.

## 1.2. Communications Interface

The Scully Modbus protocol supports the following:

1. Half Duplex RS-485 Multidrop.
2. 19200, 9600, 4800, 2400, and 1200 baud.
3. Modbus RTU (straight binary).
4. 8 bits no parity bit, or 8 bits plus an even parity bit, or 8 bits plus an odd parity bit.

The AEG Modicon test program MTS uses 8 bits plus an even parity bit. The Scully test program Intelliview uses 9600 baud 8 bits no parity by default. The Intellitrol and VIP rack control units have hardware jumpers to select line speed and parity (or no parity) type.

## 1.3. Communications Response Time

Intellitrol/VIP rack controller units typically respond to bus master (TAS) query messages within milliseconds of the last character of the command message. A rack controller unit may take longer to process some query messages. A bus collision and garbled data will occur if the bus master transmits its next query before 0.1 second has elapsed and the rack controller unit has not responded. Particularly time-consuming commands such as erasing the vehicle list may cause the rack controller unit to return the ACKNOWLEDGE exception message. The SLAVE DEVICE BUSY exception message will be returned if the bus master attempts communication and

use the feature that's is being affected, before the rack controller unit completes its current task. A minimum response time can be programmed via the *Modbus Minimum Response Time* register to allow the bus master time to turn the line around between master message and slave response. When sending broadcast messages, the bus master must wait at least 1.0 second before sending the next message.

### 1.4. General Error Recovery Strategy

Unknown function codes, out of range addresses, hardware failures (e.g. stuck bits in EEPROM), and similar problems will cause an exception response in accordance with Modicon manual Appendix A. EEPROM write failures will return the Memory Parity Error exception code.

### 1.5. Compatibility with Other Equipment

Other Modbus RTU equipment (e.g. card readers, meters, etc.) will interoperate with Intellitrol/VIP rack controller units on the same communication line so long as each Modbus unit is assigned a unique "Modbus Address". Each rack controller has jumpers to select an address in the range of 0 to 59 decimal (0 to 99 on the Intellitrol). Modbus RTU messages always start off with the Modbus address as the first byte of the message. Scully rack controller units supports a special "broadcast" address of 128 (decimal). Any message transmitted to address 128 (decimal) will be accepted by all Scully rack controller units on the communication line. This feature is used to set the real time clocks of all rack controller units at once or to load new program code into all rack controller units at once. To prevent bus contention, the rack controller units never reply to broadcast messages. The rack controller broadcast address differs from the standard Modbus broadcast address of zero. Intellitrol rack controller units should never be assigned address 128 (decimal). Rack controller units ignore messages broadcast to address zero. (Address "0" is reserved for a special "ASCII debug/trace output" mode of operation wherein all Modbus support is disabled.)

Modbus ASCII messages always start off with an ASCII colon character ":" (58 decimal). Modbus ASCII protocol units will ignore any messages that do not start off with a 58 (decimal) byte. So long as no Modbus unit is assigned address 58 (decimal), the ASCII units will ignore the RTU messages and vice versa.

Brooks Computer Protocol messages always start off with ASCII SOH character (1 decimal). Brooks protocol units will ignore any messages that do not start off with a 1 byte. So long as no Modbus unit is assigned address 1, the Brooks units will ignore the Modbus messages, and the Modbus units will ignore the Brooks messages.

Brooks Terminal Protocol is very similar to Brooks Computer, with the exception that each message starts with an exclamation point "!" (33 decimal). So long as no Modbus unit is assigned address 33 (decimal), the Brooks Units will ignore Modbus messages, and the Modbus units will ignore Brooks messages.

Waugh and Smith units use proprietary protocols that always start with an asterisk "\*" (42 decimal). So long as no Modbus Unit is assigned an address of 42 (decimal), the Smith and Waugh units will ignore Modbus messages, and the Modbus units will ignore messages to the Smith and Waugh units.

New shell programs are loaded into VIP rack controller units using Motorola S-Records. Each S-Record starts with an ASCII capital "S" 83 (decimal). So long as no Modbus Unit is assigned an address of 83 (decimal), no rack controller unit on the communication line will interpret any message as a Motorola S-Record.

In summary, avoid selecting 0, 1, 33, 42, 58, or 83 or 128 (decimal) as Modbus addresses for any Modbus unit (not just Intellitrol and VIP units) on the communication line. The firmware in the non-Modbus units must be robust enough to reject Modbus messages.



## **Intellitrol/VIP Modbus RTU Protocol Specification**

---

Scully does not recommend putting non-Modbus RTU equipment on the same serial line as the Modbus with Intellitrol and VIP units. A "mixed" protocol bus can be very difficult to troubleshoot if any unit malfunctions. Since Modbus RTU is binary, the attention and/or end of message characters for the non-Modbus unit may appear accidentally and randomly inside Modbus messages. This may confuse non-Modbus equipment, causing them to issue error messages to traffic not directed to them, resulting in communication errors and bus contention. Using only the Modbus RTU protocol on the same serial line is a more conservative design practice which will reduce the probability of intermittent bus contention.

## 2. Data Structures

The following sections detail various data structures used by the Intellitrol/VIP rack controller units when communicating with the Terminal Automation System (TAS). Some of the information is more related to actual internal operation of the units but provides a useful insight into the external ("Modbus") interface the units present to the external world.

### 2.1. Vehicle List Data Structure

Scully Truck Identification Modules (TIM's) are mounted on each tank trailer for identification. Each rack controller has the capability of maintaining an on-board (internal to the rack controller unit) Vehicle List. The Vehicle List is kept in EEPROM and persists through power cycles. The Vehicle List need be written only once at installation time, and subsequently updated only when new vehicles are acquired, or old ones removed.

A vehicle serial number is a 12-hexadecimal-digit (48 bits or 6 bytes) number maintained in the TIM (serial numbers cannot be "programmed", they are permanently assigned by the manufacturing process). No two TIMs should ever have the same serial number. While the TIMs are created in a nominally-monotonically-increasing serial number order, for practical purposes they should be simply considered as random 48-bit integers. Both the VIP and Intellitrol rack controllers calculate and store an 8-bit "Dallas™ One-Wire" CRC-8 with each vehicle serial number written into EEPROM. The serial numbers are store in two blocks. The LSB block has the least significant byte store sequentially and the other block has the remaining bytes & crc store sequentially. The LSB block is used to find the pointer to all the serial numbers with that LSB. This increases the lookup speed substantially.

The rack controller reads the TIM serial number when the truck initially connects to the rack controller and will "authorize" (permit) the load if the truck's serial number is found in the Vehicle List. The TAS authorizes a specific vehicle by entering the serial number in the Vehicle List on the rack controller, and unauthorizes it by removing the serial number (writing an entry to 000000000000). No valid TIM will ever bear the serial number 000000000000 or FFFFFFFFFFFFFF (hex).

#### 2.1.1. Intellitrol Vehicle List Features and Operations

The Intellitrol rack controller does a linear search starting at element 0 each time a vehicle hooks up and can typically search 5,000 elements in less than 25 milliseconds. Individual serial numbers are removed from the list by over writing the list element with zero.

The Intellitrol stores bypass key serial numbers is a separate Bypass Key List; the Vehicle List contains only vehicle serial numbers

The *Truck Serial Number* registers, as with the "Current Truck Register" for the VIP as defined above, will contain 0 if no truck is present, or have all bits set (FFFFFFFFFFFF hex) if a truck is present but the TIM is unreadable or not present. The *Truck Serial Number* registers are standard Modbus 16-Bit Control and Data Registers (see below).

The Intellitrol supports the *Read Multiple Vehicles* and *Write Multiple Vehicles* Modbus functions, greatly reducing the time it takes a TAS to read or write a large Vehicle List. *Read Single Vehicle* and *Write Single Vehicle* functions are also supported.

The Intellitrol supports the *CRC Multiple Vehicles* Modbus function to allow the TAS to rapidly validate the vehicle list without having to read the entire list vehicle-by-vehicle.

Since CRC-calculation is a moderately time-consuming operation, and the Intellitrol must guarantee a maximum response to external safety issues (“Overfill” conditions), the Intellitrol limits the maximum number of vehicles to be CRC’ed to 100 (which takes approximately 50 milliseconds). Since the Modbus messages involved are small (8 - 10 bytes), the entire 10000-long Vehicle List can be “verified” by 100 *CRC Multiple Vehicle* messages much faster than the vehicles can be read by the TAS.

The CRC is calculated as if the “n” consecutive vehicle serial numbers were contiguous 6-byte strings arrayed in memory. A “blank” or uninitialized TIM slot is CRC’ed as if it were 6 “00” bytes. An invalid TIM slot/serial number is CRC’ed as is — the serial numbers are not individually CRC’ed as they are extracted from the EEPROM Vehicle List to verify their individual validity.

### 2.1.1.1. Intellitrol Supported Features/Operations

1. Intellitrol Vehicle List has a maximum 10,000 vehicle capacity standard.
2. 48-bit vehicle number (48-bit unsigned integer) with 8 bits internally calculated and verified CRC for error detection.
3. Add a new vehicle by writing a new vehicle number with *Write Single Vehicle* Modbus function and read & write vehicles into the list using standard function codes.
4. Delete (erase) old vehicles by over-writing them with vehicle number 0.
5. Delete (erase) all vehicles by using the *Force Single Bit (Erase Vehicle List)* Modbus function.
6. Bypass keys are kept in separately managed Bypass Key List.
7. Verify vehicle list by reading back individual vehicle numbers with *Read Single Vehicle* Modbus function (or “guess” based on *Vehicle List Size* register).

## 2.2. Bypass Key List Data Structure

Bypass keys are like TIMs (see above) except that they identify people rather than vehicles and are used to bypass otherwise unauthorized or invalid vehicles and “force” the rack control unit to permit.

Bypass key serial numbers behave pretty much identically to vehicle serial numbers; they are 12-hexadecimal-digits (6 bytes or 48 bits) long and are not programmable (the serial number is defined in the manufacture of the key) and thus cannot be copied.

The VIP treats bypass key and vehicle serial numbers identically.

The Intellitrol maintains a separate Bypass Key List (in EEPROM) solely for bypass key serial numbers. The Intellitrol Bypass Key List is normally limited to 32 keys. The Bypass Key List size can be read from the *Bypass Key List Size* register; the number of entries that will fit in the list can then be deduced by dividing the size (in bytes) by 8.

Modbus functions *Write Bypass Keys* and *Read Bypass Keys* are used to write and read back Bypass Key List entries.

## 2.3. VIP Bypass Log Data Structure

The VIP Bypass Log contains a record of VIP bypass events. The VIP records only bypass key usage. Each log entry contains an internally generated CRC to detect EEPROM write and read errors when recording to and reading from the log.

Bypass log entries contains the bypass key serial number, along with the time and date of occurrence. Vehicles lacking an on-board TIM or vehicles with damaged equipment are permitted to load when the attendant inserts

an authorized bypass key into the "bypass" socket mounted on the rack controller unit. This bypasses the normal requirement of a truck having an authorized on-truck TIM and/or having functioning equipment, and closes the rack controller's permit relay. Intellitrol bypass keys are not interchangeable with VIP bypass keys. Each bypass action is logged by the rack controller unit in its EEPROM. The serial numbers of the bypass key and the vehicle's TIM (if possible) are recorded. Some rack controller installations support an expiration date written into TIM RAM and will permit only if the expiration date is good. Other rack controller installations work strictly on the lasered ROM serial number of the TIM. If an expiration date is found in the TIM, it will also be logged. If no expiration date exists, the expiration date fields will return all zeros. If the bypassed vehicle lacks a TIM, the vehicle number will read back all zeros. If a TIM is present but unreadable, the vehicle number reads back all ones (FFFFFFFFFFFF hex).

The *Read Log Element* Modbus function is used to read the VIP Bypass Log.

The Intellitrol does not support the VIP Bypass Log; it uses the Event Log to record all bypass activity ("events").

### 2.3.1. Supported Features and Operations

The TAS can read the log, but not write to it. The log is implemented as a circular list, such that the newest entry over-writes the oldest entry. The TAS reads the log entries one at a time, specifying the entry number of the entry to be read.

1. 32 entry Bypass Log
2. Each log entry records a single transaction, differentiated by its type code.
3. Time and date (start and stop) of occurrence (UNIX format).
4. Pointer to newest log entry.
5. Control bit to erase the entire Customer Log to zeros.
6. Internal CRC to detect log entry errors.

### 2.4. Event Log Data Structure

Rack controller units (Intellitrol and later units) maintain a general purpose on-board "Event Log" in non-volatile EEPROM memory. This event log is used to record "events of interest". For example, the Intellitrol will record initializing the EEPROM itself as an interesting event (useful to ascertain that other potentially useful information was deliberately erased by re-initializing the EEPROM-resident Event Log). Similarly, system resets, hardware errors ("Fault" conditions), and bypass events are all interesting events that are recorded automatically by the Intellitrol rack controller unit.

The general-purpose Event Log is read by using the *Read Event Log* Modbus function.

The Intellitrol nominally uses a 32-event Event Log. The actual number of event entries that the unit's Event Log can support can be derived from reading the *Event Log Size* register and dividing the byte count by 32 (the size of an individual event entry). Alternatively, the TAS can simply loop reading Event Log entries from "0" to "n", stopping when it gets a Modbus Response error code "Illegal Data Address".

The VIP does not support the general Event Log.

All Event Log entries share a common format header, but each distinct event type has its own private (event-specific) data structure. A general Event entry has the following format:

BYTE	NAME	DEFINITION
0	Type	Event type identifier
1	Subtype	Type-specific subtype identifier or value
2 - 3	RepMask	Event “repeat” count/mask
4 - 7	Time	Event time, UNIX 32-bit, Jan 1, 1970 epoch
8 - 29	Info	22 Bytes of Event-specific “private” data
30 - 31	CRC	The CRC-16 of the Info block

Each Event entry consists of 32 bytes of information. The Event type is the first (“Type”) byte.

The second byte is a Type-specific sub-code byte. It may be a bit-mapped set of flags (for example, see “Reset” event), or it may be an 8-bit integer code (for example, see “Hardware Error” event).

The RepMask is an initially-ones (FFFF hex) bit mask used to indicate how many times the “Event” repeated. In order to keep from filling up the Event Log with “redundant” entries, some events can “repeat” themselves — in essence “merging” a bunch of events into a single entry. The RepMask counts how many distinct events have been merged into this entry. Each successive “repeated” event clears the lowest order “1” bit in the RepMask. Once the RepMask is completely zeroed, successive repeated events are simply discarded. A RepMask value of FFFF (hex) indicates the event occurred once, FFFE (hex) indicated the event occurred twice, FF00 (hex) indicates the event has occurred 9 times, while 0000 (hex) indicates that the event has occurred 17 *or more* times within the repeat time window. (This scheme minimizes the wear on EEPROM bits.) The time frame in which successive events are merged depends on the event. For example, Intellitrol “Reset” events use a 4-hour window; all “Reset” events within 4 hours of the “first” event repeat; after 4 hours a “new” event entry will be created (which in its turn may be repeated, etc.).

The Time longword is the UNIX-style 32-bit GMT time (ref epoch of January 1, 1970) of the occurrence of the *first* event — all subsequent repeated events merely “increment” the RepMask counter, no other bytes in the event log entry are changed to in any way reflect further events.

The Info block is the Type-specific event data. Each event type has a different private event data block stored in the event log entry. Only the first event entry logs the private data, subsequent repeated events’ private event data are discarded.

The CRC is the Modbus-style CRC-16 calculated *solely* on the Info block (since the Info block is guaranteed unchanged, while the header may change). The rack controller unit does *not* verify the CRC field when it reads and returns an Event Log entry to the TAS in response to a *Read Event Log* Modbus command. The TAS should validate the CRC before “trusting” and reporting the event information to users. This is explicitly to maximize the potentially useable information available from the rack controller unit in the event of severe system problems (for example, trying to extract information from a rack controller unit that may have been damaged in a terminal explosion/fire).

A blank or uninitialized Event Log entry will read back as all FF (hex) bytes.

### 2.4.1. EEPROM Initialized

[Event Type 01]

Event Type code 01 (hex) records EEPROM “Format/Initialization” events. The EEPROM is initially formatted and initialized by the factory and should not need to be reinitialized once deployed to the field (since reformatting the EEPROM erases *all* previously recorded information residing in the EEPROM).

The EEPROM-Initialized Info block contains the following information:

BYTE	NAME	DEFINITION
0 - 1	HardwareVer	Unit Hardware Revision Level
2 - 3	KernelVer	Firmware Kernel version
4 - 5	ShellVer	Firmware Shell version
6 - 7	Jumpers	Hardware Jumpers (“Config-A” register)
8 - 9	Config2	More configuration (“Config-B” register)
10 - 21		Reserved

### 2.4.2. System Reset

[Event Type 02]

Event Type code 02 (hex) records System Reset events. A System Reset will occur every time power is initially provided to the rack controller, if the board reset button is pressed, if the TAS sends a *Force Single Bit(Hardware Reset)* command, if the unit is in a “Fault” state and is not cleared, or if some other internal error causes the unit to lock up. Other factors (more in the “*Acts of God*” category) like a lightning strike or severe electrostatic discharge (especially if the rack controller’s cover is open for service) can also cause the unit to undergo a reset condition.

System Reset events which occur within a 4-hour window are “repeated” as a single event in the Event Log.

The System Reset Info block contains the following information:

BYTE	NAME	DEFINITION
0 - 1	HardwareVer	Unit Hardware Revision Level
2 - 3	KernelVer	Firmware Kernel version
4 - 5	ShellVer	Firmware Shell version
6 - 7	Jumpers	Hardware Jumpers (“Config-A” register)
8 - 9	Config2	More configuration (“Config-B” register)
10 - 11	EEPROM_Stat	
12 - 21		Reserved

### 2.4.3. Bypass Activity

[Event Type 03]

Event Type code 03 (hex) records bypass activity involving the rack controller. Bypass activity can be either the TAS forcing a bypass action (e.g., *Force Single Bit (Overfill Bypass)*) or a user presenting a Bypass Key to the rack controller in response to a bypassable non-permit condition.

Bypass events are never repeated, although consecutive related bypass actions can be “merged” into one logical bypass event (e.g., if the same bypass key is used to bypass an unauthorized TIM, and 10 seconds later to bypass a Ground Fault, then both bypass actions appear in the Event Log as a “single” bypass event which bypasses two different subsystems). A different bypass key results in distinct bypass events being written to the Event Log. If

more than five minutes elapse between consecutive bypass actions, distinct bypass events are logged. If different trucks are connected (or the truck's serial number is not readable), distinct bypass events are always logged.

The Subtype event byte contains the "class" of bypass conditions in effect; see the *Bypass State* register (low order byte) for the bypass condition bits.

The Bypass Activity Info block contains the following information:

BYTE	NAME	DEFINITION
0 - 5	Key	Bypass Key Serial Number (FFFFFFFFFFFF for TAS)
6 - 11	Truck	Truck Serial Number (if any)
12 - 21		Reserved

### 2.4.4. Hardware Error

[Event Type 04]

Event Type code 04 (hex) records system hardware errors. Hardware errors are typically detected by the resident firmware's self-test diagnostics, and may indicate either a true hardware failure, or a problem external to the unit itself (for example, externally shorting the "Permit" relay contacts would manifest itself as a "Relay Error").

Hardware Events which occur with a 4-hour window are repeated and logged as a single event.

The event Subtype byte identifies the specific hardware error being logged. The Info block format depends on the Subtype identifier. The subtypes are:

BYTE	NAME	DEFINITION
0	EVHDW_CRC	FLASHRAM (Kernel and/or Shell) CRC-16 failure
1	EVHDW_RELAY	Error detected with relay operation
2 - 255		Reserved

#### 2.4.4.1. Firmware CRC-16 Fault

[Hardware Error Subtype 00]

The Firmware CRC failure Info blocks contains the following information:

BYTE	NAME	DEFINITION
0 - 1	KernelGood	Proper Firmware Kernel CRC-16 value
2 - 3	KernelReal	Actual Firmware Kernel CRC-16 value
4 - 5	ShellGood	Proper Firmware Shell CRC-16 value
6 - 7	ShellReal	Actual Firmware Shell CRC-16 value
8 - 21		Reserved

#### 2.4.4.2. Relay Fault

[Hardware Error Subtype 01]

The Relay Fault Info blocks contains the following information:

BYTE	NAME	DEFINITION
0	BackupSt	Backup Relay State Byte
1	MainSt	Main Relay State Byte

BYTE	NAME	DEFINITION
2 - 21		Reserved

### 2.4.5. Voltage Error

[Event Type 05]

Event Type code 05 (hex) records erroneous voltages detected by the firmware self-test diagnostics. Bad voltages may indicate a hardware failure, or an external condition such as a short in the channel/probes cable. Note that the Intellitrol will report voltage faults if a truck is connected to the rack controller when the unit comes out of a reset condition (the probes act as a “short” and drag the channel “Rail” voltages down).

The event Subtype field indicates what type of voltage errors are occurring; see the *Service-A* register (low-order byte) for more details.

The Voltage Error Info blocks contains the following information:

BYTE	NAME	DEFINITION
0 - 1	Raw13	Raw 13-Volt value (millivolts)
2 - 3	RefVolt	Reference Volt value (millivolts)
4 - 5	BiasVolt	Probe Bias Voltage (millivolts)
6 - 7	Ch1	Channel 1 voltage (millivolts)
8 - 9	Ch2	Channel 2 voltage (millivolts)
10 - 11	Ch3	Channel 3 voltage (millivolts)
12 - 13	Ch4	Channel 4 voltage (millivolts)
14 - 15	Ch5	Channel 5 voltage (millivolts)
16 - 17	Ch6	Channel 6 voltage (millivolts)
18 - 19	Ch7	Channel 7 voltage (millivolts)
20 - 21	Ch8	Channel 8 voltage (millivolts)

### 2.4.6. Impact Sensor Tripped

[Event Type 06]

Event Type code 01 (hex) records impacts to the rack controller unit itself. This generally means someone is severely abusing the rack controller, possibly physically damaging the unit.

The Impact Info block contains the following information:

BYTE	NAME	DEFINITION
0 - 5	Key	Bypass Key Serial Number, if any
6 - 11	Truck	Truck Serial Number, if any
12 - 21		Reserved

### 2.4.7. Overfill Info

[Event Type 07]

Event Type code 07 (hex) records Overfills. This event is logged when a probe detects a wet sensor.

The Overfill block contains the following information:



BYTE	NAME	DEFINITION
0	Probe	What kind of probe was wet
1 - 16	Probe State	Report probe state
17 - 21	Truck	Truck Serial Number

### 2.4.8. Maintenance Error

[Event Type 08]

Event Type code 08 (hex) records maintenance errors. This event is logged when high resistance is detected in the truck connection.

The Maintenance block contains the following information:

BYTE	NAME	DEFINITION
0 - 1	Ch 5 High Resistance	Channel 5 resistance is higher than expected
2 - 21		Reserved

### 2.4.9. Reset Info

[Event Type 09]

Event Type code 09 (hex) records reset info. This event is logged during a reset.

The Reset Info block contains the following information:

BYTE	NAME	DEFINITION
0	Config	
1 - 2	StatusA	
3 - 4	StatusB	
5 - 6	Iambroke	
7 - 8	Iamsuffering	
9 - 10	RefVolt	
11	Ground	
12	Main state	
13	Truck state	
14	Acquire state	
15	Probe try state	
16	Five wire state	
17	Two wire state	
18	Backup Relay state	
19	Main Relay state	
20 - 21		Reserved

### 2.4.10. Overfill Info 2

[Event Type 0A]

Event Type code 07 (hex) records Overfill Info. This event is logged when a probe detects a wet sensor.

The Overfill block contains the following information:

BYTE	NAME	DEFINITION
0 - 3	Active	
4 - 11	Probes state	Report probe states
12 - 13	Reason	No-permit code
14 - 15	Low volts	Low voltage
16 - 17	High volts	High voltage
18 - 19	RefVolt	Reference voltage
20	Probe	What kind of probe was wet
21		Reserved

### 2.5. VIP and Intellitrol Force Bit Assignments

The Modbus *Force Single Bit* function is used by the Intellitrol and VIP rack controller units as a general “action”/“reaction” command facility. Actions may be to perform some unary operation such as erasing the vehicle list or may be a Boolean “set/unset” operation such as enabling/disabling Ground Fault Detection operations within the rack controller.

Not all actions are supported by all rack controller types; different rack controller types may behave differently for the same requested action.

Current defined actions are:

#	COMMAND	MEANING	Force Bits
0000	Shutdown	Enable/disable “Shutdown” state	0000 / FF00
0002	Recover	Restore “Normal” state, clear “Shutdown” state	FF00
0003	Erase Vehicle List	Erase the EEPROM-resident Vehicle List	FF00
0004	Erase Log	Erase Bypass/Event Log	FF00
0006	Hardware Reset	Reset the unit.	FF00
0008	Force Overfill Bypass	Bypass current wet/overfill condition.	0000
0009	Force Ground Bypass	Bypass current Ground Fault condition.	0000 / FF00
000A	Enable Ground Fault	Enable/disable Ground Fault Detection operation	0000 / FF00
000F	Enable Deadman Switch	Enable/Disable Deadman Switch operation	0000
0012	Erase Bypass Key List	Erase the EEPROM-resident Bypass Key List	FF00
0013	Erase EEPROM	Erase and Reinit all EEPROM partitions	FF00
0015	Force VIP Bypass	Bypass current unauthorized (VIP) condition	0000
0016	Enable VIP	Enable/Disable VIP operation	0000

### 2.5.1. Forced Shutdown

[Force Code 0000]

Setting the Forced Shutdown bit causes the rack controller unit to unpermit (even though a dry permissible truck is connected) until the unit is reset, or until the RECOVER bit is forced. Unsetting this bit (force data value of “0”) clears “Shutdown” mode and allows the unit to permit normally again.

Forced Shutdown is useful to tell a rack controller (or all units if a broadcast Modbus message) to immediately cease and desist and enter a non-permissive state.

The Intellitrol continues to process Modbus message processing, and to respond to trucks (e.g., detecting probes that are wet, etc.), but the unit will not permit.

### 2.5.2. Forced Recover

[Force Code 0002]

The Forced Recover action resets the “Shutdown” condition and allows the unit to resume normal operations. If the unit was not shut down, this command will have no effect.

### 2.5.3. Erase Vehicle List

[Force Code 0003]

The Erase Vehicle List action reinitializes the Vehicle List. Upon completion, all vehicle serial numbers (and bypass key serial numbers for the VIP) have been erased from the rack controller. Each erasure causes a write which contributes to the wearing out of the EEPROM non-volatile memory.

The Intellitrol will allow an Erase Vehicle List operation only when the unit is in Idle state (no truck is connected). Erasing the Vehicle List may take one or two seconds on the Intellitrol.

This operation may take up to 20 seconds to complete on a VIP.

### 2.5.4. Erase Log

[Force Code 0004]

The Erase Log action reinitializes the onboard EEPROM log. For the VIP, this is the VIP Bypass Log; for the Intellitrol, this is the Event Log. Upon completion of the Erase Log operation, the EEPROM-resident log has been reinitialized — all previous stored log information is lost.

### 2.5.5. Hardware Reset

[Force Code 0006]

The Hardware Reset function is equivalent to pressing the hardware reset button on the microprocessor board. The rack controller unit will undergo a full reset condition. Actual operation of the Reset command depends on the hardware involved.

### 2.5.6. Force Overfill Bypass

[Force Code 0008]

The Force Overfill Bypass action directs the rack controller to “bypass” a current overfill non-permit condition if one exists (otherwise the command is ignored). The Force Overfill Bypass command is functionally equivalent to a bypass key operation for overfill. In particular, the TAS may not bypass any overfill condition that a bypass key could not also bypass (for example, once the bypass timer has expired, neither a bypass key nor a TAS Forced Bypass operation will cause the rack controller to permit again). A logical “1” value enters

bypass state for the overfill condition, while a logical “0” value will exit any overfill bypass state currently in effect (i.e., TAS can “unbypass” an overfill bypass).

The Intellitrol reports bypass key serial number FFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS “bypass key”.

### 2.5.7. Force Ground Bypass

[Force Code 0009]

The Force Ground Bypass action directs the rack controller to “bypass” a current ground-fault non-permit condition if one exists (otherwise the command is ignored). The Force Ground Bypass command is functionally equivalent to a bypass key operation for ground fault bypass. In particular, the TAS may not bypass any ground fault condition that a bypass key could not also bypass (for example, once the bypass timer has expired, neither a bypass key nor a TAS Forced Bypass operation will cause the rack controller to permit again). A logical “1” value enters bypass state for the ground fault condition, while a logical “0” value will exit any ground fault bypass state currently in effect (i.e., TAS can “unbypass” a ground fault bypass).

The Intellitrol reports bypass key serial number FFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS “bypass key”.

### 2.5.8. Enable Ground Fault Detection

[Force Code 000A]

The Enable Ground Fault Detection action is used to enable or disable the Ground Fault subsystem in the rack controller. A logical “1” value enables or turns on the Ground Fault Detection subsystem, while a logical “0” value disables or turns off the Ground Fault Detection subsystem.

Ground Fault Detection is further controlled by both a hardware jumper and a proprietary “Features Enable” password. All conditions must be met (Ground Fault must be enabled (e.g., by this command), the hardware jumper must be installed, and the features password must allow Ground Fault operation) before the Ground Fault Detection subsystem (or “Feature”) will work. Contact your Scully Signal representative for the Features Password for your rack controller(s).

### 2.5.9. Enable Deadman Switch

[Force Code 000F]

The Enable Deadman Switch action is used to enable or disable the Deadman Switch operation in the rack controller. A logical “1” value enables or turns on Deadman Switch operation, while a logical “0” value disables or turns off the Deadman Switch operation.

Deadman Switch operation is further controlled by both a hardware jumper and a proprietary “Features Enable” password. All conditions must be met (Deadman Switch must be enabled (e.g., by this command), the Deadman-Enable hardware jumper must be installed, and the features password must allow Deadman Switch operation) before the Deadman Switch (or “Feature”) will work. Contact your Scully Signal representative for the Features Password for your rack controller(s).

### 2.5.10. Erase Bypass Key List

[Force Code 0012]

The Erase Bypass Key List action erases (reinitializes) the Bypass Key List in EEPROM. A logical “1” value erases the Bypass Key List, while a logical “0” value is ignored.

### 2.5.11. Erase EEPROM

[Force Code 0013]

The Erase EEPROM action completely reinitializes the entire EEPROM non-volatile memory storage. All EEPROM “partitions” are erased and reinitialized to their default values.

For the Intellitrol, this command is only allowed if the rack controller is in Idle state (no truck is connected).

### 2.5.12. Force VIP Bypass

[Force Code 0015]

The Force VIP Bypass action directs the rack controller to “bypass” a current unauthorized (VIP) non-permit condition if one exists (otherwise the command is ignored). The Force VIP Bypass command is functionally equivalent to a VIP bypass key operation. In particular, the TAS may not bypass any VIP condition that a bypass key could not also bypass (for example, once the bypass timer has expired, neither a bypass key nor a TAS Forced Bypass operation will cause the rack controller to permit again). A logical “1” value enters bypass state for the VIP condition (i.e., the current truck is “authorized”), while a logical “0” value will exit any VIP bypass state currently in effect (i.e., TAS can “unbypass” a VIP bypass).

The Intellitrol reports bypass key serial number FFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS “bypass key”.

### 2.5.13. Enable VIP

[Force Code 0016]

The Enable VIP action is used to enable or disable the VIP subsystem in the rack controller. A logical “1” value enables or turns on the VIP operations, while a logical “0” value disables or turns off the VIP operations.

VIP operation is further controlled by both a hardware jumper and a proprietary “Features Enable” password. All conditions must be met (VIP must be enabled (e.g., by this command), the VIP-Enable hardware jumper must be installed, and the features password must allow VIP operation) before VIP operations will work. Contact your Scully Signal representative for the Features Password for your rack controller(s).

The various Vehicle List-related Modbus commands (e.g., *Write Single Vehicle*) continue to work regardless of the VIP enable state.

## 2.6. Input Status Bits

The Scully rack controller units (Intellitrol and VIP) maintain status information readily available via the *Input Status Bits*. The Input Status Bits are more-or-less divided into two 16-bit groupings. The first sixteen bits are the primary operating status bits and “in a glance” tell the status of the rack controller unit. The second sixteen bits are problem/error bits, serving as the first tier of error/non-permit information.

The Intellitrol presents the Input Status Bits through the Modbus *Read Input Status Bits* command, and as the *Status-A/Status-B* pair of Modbus 16-Bit Control and Data Registers. (The information is the same but repackaged to save a Modbus message transaction.)

The VIP supports only the Modbus *Read Input Status Bits* message.

## Intellitrol/VIP Modbus RTU Protocol Specification

The 32 Input Status Bits are:

INPUT BIT	CONDITION	MEANING IF READ BACK AS 1
0	FAULT	Fault (red LED blinking on the VIP or the service LED blinking on the Intellitrol).
1	TRUCK PRESENT	Truck is seen to be connected to the unit by the firmware. The truck is considered present while in bypass.
2	TRUCK TALK	Communications established with TIM or IntelliCheck.
3	TRUCK VALID	At least one Truck Serial Number is Authorized
4	BYPASS	Rack controller unit is in a bypass state
5	IDLE	Rack controller unit is idle (and non-permissive).
6	PERMITTING	Rack controller is permissive.
7	NON-PERMISSIVE	Rack controller is non-permissive (but bypassable).
8	DEBUG JUMPER	Debug Jumper is enabled
9	CH 5 HIGH RESISTANCE	Channel 5 resistance is higher than expected
10		Reserved
11		Reserved
12	DEADMAN OK	Deadman switch is closed.
13	DIODE GND	Diode ground is enabled
14	RESISTIVE GND	Resistive ground is enabled
15	INTELLICHECK	Connected to Intellicheck
16		Reserved
17	BAD EEPROM	Problems with EEPROM
18	ADC TIME-OUT	This bit is set if the ADC times-out during a conversion.
19	SHELL CRC ERROR	Checksum failure occurred in shell program code.
20	CLOCK ERROR	On board Dallas™ real time clock/calendar failure occurred.
21	BAD CPU	Stuck bits in CPU registers or stuck U1 I/O pins.
22	TRUCK	Truck is connected
23	KERNEL CRC ERROR	Checksum failure in Kernel firmware program code occurred.
24	VOLTAGE ERROR	Problems with one or more onboard voltage levels.
25		Reserved
26	TIM DATA LINE FAULT	Error communicating with TIM
27		Reserved
28	GROUND FAULT	The Ground Fault Detection subsystem cannot verify proper ground (earth) connection on the truck.
29	SPECIAL OPS MODE	The rack controller is in “Special Operations” mode
30	SHUTDOWN	The rack controller is “Shutdown” and will not permit, although it continues to otherwise operate normally.
31	RELAY ERROR	Problems detected with permit relay(s).
32 - 65535		Reserved

### 2.6.1. Fault

#### [Input Status Bit 00]

The Fault bit indicates that the rack controller needs “service”. Typically, this means that the rack controller firmware has detected a software or hardware problem that is keeping the unit from normal and safe operation. Certain hardware failures may also cause a “Fault” condition that may or may not be reportable (via the Modbus) as the “Fault” status bit.

In general, if the Fault bit is set, then one or more of the “problem” status bits (16 - 31) will also be set to indicate what the Fault condition is. Further information may be obtained (guided by Input Status Bits 16 - 31) from the various diagnostic and other informational registers.

The Intellitrol will not permit while there is a Fault condition detected (unless the Debug jumper is in place, defeating most safety checks).

For the Intellitrol, if the Fault status bit is set, then the front panel “Service” LED should also be blinking. If either the main or backup (or both) microprocessors should fail, then the Service LED will blink. If communications are lost between microcontrollers the service LED will blink.

The Intellitrol is periodically running self-test diagnostics. These diagnostics will both catch problems that arise while the unit is operating and will also detect when a problem “fixes itself”. For example, if the permit relay contacts are shorted together (resulting in a false permit), then the Fault bit will be set, and the Service LED will start blinking. If the short then goes away, the unit will notice that the permission output is no longer shorted and will clear the Fault condition and allow normal truck/permit operations.

### 2.6.2. Truck Present

#### [Input Status Bit 01]

The Truck Present bit indicates that the rack controller has detected “something” on one or more of the probe channels, or on the ground pin. Typically, when a truck connects, one or more of the probe channels will no longer appear to be an open circuit (the probes draw power from the rack controller, and the rack controller can detect this power drain), or the pin-9 Ground may appear grounded or a TIM can be read. If a properly configured and operating truck connects, the rack controller should quickly be able to determine the probe type and configuration and go into the normal “Active” mode of operation.

For the Intellitrol, either the Idle or the Truck Present bit should always (and mutually-exclusively) be set. Note that simply shorting any of the probe channels together, or to ground can cause the Intellitrol to set Truck Present. but as soon as the short is removed, the unit should return to Idle after a short period. (Note also that one or more of the channel status LEDs may be blinking to indicate a short or grounded channel.)

### 2.6.3. Truck Talk

#### [Input Status Bit 02]

The Truck Talk status bit indicates that the rack controller unit has successfully established communications with a TIM.

For the Intellitrol, Truck Talk set means that the *Truck Serial Number* register is meaningful (e.g., a TIM was sensed; a truck serial number of FFFFFFFF (hex) would then indicate that the TIM or the communications was faulty). Simply sensing overfill probes does not set Truck Talk.

### 2.6.4. Truck Valid

#### [Input Status Bit 03]

The Truck Valid bit means that At least one Truck Serial Number is Authorized.

### 2.6.5. Bypass

[Input Status Bit 04]

The rack controller has one or more bypass conditions currently in effect.

The *Bypass State* register contains the current active bypass information.

### 2.6.6. Idle

[Input Status Bit 05]

The Idle status bit indicates that the Intellitrol currently has no truck connected, and is not permitting, but is operating normally.

### 2.6.7. Permitting

[Input Status Bit 06]

The Permitting status bit indicates the rack controller unit is actively “permitting”, that is, all criteria required of a “good truck” has been met. These criteria can include vehicle authorization by serial number, vehicle ground safety verification in addition to dry and functional overfill probes. Alternatively, the control unit have one or more of the above criteria “bypassed” and permitting despite failure to satisfy one or more of the configured criteria.

In the Intellitrol, if the green Main “Permissive” Status Indicator is lit, the unit is permitting due to all permit criteria satisfied; if the red “Non Permit” LED is blinking, then the unit is in an overfill bypass condition; if the green Permit LED is blinking then the unit is in some other (Ground Fault, Unauthorized, etc.) bypass condition. The *Bypass State* register can be read to determine the current Intellitrol bypass condition, and the *Non-Permit Reasons* register can be read to determine why the unit is not in Permitting state.

### 2.6.8. Non-Permissive

[Input Status Bit 07]

The Non-Permissive status bit indicates that the rack controller unit is not permitting.

For the Intellitrol, the *Non-Permit Reasons* register can be read to determine what is preventing the rack controller from entering the permissive state.

### 2.6.9. Debug Jumper

[Input Status Bit 08]

The Debug Jumper status bit indicates that the debug jumper is enabled.

### 2.6.10. CH 5 High Resistance

[Input Status Bit 09]

The CH 5 High Resistance status bit indicates that the Channel 5 resistance is higher than expected.

### 2.6.11. Deadman OK

[Input Status Bit 12]

The Deadman OK status bit indicates that the unit is configured to require the Deadman Switch, and that the switch appears to be properly closed.



### 2.6.12. Diode GND

[Input Status Bit 13]

The Diode GND status bit indicates that diode ground is enabled.

### 2.6.13. Resistive GND

[Input Status Bit 14]

The Resistive GND status bit indicates that resistive ground is enabled.

### 2.6.14. Intellicheck

[Input Status Bit 15]

The Intellicheck status bit indicates that the unit is connected to an Intellicheck.

### 2.6.15. Bad EEPROM

[Input Status Bit 17]

The Bad EEPROM status bit indicates that the rack controller has detected one or more errors in dealing with the on-board EEPROM non-volatile memory store. Typically, this is not a fault condition, the unit continues to operate in a normal and safe manner. It may indicate a simple “data error” retrieving a truck serial number from the Vehicle List, for example (requiring bypassing VIP authorization for that truck serial number).

For the Intellitrol, details on the current EEPROM status may be obtained from the *EEPROM Status* register.

### 2.6.16. ADC Time-out

[Input Status Bit 18]

The ADC Time-out status bit indicates that the ADC has timed-out during a conversion.

### 2.6.17. Shell CRC Error

[Input Status Bit 19]

The Shell CRC Error status bit means that the firmware has detected a bad firmware program image. This typically requires reloading the Shell firmware in order to fix the problem, although simply resetting (or, possibly, power-cycling) the rack controller may “cure” this problem, especially if originally caused by, e.g., an electrostatic discharge or other “weird” power glitch.

The Intellitrol records both correct and actual CRCs as part of the hardware error event in the Event Log.

### 2.6.18. Clock Error

[Input Status Bit 20]

The Clock Error status bit indicates that the rack controller cannot correctly read the on-board (battery-backed where legal) real-time calendar clock. A clock error does not constitute a Fault condition. FM & CSA approved Intellitrol models have a battery backed clock which is factory set. CENELEC approved Intellitrol models have a capacitor backed clock which may need to be set if the power is removed from the unit for more than 4 days.

The clock may be unreadable, set to date before 1992 or after 2050, or not keeping time. Setting the clock via Modbus will clear this error if the clock becomes readable

For the Intellitrol, the *Clock Status* register contains more-detailed information on the current clock status.

### 2.6.19. Bad CPU

[Input Status Bit 21]

The Bad CPU status bit indicates that there are Stuck bits in CPU registers or stuck U1 I/O pins.

### 2.6.20. Truck

[Input Status Bit 22]

The Truck status bit indicates that there is a truck connected.

### 2.6.21. Kernel CRC Error

[Input Status Bit 23]

The Kernel CRC Error status bit indicates that a checksum failure in Kernel firmware program code has occurred.

### 2.6.22. Voltage Error

[Input Status Bit 24]

The Voltage Error status bit indicates that the rack controller firmware self-test diagnostics has detected one or more bad voltages in the unit.

For the Intellitrol, the *Service-A* register indicates more specifically which voltage family is in error. The Voltage Diagnostic registers may be read to individually examine and/or report the separate voltage levels.

The Intellitrol records the erroneous voltages and conditions as part of the hardware error event in the Event Log.

### 2.6.23. TIM Data Line Fault

[Input Status Bit 26]

The TIM Data Line Fault status bit indicates that there is an error communicating with TIM.

### 2.6.24. Ground Fault

[Input Status Bit 28]

The rack controller unit has detected a Ground Fault condition.

For an Intellitrol unit, the Ground Fault Detection subsystem cannot detect a Ground Bolt (or simple wired-to-ground for European “100 Ohm” configurations), or possibly that the Ground Fault Detect circuitry itself has failed. More detailed information is available in the *Ground Status* register.

### 2.6.25. Special Ops Mode

[Input Status Bit 29]

The Special Ops Mode status bit indicates that the rack controller is running in Special Operations mode.

In Special Operations Mode, normal truck-related actions are completely ignored. The rack controller will never permit (cannot even be bypassed) in Special Operations mode. The rack controller will continue to respond to Modbus commands. The hardware must be rejumped (and reset) to exit Special Operations mode and return to normal operations.

For the Intellitrol, this means that a special hardware jumper has been installed, directing the unit to perform non-standard operations. The two currently defined “Special” operations are to erase the Bypass Key List, and to add the current bypass key to the Bypass Key List.

The hardware jumper on the CPU board must be removed and the control unit reset to exit Special Operations mode and to return to normal operation.

### 2.6.26. Shutdown

#### [Input Status Bit 30]

The Shutdown status bit indicates that the rack controller is shut down by command. This means that the rack controller will not permit and cannot be bypassed. The rack controller otherwise operates normally, responding to trucks connecting, and to Modbus commands.

The Intellitrol gives no “front panel” indications that the unit is Shutdown. The Intellitrol can only be Shut down by a Modbus command. The Intellitrol can only be “Recovered” from the Shutdown mode by a Modbus command which clears the Shutdown status.

### 2.6.27. Relay Error

#### [Input Status Bit 31]

The Relay Error status bit indicates that the rack controller firmware has detected a problem with the Main permit relay. It may be shorted or never close. For the Intellitrol, the current status of both the main and backup permit relays may be read via the *Relay State* register.

## 2.7. Output Status Bits

The Output Status Bits are generally not “unit status” so much as they are “unit representation” bits, generally to indicate the rack controller’s appearance and in particular the state of specific front panel LEDs that may or may not be immediately deducible from the *Input Status Bits*.

The Output Status Bits, as with the Input Status Bits, are more-or-less divided into two 16-bit groupings. The first sixteen bits are the LED status bits. The second sixteen bits are the VIP S-record counter.

The Intellitrol presents the Output Status Bits through the Modbus *Read Output Status Bits* command, and as the *Status-O/Status-P* pair of Modbus 16-Bit Control and Data Registers. (The information is the same but repackaged to save a Modbus message transaction.)

The VIP supports only the Modbus *Read Output Status Bits* message.

The 32 Output Status Bits are:

OUTPUT BIT	OUTPUT NAME	MEANING IF READ BACK AS 1
00		Reserved
01		Reserved
02		Reserved
03		Reserved
04	STSO_5WIRE_PULSE1	Five-wire-optic output pulse “right now”
05	STSO_5WIRE_PULSE2	Five-wire-optic output pulse “recently”
06	STSO_5WIRE_ECHO1	Five-wire-optic echo return pulse “right now”
07	STSO_5WIRE_ECHO2	Five-wire-optic echo return pulse “recently”
08	STSO_COMM_VEHICLE1	Communicating with vehicle “right now”
09	STSO_COMM_VEHICLE2	Communicated with vehicle “recently”

OUTPUT BIT	OUTPUT NAME	MEANING IF READ BACK AS 1
10 - 15		Reserved

### 2.7.1. Five-Wire-Optic Output Pulse

[Output Status Bits 04 - 05]

The Five-Wire-Optic Output Pulse status bits indicate the that rack controller is actively transmitting output pulses for Five-Wire optic probes. The STSO\_5WIRE\_PULSE1 bit indicates that the controller is “just now” transmitting an output pulse. This bit quickly (within roughly 100 milliseconds) “shifts” to become the STSO\_5WIRE\_PULSE2 bit which is eventually (within roughly 1000 milliseconds) cleared.

The Intellitrol “polls” Five-Wire optic probes roughly every 50 milliseconds, so if the Intellitrol is connected to a set of Five-Wire optic probes, one of these two bits will be set. The TAS may interpret either bit being set to mean that the “Optic Pulse Out” LED is on.

### 2.7.2. Five-Wire-Optic Echo Pulse

[Output Status Bits 06 - 07]

The Five-Wire-Optic Echo Pulse status bits indicate the that rack controller is in 5 wire optic mode and receiving return pulses from the 5 wire probes in response to transmitting 5 wire optic probe output pulses. The STSO\_5WIRE\_ECHO1 bit indicates that the controller has “just received” a return pulse. This bit quickly (within roughly 100 milliseconds) “shifts” to become the STSO\_5WIRE\_ECHO2 bit which is eventually (within roughly 1000 milliseconds) cleared.

The Intellitrol produces 5 wire optic probes roughly every 50 milliseconds, so if the Intellitrol is connected to a set of “dry” Five-Wire optic probes, it should be receiving return pulses roughly every 50 milliseconds, and one of these two bits will be set. The TAS may interpret either bit being set to mean that the “Optic Pulse In” LED is on.

### 2.7.3. Vehicle Communications

[Output Status Bits 08 - 09]

The Vehicle Communications status bits indicate the that rack controller is actively communicating with the attached truck’s onboard TIM. The STSO\_COMM\_VEHICLE1 bit indicates that the controller has “just now” communicated with the truck. This bit quickly (within roughly 100 milliseconds) “shifts” to become the STSO\_COMM\_VEHICLE2 bit which is eventually (within roughly 1000 milliseconds) cleared.

TAS may interpret either bit being set to mean that the “Vehicle Communications” LED is on (although more likely is that it “was on, briefly”).

### 2.7.4. S-Record Counter

[Output Status Bits 16 - 31]

This 16-bit S-Record Counter contains the number of error free S-Records received during boot-load. Output bit 16 is bit 0 of the count and output bit 31 is bit 15 of the 16-bit counter. The S-Record count is embedded in the S0 record after the last S-Record is sent, the unit does a comparison of the number of S-Records received to the S-Record count. If these agree, the boot-load was successful, and the unit will run the Shell.

The Intellitrol does not support S-Record downloading in version 1.0 and will return these bits as all 0s.

### 2.8. 16-Bit Control and Data Registers

Both the Intellitrol and the VIP utilize the Modbus “Register Set” (Modbus message codes 03, 06, and 10) as a mechanism for the passing of information to and from the employed Terminal Automation System. Mostly, the registers are used by the Intellitrol and for presenting data to the TAS, but some of the registers are writable, that is, the TAS can “write” a new value to the “register” which the Intellitrol can act upon (for example, “date and time” registers).

The Intellitrol provides many times more information and introduces a sub-setting or partitioning of the Modbus Register “Address Space” in order to more-logically group similar registers together. Specifically, the register mapping used by the Intellitrol groups all “normal” status/information registers together such that the TAS can, in one Modbus read operation, obtain the Intellitrol’s complete “normal operating status”.

#	GROUP NAME	DESCRIPTION
0000 - 000F	VIP/Configuration registers	This is the original VIP contiguous register set. The Intellitrol supports many of these registers, duplicates some, and doesn’t support others (reads will return 0).
0010 - 001F	Standard Registers	
0020 - 002F	Static Unit Configuration	This block of registers contains the unit’s “static” configuration, which is that information not typically subject to change during normal operation. For example, the unit’s serial number, or the setting of hardware jumpers.
0030 - 005F	Voltage References	
0060 - 006F	States	
0070 - 008F	Settable Configuration	
0090 - 009F		Reserved
00A0 - 00BF	EEPROM Information	These registers provide detailed EEPROM configuration information such as the size of the different EEPROM partitions (which allows the TAS to determine supported sizes of, e.g., the vehicle list size, number of event log entries, etc.).
00C0 - 00DF		Reserved
00E0 - 00FF	Errors	
0100 - 01FF	Dynamic Unit State	These registers group together the unit’s “normal” or “most useful” status and data registers in order to minimize the number of Modbus transactions needed to read the information.
0200 - 02FF		Reserved
0300 - 04FF	TIM memory	
0500 - FFFF		Reserved

All register values are presented to the network in “Big Endian” form. Multiple-register values similarly are “big-endian” format (i.e., more-significant bytes come first). Certain “subblocks” of registers present a “string” of data bytes, again in “big-endian” format (i.e., “first” byte is high-order byte of “first” register, “second” byte is low-order byte of “first” register, etc.).

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0000	VIP Year	R	—	Non-Volatile	
0001	VIP Month	R	—	Non-Volatile	
0002	VIP Day	R	—	Non-Volatile	

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0003	VIP Hour	R	—	Non-Volatile	
0004	VIP Minute	R	—	Non-Volatile	
0005	Shell version	R	—	Non-Volatile	
0006	VIP Serial Number	R	—	Non-Volatile	
0007	Latest Log pointer	R	—	Non-Volatile	
0008	Wait for TAS Delay	R / W	0 - 60	Non-Volatile	Time (seconds) to wait for TAS
0009	Bypass Active Time	R / W	120 - 3600	Non-Volatile	Bypass time-out (seconds)
000A	Terminal Number	R / W	0 - 9999	Non-Volatile	Terminal ID (e.g., for DateStamp)
000B	Modbus Minimum Response Time	R / W	0 - 1000	Non-Volatile	Modbus command response wait (milliseconds)
000C	Shell Checksum	R	—	Permanent	Checksum of uploaded shell program (hex)
000E	Mode Control	R / W	0 - 5	Dynamic	Authorization mode control
000F	Kernel Version	R	—	Non-Volatile	
0012	Scully type / model	R	—	Non-Volatile	
0020 - 0023	Unit Serial Number	R	—	On-board Dallas™ TOD clock	48-bit Serial number derived from on-board Dallas™ clock
0024	Hardware Revision Level	R	—	Permanent	Hardware Version Number
0025	Config-A	R	—	Hardware-Based	Assorted Hardware jumpers, etc.
0026	Config-B	R	—	Non-Volatile	Assorted Software “jumpers”, etc.
0027	Config-C	R	—	Non-Volatile	
0028	Config-D	R	—	Non-Volatile	
0029	RAM size (in KB)	R	—	Non-Volatile	
002A	FLASHRAM size (in KB)	R	—	Non-Volatile	
002B	EEPROM size (in KB)	R	—	Non-Volatile	
002C	Max ModBus message size	R	—	Non-Volatile	
002D	Number of 5 wire probes	R	—	Non-Volatile	
002E	Enable Modbus Features	R	—	Non-Volatile	
0030	Reference volt	R	—	Dynamic	
0031	Raw Power Supply (13 V)	R	—	Dynamic	
0032	Probe Bias Voltage	R	—	Dynamic	
0033	5-Wire Optic Pulse	R	—	Dynamic	
0038	Channel 0 Noise Voltage	R	—	Dynamic	
0039	Channel 1 Noise Voltage	R	—	Dynamic	
003A	Channel 2 Noise Voltage	R	—	Dynamic	

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
003B	Channel 3 Noise Voltage	R	—	Dynamic	
003C	Channel 4 Noise Voltage	R	—	Dynamic	
003D	Channel 5 Noise Voltage	R	—	Dynamic	
003E	Channel 6 Noise Voltage	R	—	Dynamic	
003F	Channel 7 Noise Voltage	R	—	Dynamic	
0040	Channel 0 10-Volt Rail	R	—	Dynamic	
0041	Channel 1 10-Volt Rail	R	—	Dynamic	
0042	Channel 2 10-Volt Rail	R	—	Dynamic	
0043	Channel 3 10-Volt Rail	R	—	Dynamic	
0044	Channel 4 10-Volt Rail	R	—	Dynamic	
0045	Channel 5 10-Volt Rail	R	—	Dynamic	
0046	Channel 6 10-Volt Rail	R	—	Dynamic	
0047	Channel 7 10-Volt Rail	R	—	Dynamic	
0048	Channel 0 20-Volt Rail	R	—	Dynamic	
0049	Channel 1 20-Volt Rail	R	—	Dynamic	
004A	Channel 2 20-Volt Rail	R	—	Dynamic	
004B	Channel 3 20-Volt Rail	R	—	Dynamic	
004C	Channel 4 20-Volt Rail	R	—	Dynamic	
004D	Channel 5 20-Volt Rail	R	—	Dynamic	
004E	Channel 6 20-Volt Rail	R	—	Dynamic	
004F	Channel 7 20-Volt Rail	R	—	Dynamic	
0050	Channel 0 Current Voltage	R	—	Dynamic	
0051	Channel 1 Current Voltage	R	—	Dynamic	
0052	Channel 2 Current Voltage	R	—	Dynamic	
0053	Channel 3 Current Voltage	R	—	Dynamic	

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0054	Channel 4 Current Voltage	R	—	Dynamic	
0055	Channel 5 Current Voltage	R	—	Dynamic	
0056	Channel 6 Current Voltage	R	—	Dynamic	
0057	Channel 7 Current Voltage	R	—	Dynamic	
0060	Read Clock Status	R	—	Dynamic	
0061	Relay Status	R	—	Dynamic	
0062	EEPROM status	R	—	Dynamic	
0064	"Acquire" state	R	—	Dynamic	
0065	"Probe-Try" state	R	—	Dynamic	
0066	Optic 5-wire state	R	—	Dynamic	
0067	Optic/Thermal 2-wire state	R	—	Dynamic	
0068	VIP status (inc DateStamp)	R	—	Dynamic	
0069	Service "A" flags	R	—	Dynamic	
006A	Service "B" flags	R	—	Dynamic	
006B	VIP status (fuel_type_fails)	R	—	Dynamic	
006C	VIP status (badvipdscode)	R	—	Dynamic	
006D	Ground status	R	—	Dynamic	
006E	VIP status (badvipflag)	R	—	Dynamic	
006F	VIP status (CERTIFICATE_ds_fails)	R	—	Dynamic	
0070	Shorts test active	R / W	0 - 1	Non-Volatile	
0071 - 0734	Debug Pulse on Code	R / W	0 - 255	Non-Volatile	
0075 - 0078	Debug Pulse on Failed Code	R / W	0 - 255	Non-Volatile	
0079 – 007A	Software Feature Enable Code	R / W	0 - 255	Non-Volatile	
007B	TIM read enable	R / W	0 - 1	Non-Volatile	
007C	Resistive Ground reference value	R / W	0 - 3	Non-Volatile	
007E	Enable Ground Display	R / W	0 - 1	Non-Volatile	
007F	5-Wire Compartment Count Display Time	R / W	0 - 255	Non-Volatile	
0080	Active Deadman Enabled	R / W	0 - 255	Non-Volatile	



## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0081	Active Deadman Max open time	R / W	0 - 30	Non-Volatile	
0082	Active Deadman Max close time	R / W	0 - 600	Non-Volatile	
0083	Active Deadman Warning time	R / W	0 - 60	Non-Volatile	
0084	Software Feature Enable Unload Terminal	R / W	0 - 255	Non-Volatile	
0085	SuperTim max unload time	R / W	0 - 65535	Non-Volatile	
0086	SuperTim Certificate date enable mask	R / W	0 - 31	Non-Volatile	
0087	Software Feature Enable compartment count check	R / W	0 - 255	Non-Volatile	
0088	SuperTim fuel type check mask	R / W	0 - 255	Non-Volatile	
0089	Software Feature Enable auto write fuel type flag	R / W	0 - 255	Non-Volatile	
008A - 008B	SuperTim default fuel type	R / W	0 - 65535	Non-Volatile	
00A0 - 00A1	EEPROM base address	R	—	Non-Volatile	
00A2	Home block size	R	—	Non-Volatile	
00A3	Home block offset	R	—	Non-Volatile	
00A4	Boot block size	R	—	Non-Volatile	
00A5	Boot block offset	R	—	Non-Volatile	
00A6	Crash block size	R	—	Non-Volatile	
00A7	Crash block offset	R	—	Non-Volatile	
00A8	System block size	R	—	Non-Volatile	
00A9	System block offset	R	—	Non-Volatile	
00AA	Event log block size	R	—	Non-Volatile	
00AB	Event log block offset	R	—	Non-Volatile	
00AC	Bypass key block size	R	—	Non-Volatile	
00AD	Bypass key block offset	R	—	Non-Volatile	
00AE	Truck ID block size	R	—	Non-Volatile	
00AF	Truck ID block offset	R	—	Non-Volatile	
00E0	Soft COMM_ID ground errs	R	—	Dynamic	
00E1	Hard COMM_ID ground errs	R	—	Dynamic	

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0100 - 0101	Date & Time	R / W	—	On-board Dallas™ TOD clock	32-Bit current date & time in UNIX format, reference epoch 1-Jan-1970.
0102 - 0103	Event Elapsed Time	R	—	Dynamic	32-Bit elapsed time (milliseconds)
0104	Status-A	R	—	Dynamic	Dynamic status bits 0 - 15 (same as “Input Status” bits)
0105	Status-B	R	—	Dynamic	Dynamic status bits 16 - 31 (same as “Input Status” bits)
0106	Status-O	R	—	Dynamic	“Output Status” bits 0 - 15
0107	Status-P	R	—	Dynamic	“Output Status” bits 16 - 31
0108	Main State	R	—	Dynamic	“Main” state engine
0109	Truck-Type State	R	—	Dynamic	Overall Truck/Tank/Probes state
010A - 010C	Truck Serial Number	R	—	Dynamic	48-Bit Truck ID/serial number
010D - 0114	Probes State Bytes	R	—	Dynamic	Individual tank/probe state (one byte per probe/channel)
0115	Bypass State	R	—	Dynamic	Bypass status flags
0116 - 0118	Bypass Key Number	R	—	Dynamic	48-Bit Bypass Key serial number
0119	Bypass Time	R	—	Dynamic	Time unit has been in bypass mode (seconds)
011A	Non-Permit Reasons	R	—	Dynamic	Flags indicating why non-idle unit is non-permissive
011B	Latest Log pointer	R	—	Dynamic	
011C	Optic 2 wire Threshold	R / W	0 - 4375	Non-Volatile	
011D	Hysteresis	R / W	0 - 700	Non-Volatile	
011E	Thermistor Threshold	R / W	0 - 3675	Non-Volatile	
0120	Number of truck compartments	R	—	Dynamic	
0121	Stop logging dome out events	R / W	0 - 1	Dynamic	
0122	TIM info logged	R	—	Dynamic	
0123	TIM size	R	—	Dynamic	
0124	Super TIM code	R	—	Dynamic	
0125	log data state	R	—	Dynamic	
0126	compare volts	R	—	Dynamic	Channel 5 voltage from truck connection

### 2.8.1. Shell Version

#### [Register 005]

Reading the Shell Version register returns the version number of the running “Shell” program. The “Shell” program (or firmware) is the main control program of the rack controller.

The version number is a three-part number in the format MajorVersion in top 4 bits + MinorVersion in next four bits + EditVersion in low 8 bits. For example, version 1.3.17 would be 1317 hex.

The “MajorVersion” number indicates the primary release or feature set. The “MinorVersion” indicates the maintenance (or bug fix) level of release within the MajorVersion release, and resets with each major release. The “EditVersion” resets with each major release (increase in the “MajorVersion” number).

### 2.8.2. Wait for TAS Delay

[Register 008]

This Read/Write register causes the rack controller to wait a specified time (0 to 60 seconds) before attempting to authorize a truck. This delay gives the TAS time to read the truck register, do a lookup on a truck number, and decide to explicitly authorize or unauthorize the truck (see the VIP Mode Register). If the TAS should go down or otherwise not exert explicit “VIP Mode” control, the rack controller will take over and consult the onboard vehicle list after the wait for TAS time has expired.

A wait time of zero disables this feature. While in TAS Delay, the VIP “Standby” LED will flash.

Intellitrols are shipped with the wait for TAS delay set to zero seconds.

### 2.8.3. Bypass Active Time

[Register 009]

This Read/Write register sets the desired bypass active time-out.

In the Intellitrol units, this register determines the maximum amount of time the unit can remain bypassed from the presentation of a Bypass Authorizer (Bypass key) or from a TAS issued bypass command. Bypass ends when the truck departs. The Intellitrol will allow the full 16-bits’ worth of timer value (FFFF hex), or about 18 hours as the maximum bypass active time.

Intellitrols are shipped with Bypass Active Time set to 3600 seconds (1 hour).

### 2.8.4. Terminal Number

[Register 00A]

This Read/Write register is the desired terminal identification number (0-9999). Intellitrols are shipped with the terminal number set to 0.

### 2.8.5. Modbus Minimum Response Delay

[Register 00B]

This Read/Write register is the desired Modbus minimum response delay time (milliseconds 0 to 1024). This minimum response time is the time that the unit will be guaranteed to delay before initiating transmission of any Modbus Response message. A response time of 100 will force the rack controller to wait at least 100 milliseconds before transmitting the response to any Modbus query message. A response time of 0 allows the rack controller to respond as soon as it can (typically 10 milliseconds). In some cases (especially some older RS-485/RS-232 converters), the rack controller may reply before the bus master can turn the half duplex line around, causing the bus master to miss the rack controller response. In this case, increase the response time. The smallest delay time commensurate with reliable communications should be used (and it may take a lot of field experimentation to determine this delay value) in order to maximize Modbus communications throughput.

Intellitrol are shipped with the minimum response time set to 100 milliseconds.

### 2.8.6. Shell Checksum

[Register 00C]

Computed checksum of the shell program.

### 2.8.7. VIP Mode Control

[Register 00E]

The Read/Write VIP Mode Control register allows the TAS to exert explicit control over VIP operation. The VIP subsystem can only be in one of the modes below. Changing the mode cancels the previous mode. This register only affects VIP operation (e.g. setting this register will not cause the Intellitrol to permit if a wet truck is attached).

This register is normally used when the TAS validates a vehicle with its database. During the time the unit is in TAS Delay, the VIP “Standby” LED will flash. When the TAS writes to the VIP Mode Control register, the Standby LED will stop flashing and the VIP subsystem will respond to which mode was selected. If the wait for TAS timer times out before the TAS sets the mode register, the VIP subsystem will determine truck authorization from its onboard vehicle list, until the TAS does finally changes the mode.

When the TAS is not involved in the validation process, but only for vehicle list maintenance standpoint, the Local Operation Mode is selected. This mode is the power up and reset default mode.

The current mode is indicated when the register is read. As an example, setting the mode to Remote Unauthorization when a vehicle is not at the rack, will cause a read back indicating local operation, because a Remote Unauthorization ceases when the truck departs.

#### 2.8.7.1. VIP Mode Control Bit Assignments

MODE	OPERATION	MEANING
0	LOCAL OPERATION	Normal operation without any Modbus override. Authorizes trucks from the TIM vehicle list.
1	REMOTE BYPASS	Will cause the unit to bypass the VIP function and act as if a Bypass Authorizer had just been presented to the bypass port on the unit. When the truck departs or bypass timer expires, the unit resumes local operation mode. The bypass will be logged.
2	REMOTE UNAUTHORIZATION	Will cause the unit to Unauthorize the present vehicle (VIP Unbypass). When the truck departs, the unit resumes local operation mode.
3	REMOTE AUTHORIZATION	This is equivalent to setting the remote bypass mode for the VIP function, with the exception that no entry of the bypass will be made in the log. When truck departs, the unit resumes local operation mode.
4	PERM. AUTHORIZATION	Will cause the unit to authorize (VIP override) until the mode is changed via Modbus or reset.
5	READ VIP ONLY	
7- 65535		Reserved, return exception response message.

### 2.8.8. Unit Serial Number

[Registers 020-023]

The Unit Serial Number register(s) contain the rack controller unit’s 64-bit serial number.

### 2.8.9. Hardware Revision Level

[Register 024]

The Hardware Revision Level reflects the “Rev Level” of the hardware of the rack controller unit. The hardware revision level is in the same format as the shell version.

### 2.8.10. Config-A

[Register 025]

The Config-A register is used to read the setting of the “features” hardware jumpers. Changing any of these jumpers will set the unit into a “fault” condition until the unit is reset (which means that generally the unit must be reset before the change-in-jumpers will take affect).

Some of the hardware jumper-controlled features are further controlled by software. This means that the Intellitrol firmware may selectively “disable” a feature “enabled” by hardware jumper. Further, some of these software-controllable features are master-controlled by a “Features Enable Password” that Scully uses to sell different models based on software features. The “VIP”, “Ground Fault Detection”, “Vapor Flow”, and “Deadman Switch” operations (or “subsystems”) are in this group. For one of these features to be enabled, *all three* controls must be active -- the hardware jumper must be in place, the software enable must be on, and the unit’s Features Password must authorize the feature.

MASK (hex)	NAME	MEANING
0001		Reserved.
0002	ENA_TRUCK_HERE	Enable the “Truck Here” logic.
0004	ENA_VIP	Enable the VIP subsystem code.
0008	ENA_GROUND	Enable the Ground-Fault-Detection subsystem code.
0010	ENA_ADD_1_KEY	Boot up in “Special Operations” mode for manually adding Bypass Authorizers to the internal Bypass Authorizer serial number list stored in EEPROM.
0020	ENA_ERASE_KEYS	Boot up in “Special Operations” mode and erase the internal EEPROM Bypass authorizer List.
0040	ENA_DEADMAN	Enable the Deadman Switch subsystem code.
0080	ENA_VAPOR_FLOW	Enable the Vapor Flow sensing subsystem.
0100	CFGA_8COMPARTMENT	The “8-compartment” jumper is installed, the Intellitrol will utilize all 8 probe channels (2 wire mode).
0200	CFGA_EURO8VOLT	The “CENELEC” voltage-operation jumper is installed.
0400	CFGA_OHMGND	The “Resistive Short” Ground Verification jumper is installed. VIP functionality is implicitly disabled.
7800		Reserved.
8000		Reserved

### 2.8.11. Config-B

[Register 026]

The Config-B register is the “software” analog of the hardware jumpers register. This register allows the TAS system to inquire of the unit what software features are enabled. Each bit in the Config-B register parallels the corresponding bit in the Config-A hardware-jumpers register.

## Intellitrol/VIP Modbus RTU Protocol Specification

Some of these software-controllable features are further controlled by a “Features Enable Password”. VIP, Ground, Vapor Flow, and Deadman Switch operation are in this group. For one of these “features” to be enabled, *all three* controls must be active -- the hardware jumper must be in place, the software enable must be on, and the unit’s Features Password must authorize the feature.

MASK (hex)	NAME	MEANING
0001		Reserved
0002	ENA_TRUCK_HERE	Enable the “Truck Here” logic.
0004	ENA_VIP	Enable VIP operation.
0008	ENA_GROUND	Enable Ground-Check operation.
0010	ENA_ADD_1_KEY	Boot up in “Special Operations” mode for manually adding bypass keys to the internal Bypass Authorizers to the internal Bypass Authorizer List.
0020	ENA_ERASE_KEYS	Boot up in “Special Operations” mode and erase the internal Bypass Authorization List.
0040	ENA_DEADMAN	Enable Deadman Switch operation.
0080	ENA_VAPOR_FLOW	Enable Vapor Flow sensing operation.
0100	CFGA_8COMPARTMENT	The 6/8 compartment jumper is installed, the unit is in 8 channels location. The unit is sensing 8 probe channels (compartments).
0200	CFGA_EURO8VOLT	The “CENELEC” voltage operation jumper is installed.
0400	CFGA_100OHMGND	The “Resistive Short” Ground Verification mode jumper is installed. VIP functionality is implicitly disabled.
F800		Reserved.

### 2.8.12. UNIX Date/Time

### [Registers 100-101]

These Read/Write registers allow setting of date and time, and verification of the battery powered real time clock/calendar. The two registers taken together implement a single 32-bit unsigned integer value. Setting the date/time starts the clock running. Both UNIX Date/Time registers should be set at once with one query. The date and time are stored in UNIX format as the number of seconds since the “epoch” midnight January 1, 1970 GMT. (Other “epoch” baseline times exist; another popular one is January 1, 1900.) Valid dates & times are settable within the years 1992 through 2050.

The 32-bit time is nominally specified as “GMT” (Greenwich Mean Time) or “UCT” (Universal Coordinated time), and the Intellitrol units are shipped from the factory pre-initialized with a running (if battery-backed) clock to maintain the time. The TAS system may choose to implement a “different” time base (e.g., local time), if it is consistent in the usage and interpretation of the time base, and if the chosen values appear correct to the rack controller firmware. GMT-based is strongly encouraged. The Intelliview program utilizes GMT-based values.

The registers are designated as follows:

MSB	LSB	MSB	LSB
Register 100		Register 101	
Date 3 MSB	Date 2	Date 1	Date 0 LSB

**NOTE:** Always verify time and date after any change.

## 2.8.13. Event Elapsed Time

[Registers 102-103]

The Event Elapsed Time registers return the elapsed time (in milliseconds) since the beginning of the current “event” (i.e., since the current truck initially connected to the rack controller unit). The elapsed timer is a 32-bit millisecond counter.

MSB	LSB	MSB	LSB
Register 102		Register 103	
ms 3 MSB	ms 2	ms 1	ms 0 LSB

## 2.8.14. Status-A

[Register 104]

The Status-A register “maps” the first sixteen Input Status Bits, making them available along with other dynamic rack controller unit “status” information in a single *Modbus Read Registers* protocol message, thus helping minimize Modbus message traffic.

Input Status Bits 0 - 15 are the primary “active unit status” flags and serve as the starting point for a TAS to determine the current status of a rack controller unit.

The Intellitrol Status-A register is bitmapped as follows:

MASK (hex)	NAME	DEFINITION
0001	STSA_FAULT	The unit is in a Fault condition.
0002	STSA_TRK_PRESENT	Something is connected to the unit.
0004	STSA_TRK_TALK	Communications have been established on the Pin 9 truck “bus” — e.g., a TIM
0008	STSA_TRK_VALID	Truck (TIM) serial number is Authorized.
0010	STSA_BYPASS	One or more bypass conditions are in effect.
0020	STSA_IDLE	The unit is idle; no truck is connected.
0040	STSA_PERMIT	The unit is actively permitting.
0080	STSA_BYPASSABLE	One or more bypassable conditions exist.
0100	STSA_DEBUG	Debug Jumper is enabled
0200	CH5_HIGH_RESISTANCE	Channel 5 resistance is higher than expected

<b>MASK</b> (hex)	<b>NAME</b>	<b>DEFINITION</b>
0400		Reserved.
0800		Reserved.
1000	STSA_DED_CLOSED	The Deadman Switch is closed.
2000	STSA_DIODE_GND	Diode ground is enabled
4000	STSA_RESISTIVE_GND	Resistive ground is enabled
8000	STSA_INTELLICHECK	Connected to Intellicheck

### 2.8.15. Status-B

[Register 105]

The Status-B register “maps” the second sixteen Input Status Bits, making them available along with other dynamic rack controller unit “status” information in a single Modbus Read Registers protocol message, thus helping minimize Modbus message traffic.

Input Status Bits 16 - 31 are primarily “error/fault status” flags. In general, if the STSA\_FAULT Status bit is set, then Status-B is the starting point to determine what the unit fault is. From Status-B, a TAS might proceed to any of several other registers for more detailed diagnostic information. For example, if STSA\_FAULT is set, then the TAS might examine the Service-A flags to set, then the TAS might examine further to see that the STSB\_BAD\_CLOCK bit is set. The Intellitrol Status-B register is bit-mapped as follows:

<b>MASK</b> (hex)	<b>NAME</b>	<b>DEFINITION</b>
0001		Reserved.
0002	STSB_ERR_EEPROM	One or more errors have occurred with the EEPROM.
0004	STSB_ADC_TIMEOUT	ADC has Timed-out
0008	STSB_CRC_SHELL	The Shell CRC-16 value is bad.
0010	STSB_BAD_CLOCK	Problems with on-board real-time clock/calendar
0020	STSB_BAD_CPU	Stuck bits in CPU registers or stuck U1 I/O pins.
0040	STSB_TRUCK	Truck is connected
0080	STSB_CRC_KERNEL	Checksum failure in Kernel firmware program code occurred.
0100	STSB_ERR_VOLTAGE	One or more voltages are at fault. See Service-A.
0200		Reserved.
0400	STSB_GRND_FAULT	The Ground Fault Detection subsystem cannot verify proper ground (earth) connection on the truck.
0800		Reserved.
1000	STSB_GRND_FAULT	The Ground Fault Detect circuit has failed.
2000	STSB_SPECIAL	The unit is in “Special Operations” mode.
4000	STSB_SHUTDOWN	The unit is in “Shutdown” mode.
8000	STSB_BAD_RELAY	One or more relay errors have been detected.

### 2.8.16. Truck Serial Number

[Registers 10A-10C]

The Truck Serial Number registers contain the currently connected truck’s “serial number”, if any. This register is “meaningful” only when the Status-A STSA\_TRK\_PRESENT bit is set. A value of 0 means there is no truck



## Intellitrol/VIP Modbus RTU Protocol Specification

serial number, a value of FFFFFFFF means the serial number has not yet been determined. Any other value is the truck's "authorization" serial number (e.g., TIM).

The layout of the truck's serial number in the three Truck Serial Number registers is:

MSB	LSB	MSB	LSB	MSB	LSB
Register 10A		Register10B		Register 10C	
Truck S/N MSB	Truck S/N LSB	Truck S/N MSB	Truck S/N LSB	Truck S/N MSB	Truck S/N LSB

### 2.8.17. Probe State

[Registers 10D-114]

The Probe State real-time registers contain the current state of each of the 16 finite state machines used to determine what state a "logical" probe (as opposed to a physical "channel") is in. Probes 9 - 16 are meaningful only for 5-Wire-Optic style probes.

The layout is as follows:

MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB
Register 10D		Register 10E		Register 10F		Register 110		Register 111		Register 112		Register 113		Register 114	
Probe 1 State	Probe 2 State	Probe 3 State	Probe 4 State	Probe 5 State	Probe 6 State	Probe 7 State	Probe 8 State	Probe 9 State	Probe 10 State	Probe 11 State	Probe 12 State	Probe 13 State	Probe 14 State	Probe 15 State	Probe 16 State

#### 2.8.17.1. Probe States

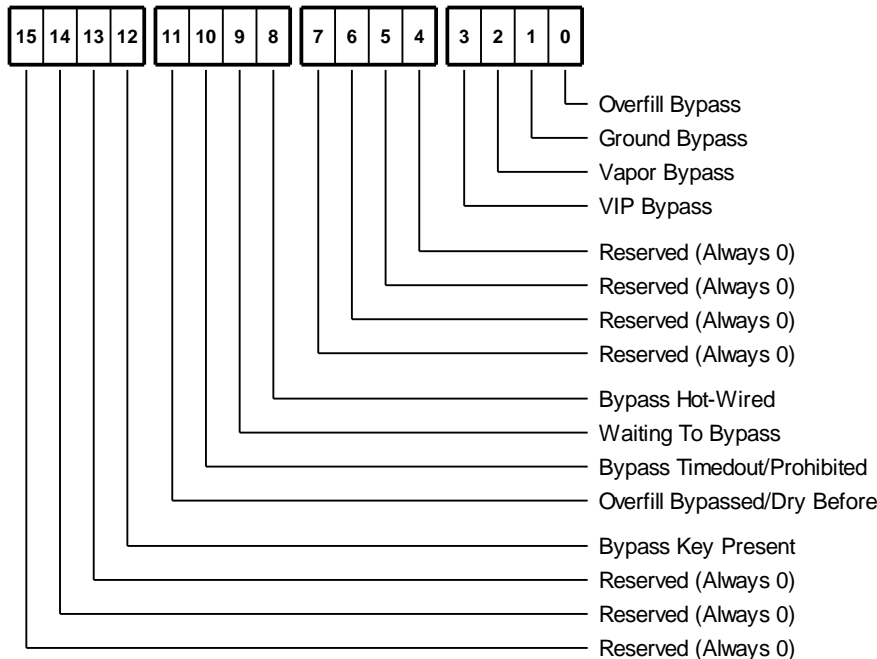
The probe states can be one of the following:

VALUE	NAME	DEFINITION
0	Unknown	Initial unknown state.
1	Wet	Truck probe wet (not oscillating).
2	Dry	Truck probe dry (oscillating).
3		Reserved
4		Reserved
5	Open	Bad probe on truck, channel/probe open
10		Reserved
11	Grounded	Channel/Probe shorted to ground
12	Shorted	Channel/Probe shorted to another probe or other power source
13 - 65535		Reserved

**2.8.18. Bypass State**

**[Register 115]**

The Bypass State real-time register contains the current bypass state flags. The Bypass State register is only meaningful when the Status-A STSA\_BYPASS flag is set. The Bypass State register is bit-mapped as follows:



The low-order byte contains the mask of bypassable conditions that are currently bypassed. The high-order byte contains related-to-bypass flags:

- The “Bypass Hot-Wired” flag indicates that the rack controller has determined that the bypass key has been “hardwired” and will be ignored.
- The “Waiting to Bypass” flag indicates that the rack controller is in the initial connection overfill wait timer. This timer prevents premature attempts at bypassing probes that simply haven’t warmed up yet, or otherwise settled down. As of this writing, the timer is set at 60 seconds for thermistor probes (you cannot bypass a “wet” thermistor probe until at least 60 seconds have elapsed after initial truck connection) and 20 seconds for all optic probe configurations.
- The “Bypass Timed out/Prohibited” flag indicates that the rack controller Bypass Timer has expired. The unit cannot be further bypassed; the truck must disconnect.
- The “Overfill Bypassed/Dry Before” flag indicates that the unit was successfully “permissive” for long enough that the unit will not allow an overfill bypass. The truck must disconnect.
- The “Bypass Key Present” flag indicates that a bypass key is currently present and being read.

**2.8.19. Bypass Serial Number**

**[Registers 116-118]**

The Bypass Serial Number register contains the last bypass key serial number used to put the unit into active bypass state. The Bypass Serial Number register is only meaningful when the Status-A STSA\_BYPASS flag is set. The bypass serial number layout is as follows:

MSB	LSB	MSB	LSB	MSB	LSB
Register 116		Register 117		Register 118	
Bypass S/N MSB	Bypass S/N LSB	Bypass S/N MSB	Bypass S/N LSB	Bypass S/N MSB	Bypass S/N LSB

2.8.20. Bypass Time

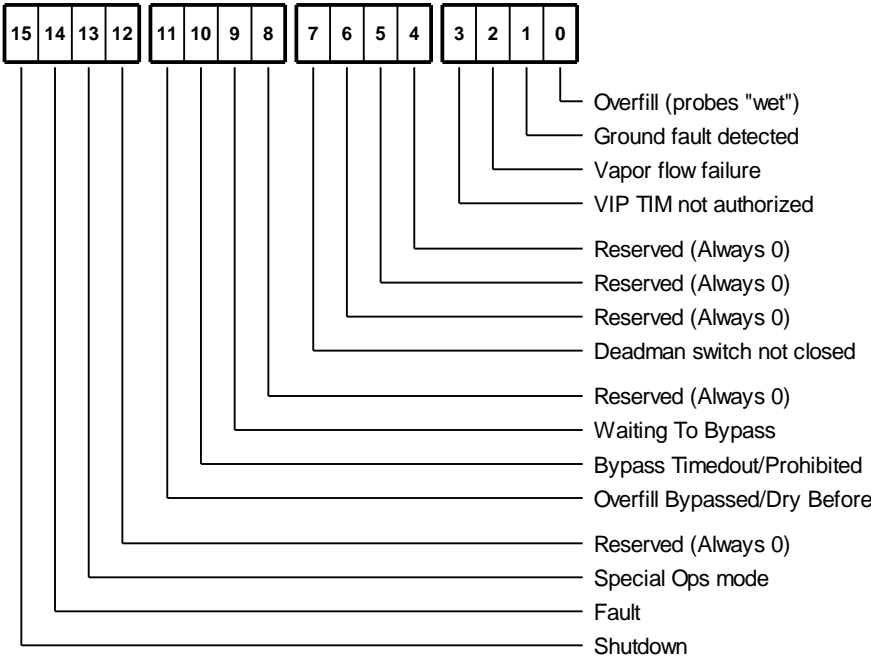
[Register 119]

The Bypass Time register contains the current elapsed time (in seconds) that the rack controller has been in a bypass state. The Bypass Time register is only meaningful when the Status-A STSA\_BYPASS flag is set.

2.8.21. Non-Permit Reasons

[Register 11A]

The Non-Permit Reasons dynamic register contains flags detailing why the rack controller unit is currently not permitting. The Non-Permit Reasons register is only meaningful when the Status-A STSA\_TRK\_PRESENT flag is set. The Non-Permit Reasons register is bit-mapped as follows:



The low-order byte contains the mask of bypassable conditions that are currently not bypassed (except that the Deadman Switch subsystem is not bypassable). The high-order byte are flags as follows:

- The “Waiting to Bypass” flag indicates that the rack controller is in the initial connection overfill wait timer. This timer prevents premature attempts at bypassing probes that simply haven’t warmed up yet, or otherwise settled down. As of this writing, the timer is set at 60 seconds for thermistor probes (you cannot bypass a “wet” thermistor probe until at least 60 seconds have elapsed after initial truck connection) and 20 seconds for all optic probe configurations.

## Intellitrol/VIP Modbus RTU Protocol Specification

- The “Bypass Timed out/Prohibited” flag indicates that the rack controller Bypass Timer has expired (the unit has been in bypass condition too long). The unit cannot be further bypassed; the truck must disconnect.
- The “Overfill Bypassed/Dry Before” flag indicates that the unit was successfully “permissive” for long enough that the unit will not allow an overfill bypass. The truck must disconnect.
- The “Special Ops mode” flag indicates that the rack controller unit was booted up in “Special Operations” mode and thus by definition will *never* permit.
- The “Fault” flag indicates that the rack controller unit is in a “Fault” condition, and thus cannot permit.
- The “Shutdown” flag indicates that the unit is in “Shutdown by TAS Command” mode, and thus will *not* permit, even though the unit otherwise appears to be operational (seems to respond to trucks connecting, dry probes, etc.)

### 2.9 TIM Builder Info

#	REGISTER NAME	DIR	DESCRIPTION
01	CARRIER NAME	R	Name of the carrier company
02	CARRIER ADDRESS	R	Address of the carrier company
03	CONTRACT NUMBER	R	Carriers contract number
04	OPERATING SERVICE	R	Operating service’s name
05	DRIVER ID	R	ID of the driver
06	ALLOWABLE VOL COMPARTMENT 1	R	Allowable volume of compartment 1
07	ALLOWABLE VOLUME COMPARTMENT 2	R	Allowable volume of compartment 2
08	ALLOWABLE VOLUME COMPARTMENT 3	R	Allowable volume of compartment 3
09	ALLOWABLE VOLUME COMPARTMENT 4	R	Allowable volume of compartment 4
0A	ALLOWABLE VOLUME COMPARTMENT 5	R	Allowable volume of compartment 5
0B	ALLOWABLE VOLUME COMPARTMENT 6	R	Allowable volume of compartment 6
0C	ALLOWABLE VOLUME COMPARTMENT 7	R	Allowable volume of compartment 7
0D	ALLOWABLE VOLUME COMPARTMENT 8	R	Allowable volume of compartment 8
0E	ALLOWABLE VOLUME COMPARTMENT 9	R	Allowable volume of compartment 9
0F	ALLOWABLE VOLUME COMPARTMENT 10	R	Allowable volume of compartment 10
10	ALLOWABLE VOLUME COMPARTMENT 11	R	Allowable volume of compartment 11
11	ALLOWABLE VOLUME COMPARTMENT 12	R	Allowable volume of compartment 12
12	ALLOWABLE VOLUME COMPARTMENT 13	R	Allowable volume of compartment 13

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	DESCRIPTION
13	ALLOWABLE VOLUME COMPARTMENT 14	R	Allowable volume of compartment 14
14	ALLOWABLE VOLUME COMPARTMENT 15	R	Allowable volume of compartment 15
15	ALLOWABLE VOLUME COMPARTMENT 16	R	Allowable volume of compartment 16
16	VAP TIGHT CERTIFICATE TYPE	R	Certificate type for the vapor tightness certificate. For vapor tightness type this value will always be 1.
17	VAP TIGHT CERTIFICATE DATE	R	Expiration date for the vapor tightness certificate
18	VAP TIGHT CERTIFICATE NUMBER	R	Vapor tightness certificate number
19	SAFE PASS CERTIFICATE TYPE	R	Safe loading pass certificate type. For safe loading pass type this value will always be 2.
1A	SAFE PASS CERTIFICATE DATE	R	Safe loading pass certificate expiration date
1B	SAFE PASS CERTIFICATE NUMBER	R	Safe loading pass certificate number
1C	CERTIFICATE 3 TYPE	R	Certificate 3 type
1D	CERTIFICATE 3 DATE	R	Certificate 3 expiration date
1E	CERTIFICATE 3 NUMBER	R	Certificate 3 number
1F	CERTIFICATE 4 TYPE	R	Certificate 4 type
20	CERTIFICATE 4 DATE	R	Certificate 4 expiration date
21	CERTIFICATE 4 NUMBER	R	Certificate 4 number
22	CERTIFICATE 5 TYPE	R	Certificate 5 type
23	CERTIFICATE 5 DATE	R	Certificate 5 expiration date
24	CERTIFICATE 5 NUMBER	R	Certificate 5 number
25	TABLE VALID	R	A flag to indicate if the TIM data is valid
26	TABLE REVISION	R	The revision number of the TIM data
27	ALTERNATE TIM ID VALID	R	A flag to indicate the alternate TIM ID is a valid ID. If this flag is a 0x33 the alternate TIM ID is valid.
28	ALTERNATE TIM ID	R	Alternate TIM ID
29	NUMBER OF COMPARTMENTS	R	Number of compartments on the truck
2A	COMPARTMENT VOLUME UNITS	R	Volume units for the compartments
2B	TRAILER ID NUMBER	R	Trailer ID number
2C	COMPARTMENT CONFIG	R	Current configuration of the compartments
2D	VAPOR INTERLOCK TYPE	R	Trucks vapor interlock type
2E	COMPARTMENT 1 TYPES ALLOWED	R	Product types allowed in compartment 1
2F	COMPARTMENT 2 TYPES ALLOWED	R	Product types allowed in compartment 2
30	COMPARTMENT 3 TYPES ALLOWED	R	Product types allowed in compartment 3
31	COMPARTMENT 4 TYPES ALLOWED	R	Product types allowed in compartment 4

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	DESCRIPTION
32	COMPARTMENT 5 TYPES ALLOWED	R	Product types allowed in compartment 5
33	COMPARTMENT 6 TYPES ALLOWED	R	Product types allowed in compartment 6
34	COMPARTMENT 7 TYPES ALLOWED	R	Product types allowed in compartment 7
35	COMPARTMENT 8 TYPES ALLOWED	R	Product types allowed in compartment 8
36	COMPARTMENT 9 TYPES ALLOWED	R	Product types allowed in compartment 9
37	COMPARTMENT 10 TYPES ALLOWED	R	Product types allowed in compartment 10
38	COMPARTMENT 11 TYPES ALLOWED	R	Product types allowed in compartment 11
39	COMPARTMENT 12 TYPES ALLOWED	R	Product types allowed in compartment 12
3A	COMPARTMENT 13 TYPES ALLOWED	R	Product types allowed in compartment 13
3B	COMPARTMENT 14 TYPES ALLOWED	R	Product types allowed in compartment 14
3C	COMPARTMENT 15 TYPES ALLOWED	R	Product types allowed in compartment 15
3D	COMPARTMENT 16 TYPES ALLOWED	R	Product types allowed in compartment 16
3E	MAX LOADING TEMPERATURE	R	Maximum loading temperature
3F	TEMPERATURE UNITS	R	Temperature units
40	COMPARTMENT 1 TYPE LOADED	R / W	Fuel type loaded into compartment 1
41	COMPARTMENT 1 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 1
42	COMPARTMENT 1 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 1
43	COMPARTMENT 2 TYPE LOADED	R / W	Fuel type loaded into compartment 2
44	COMPARTMENT 2 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 2
45	COMPARTMENT 2 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 2
46	COMPARTMENT 3 TYPE LOADED	R / W	Fuel type loaded into compartment 3
47	COMPARTMENT 3 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 3
48	COMPARTMENT 3 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 3
49	COMPARTMENT 4 TYPE LOADED	R / W	Fuel type loaded into compartment 4
4A	COMPARTMENT 4 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 4

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	DESCRIPTION
4B	COMPARTMENT 4 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 4
4C	COMPARTMENT 5 TYPE LOADED	R / W	Fuel type loaded into compartment 5
4D	COMPARTMENT 5 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 5
4E	COMPARTMENT 5 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 5
4F	COMPARTMENT 6 TYPE LOADED	R / W	Fuel type loaded into compartment 6
50	COMPARTMENT 6 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 6
51	COMPARTMENT 6 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 6
52	COMPARTMENT 7 TYPE LOADED	R / W	Fuel type loaded into compartment 7
53	COMPARTMENT 7 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 7
54	COMPARTMENT 7 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 7
55	COMPARTMENT 8 TYPE LOADED	R / W	Fuel type loaded into compartment 8
56	COMPARTMENT 8 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 8
57	COMPARTMENT 8 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 8
58	COMPARTMENT 9 TYPE LOADED	R / W	Fuel type loaded into compartment 9
59	COMPARTMENT 9 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 9
5A	COMPARTMENT 9 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 9
5B	COMPARTMENT 10 TYPE LOADED	R / W	Fuel type loaded into compartment 10
5C	COMPARTMENT 10 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 10
5D	COMPARTMENT 10 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 10
5E	COMPARTMENT 11 TYPE LOADED	R / W	Fuel type loaded into compartment 11
5F	COMPARTMENT 11 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 11
60	COMPARTMENT 11 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 11
61	COMPARTMENT 12 TYPE LOADED	R / W	Fuel type loaded into compartment 12
62	COMPARTMENT 12 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 12
63	COMPARTMENT 12 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 12

## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	DESCRIPTION
64	COMPARTMENT 13 TYPE LOADED	R / W	Fuel type loaded into compartment 13
65	COMPARTMENT 13 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 13
66	COMPARTMENT 13 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 13
67	COMPARTMENT 14 TYPE LOADED	R / W	Fuel type loaded into compartment 14
68	COMPARTMENT 14 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 14
69	COMPARTMENT 14 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 14
6A	COMPARTMENT 15 TYPE LOADED	R / W	Fuel type loaded into compartment 15
6B	COMPARTMENT 15 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 15
6C	COMPARTMENT 15 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 15
6D	COMPARTMENT 16 TYPE LOADED	R / W	Fuel type loaded into compartment 16
6E	COMPARTMENT 16 BATCH ID LOADED	R / W	Fuel batch date code loaded into compartment 16
6F	COMPARTMENT 16 VOLUME LOADED	R / W	Volume of fuel loaded into compartment 16
70	TERMINAL NAME	R / W	Name of the terminal for the last load
71	TERMINAL ADDRESS	R / W	Address of the terminal for the last load
72	TERMINAL GANTRY NUMBER	R / W	Gantry number for the last load
73	FAULT LOG 1	R	Fault entry 1
74	FAULT LOG 2	R	Fault entry 2
75	FAULT LOG 3	R	Fault entry 3
76	FAULT LOG 4	R	Fault entry 4
77	FAULT LOG 5	R	Fault entry 5
78	SERVICE CENTER NAME	R	Name of the last service center
79	SERVICE CENTER ADDRESS	R	Address of the last service center
7A	BUILDER NAME	R	Name of the company that built the truck
7B	BUILDER ADDRESS	R	Address of the company that built the truck
7C	TRUCK SERIAL NUMBER	R	Trucks serial number
7D	TRUCK VIN	R	Trucks vehicle identification number
7E	TRUCK BUILD DATE	R	Date the truck was built
7F	TRUCK WEIGHT UNITS	R	Weight unit for the vehicle weight
80	TRUCK GROSS VEHICLE WEIGHT	R	Weight of the vehicle
81	INTELLICHECK TYPE	R	Is an Intellicheck being used
82	OVERFILL SENSOR TYPE	R	Are overfill sensors being used
83	RETAINED SENSOR TYPE	R	Are retained sensors being used



## Intellitrol/VIP Modbus RTU Protocol Specification

#	REGISTER NAME	DIR	DESCRIPTION
84	COMPARTMENT 1 BUILD VOLUME	R	Compartment 1 tank volume
85	COMPARTMENT 2 BUILD VOLUME	R	Compartment 2 tank volume
86	COMPARTMENT 3 BUILD VOLUME	R	Compartment 3 tank volume
87	COMPARTMENT 4 BUILD VOLUME	R	Compartment 4 tank volume
88	COMPARTMENT 5 BUILD VOLUME	R	Compartment 5 tank volume
89	COMPARTMENT 6 BUILD VOLUME	R	Compartment 6 tank volume
8A	COMPARTMENT 7 BUILD VOLUME	R	Compartment 7 tank volume
8B	COMPARTMENT 8 BUILD VOLUME	R	Compartment 8 tank volume
8C	COMPARTMENT 9 BUILD VOLUME	R	Compartment 9 tank volume
8D	COMPARTMENT 10 BUILD VOLUME	R	Compartment 10 tank volume
8E	COMPARTMENT 11 BUILD VOLUME	R	Compartment 11 tank volume
8F	COMPARTMENT 12 BUILD VOLUME	R	Compartment 12 tank volume
90	COMPARTMENT 13 BUILD VOLUME	R	Compartment 13 tank volume
91	COMPARTMENT 14 BUILD VOLUME	R	Compartment 14 tank volume
92	COMPARTMENT 15 BUILD VOLUME	R	Compartment 15 tank volume
93	COMPARTMENT 16 BUILD VOLUME	R	Compartment 16 tank volume
94	SCULLY SENSORS	R	Are Scully sensors installed
95	TANK MODEL NUMBER	R	Model number of the tank
96	MAX WORKING PRESSURE	R	Maximum tank pressure
97	ALLOWABLE WORKING PRESSURE	R	Working tank pressure
98	PRESSURE UNITS	R	Pressure units
99	BULKHEADS	R	Number of bulkheads on a truck
9A	TANK PROFILE	R	Profile description of truck tanks
9B	OVERFILL SENSOR 1 LENGTH	R	Length of sensor 1
9C	OVERFILL SENSOR 2 LENGTH	R	Length of sensor 2
9D	OVERFILL SENSOR 3 LENGTH	R	Length of sensor 3
9E	OVERFILL SENSOR 4 LENGTH	R	Length of sensor 4

## Intellitrol/VIP Modbus RTU Protocol Specification

---

#	REGISTER NAME	DIR	DESCRIPTION
9F	OVERFILL SENSOR 5 LENGTH	R	Length of sensor 5
A0	OVERFILL SENSOR 6 LENGTH	R	Length of sensor 6
A1	OVERFILL SENSOR 7 LENGTH	R	Length of sensor 7
A2	OVERFILL SENSOR 8 LENGTH	R	Length of sensor 8
A3	OVERFILL SENSOR 9 LENGTH	R	Length of sensor 9
A4	OVERFILL SENSOR 10 LENGTH	R	Length of sensor 10
A5	OVERFILL SENSOR 11 LENGTH	R	Length of sensor 11
A6	OVERFILL SENSOR 12 LENGTH	R	Length of sensor 12
A7	OVERFILL SENSOR 13 LENGTH	R	Length of sensor 13
A8	OVERFILL SENSOR 14 LENGTH	R	Length of sensor 14
A9	OVERFILL SENSOR 15 LENGTH	R	Length of sensor 15
AA	OVERFILL SENSOR 16 LENGTH	R	Length of sensor 16
AB	OVERFILL SENSOR 17 LENGTH	R	Length of sensor 17
AC	OVERFILL SENSOR 18 LENGTH	R	Length of sensor 18
AD	OVERFILL SENSOR 19 LENGTH	R	Length of sensor 19
AE	OVERFILL SENSOR 20 LENGTH	R	Length of sensor 20
AF	OVERFILL SENSOR 21 LENGTH	R	Length of sensor 21
B0	OVERFILL SENSOR 22 LENGTH	R	Length of sensor 22
B1	OVERFILL SENSOR 23 LENGTH	R	Length of sensor 23
B2	OVERFILL SENSOR 24 LENGTH	R	Length of sensor 24

### 3. Modbus Functions

Unless specified otherwise all numerical values in this section are hexadecimal. In accordance with Modbus custom, all addresses are referenced to zero. Commands generally cannot be longer than 64 bytes including the CRC. The first element of the vehicle list is element zero and the first bit of bit fields is bit zero. Here is a summary of the supported Modbus commands:

CODE (Hex)	FUNCTION
01	Read Output Status
02	Read Input Status Bits
03	Read Multiple 16-bit Registers
05	Force (Set) Single Bit
06	Write Single 16-bit Register
41	Write Single Vehicle
42	Read Single Vehicle
43	Read VIP Log
44	Write Password
45	Write Company ID
46	Write Multiple Vehicles
47	Read Multiple Vehicles
48	Backup Functions
49	Read TRL Log
4A	CRC-16 Multiple Vehicles
4B	Write Bypass Keys
4C	Read Bypass Keys
4D	Write "Features Enable" Password
4E	Read EEPROM
4F	Write EEPROM
50	Report Compartment Volume
51	Read TIM Area
52	Write TIM Area
53	Read Builder Info
54	Write Builder Info
55	Read Third Party
56	Write Third Party
57	Read Builder Area
58	Write Builder Area
59	Insert Vehicle
5A	Remove Vehicle
5B	Read Number of Probes
5C	Alternate between updated 1.7.0 ADC table and original pre 1.7.0 ADC table for determining number of connected probes
5D	Get Current ADC Table

## 3.1. Read Output Status

## Function Code 01

These boot load status bits reflect the status of the VIP and Intellitrol outputs and match the LED's on the units. Bits 16 through 31 report the S-Record count after a boot-load. Function 01 works in accordance with Modicon Manual page 3-3. We recommend reading the LED status using Function 02. Typically Function 01 would be used to verify a successful PROGRAM LOAD, since the LED status information is also available using Function 02.

### 3.1.1. Example Query Message Reading Status Bits 0 - 2

Read Output Status bits 0 thru 2 Query Message

Addr	Func	Start Bit # MSB	Start Bit # LSB	Bit Count MSB	Bit Count LSB	Error Check Field 16 Bit CRC
00-63	01	00	00	00	03	CRC

### 3.1.2. Example Response Message Reading Status Bits 0 - 2

Read Output Status bits 0 thru 2 Normal Response Message

Addr	Func	Byte Count	Bit Value	Error Check Field 16 Bit CRC
00-63	01	01	02	CRC

Green "Authorized" LED is on and unit permitting.

### 3.1.3. Example Query Message Reading Status Bits 16 - 31

Read Output Status bits 16 thru 31 Query Message

Addr	Func	Start Bit # MSB	Start Bit # LSB	Bit Count MSB	Bit Count LSB	Error Check Field 16 Bit CRC
00-63	01	00	10	00	10	CRC

## 3.2. Read Input Status Bits

## Function Code 02

The TAS checks these bits to determine the presence of a truck, and the status of the unit's hardware. Function 02 works in accordance with the Modicon manual chapter 2.

### 3.2.1. Example Query Message Reading Input Status Bits 0 - 15

Read Input Status bits 0 thru 15 Query Message

Addr	Func	Start Bit# MSB	Start Bit# LSB	Bit Count MSB	Bit Count LSB	Error Check Field 16 Bit CRC
00-63	02	00	00	00	10	CRC

### 3.2.2. Example Response Message Reading Input Status Bits 0 - 15

Read Input bits 0 thru 15 Normal Response Message

Addr	Func	Byte Count	Data Bits 7-0	Data Bits 15-8	Error Check Field 16 Bit CRC
00-63	02	02	4E	00	CRC

Data indicates truck present, communicating with VIP, in vehicle list, and authorized.

## 3.3. Read Multiple 16-Bit Registers

## Function Code 03

Function code 03 reads one or more 16-bit registers. Function code 03 query and response messages are in accordance with the Modicon Manual chapter 2. The 16-bit registers are listed in section *16-Bit Control and Data Registers*. The registers which must hold their value when power is removed are stored in EEPROM non-volatile memory, except the clock registers which are stored in the on-board (battery-backed FM & CSA approved models) clock.

### 3.3.1. Example Query Message Read Registers 3 and 4

Read Registers 3 thru 4 Query Message

Addr	Func	Start Reg# MSB	Start Reg# LSB	Reg Count MSB	Reg Count LSB	Error Check Field 16 Bit CRC
00-63	03	00	03	00	02	CRC

### 3.3.2. Example Response Message Read Registers 3 and 4

Read Registers 3 thru 4 Normal Response Message

Addr	Func	Byte Count	Data Reg 3 MSB	Data Reg 3 LSB	Data Reg 4 MSB	Data Reg 4 LSB	Error Check Field 16 Bit CRC
00-63	03	04	00	0B	00	1E	CRC

Time reads 11:30 AM

## 3.4. Force (Set) Single Bit

## Function Code 05

Function 05 query and response messages are in accordance with the Modicon Manual chapter 2. In the Intellitrol, the Force Single Bit commands are used as “Action” commands, to tell the controller to perform some action or function. The “force bit” number is the command index, and the “force data” value is the “bit value”. A “bit value” of “0” may indicate that in fact the command should be ignored (no action performed), or alternatively that the action should be set “Off”, or in some manner complemented from the “1” or “On” state. The various Force Bit commands are listed in section *VIP and Intellitrol Force Bit Assignments*.

The VIP accepts “force data” fields of 0000 (hex) as “0” and FF00 (hex) as “1” values. In general, the VIP treats “0” values as no-operation commands.

The Intellitrol accepts “force data” fields of 0000 (hex) as “0” and either FF00 (hex) or 0001 (hex) as “1” values. In general, the Intellitrol treats “0” values as complementing the “1” value action where meaningful, or as no-operation commands otherwise. For example, a “0” value to Force Bit 3 (Erase Vehicle List) is a no-operation action, while a “0” value to Force Bit 8 (Set Overfill Bypass) would *clear* any overfill-bypass currently in effect, and a “1” value to Force Bit 8 would *set* overfill-bypass (if the unit is in a bypassable overfill state).

While there is no pressing reason why the 16-bit “force data” value cannot range over the entire *short integer* range, at present no Force Bit function uses the force data value as anything other than a logical “0” or “1” (“Off” or “On” respectively), and will return a Data Error on any force data value other than the ones listed.

### 3.4.1. Example Query Message to Erase the Vehicle List

Force Single Bit Query Message To Erase The Vehicle List

Addr	Func	Force Bit# MSB	Force Bit# LSB	Force Data MSB	Force Data LSB	Error Check Field 16 Bit CRC
00-63	05	00	03	FF	00	CRC

### 3.4.2. Example Response Message to Erase the Vehicle List

Force Single Bit Normal Response Message

Addr	Func	Force Bit# MSB	Force Bit# LSB	Force Data MSB	Force Data LSB	Error Check Field 16 Bit CRC
00-63	05	00	03	FF	00	CRC

## 3.5. Write Single 16-Bit Register

## Function Code 06

Function code 06 write one 16-bit register. Function code 06 query and response messages are in accordance with the Modicon Manual chapter 2. The 16-bit registers are listed in section *16-Bit Control and Data Registers*. The registers which need to persist through power down are stored in EEPROM non-volatile memory. To avoid prematurely wearing out of the EEPROM, do not write to the registers unless necessary.

## 3.5.1. Example Query Message Write Register 0E

Write Single Register Query Message

Addr	Func	Reg Numb MSB	Reg Numb LSB	Reg Data MSB	Reg Data LSB	Error Check Field 16 Bit CRC
00-63	06	00	0E	00	03	CRC

This command remotely authorizes the VIP function of the unit.

## 3.5.2. Example Response Message Write Register 0

Write Single Register Normal Response Message

Addr	Func	Reg Numb MSB	Reg Numb LSB	Reg Data MSB	Reg Data LSB	Error Check Field 16 Bit CRC
00-63	06	00	0E	00	03	CRC

## 3.6. Write Multiple 16-Bit Registers

### Function Code 10 Hex

Function code 10 (hex) writes one or more 16-bit registers. Function code 10 query and response messages are in accordance with the Modicon Manual chapter 2. The 16-bit registers are listed in section *16-Bit Control and Data Registers*. The registers which need to persist through power down are stored in EEPROM non-volatile memory. The register contents will persist through power down and need not be refreshed upon power up.

### 3.6.1. Example Query Message Write Registers 100 - 101 (Set Intellitrol Time)

Write Registers 100 thru 101 Query Message

Addr	Func	Start Reg# MSB	Start Reg# LSB	Reg Count MSB	Reg Count LSB	Byte Count	Data Reg 0 MSB	Data Reg 0 LSB	Data Reg 1 MSB	Data Reg 1 LSB	Error Check Field 16 Bit CRC
00-63	10	01	00	00	02	04	2D	1C	5C	78	CRC

Set the Intellitrol date and time to 3:30 PM Christmas 1993.

### 3.6.2. Example Response Message Read Registers 100 - 101 (Set Intellitrol Time)

Write Registers 100 thru 101 Normal Response Message

Addr	Func	Start Reg# MSB	Start Reg# LSB	Reg Count MSB	Reg Count LSB	Error Check Field 16 Bit CRC
00-63	10	01	00	00	02	CRC

## 3.7. Write Single Vehicle

## Function Code 41 Hex

Write Single Vehicle Function Code 41 (hex) writes a single element into the vehicle list. In the VIP, bypass key serial numbers are stored within the 5,000-vehicle list; but on the Intellitrol, the bypass key list is maintained as a separate list. All numbers are in **hex** in the example below. Note that the Electronic Serial Number (TIM number) is written **Most Significant Byte first**:

### 3.7.1. Example Query Message

Write Vehicle List Element 0123456789AB to 1234 Query Message

Addr	Func	Elmnt Numbr MSB	Elmnt Numbr LSB	ESN MSB	ESN	ESN	ESN	ESN	ESN LSB	Error Check Field 16 Bit CRC
00-63	41	12	34	01	23	45	67	89	AB	CRC

ESN stands for Vehicle Identification Number read from the Scully TIM

### 3.7.2. Example Response Message

The control unit will echo the Write Vehicle list message back to the bus master to indicate success. An exception response message will be returned if the element number is out of range, or the unit's internal EEPROM fails to program (hardware failure).

Write Vehicle List Element 0123456789AB to 1234 Normal Response Message

Addr	Func	Elmnt Numbr MSB	Elmnt Numbr LSB	ESN MSB	ESN	ESN	ESN	ESN	ESN LSB	Error Check Field 16 Bit CRC
00-63	41	12	34	01	23	45	67	89	AB	CRC

## 3.8. Read Single Vehicle

## Function Code 42 Hex

Function Code 42 (Hex) reads back a single Vehicle List element. The response to the read request echoes back the slave's address, the function code, and the element number, as well as the requested data. If the requested vehicle element is FFFF (hex), this function will return the serial number of the currently attached vehicle. **In the VIP, bypass key serial numbers are stored within the Vehicle List. The slave's response looks just like the master's write command, except for bearing function code 42 instead of 41:**

### 3.8.1. Example Query Message to Read Single Vehicle

Read Single Vehicle Element 1234 Query Message

Addr	Func	Elmnt Numbr MSB	Elmnt Numbr LSB	Error Check Field 16 Bit CRC
00-63	42	12	34	CRC



## 3.8.2. Example Response Message

Read Single Vehicle Element 1234 Normal Response Message

Addr	Func	Elmnt Numbr MSB	Elmnt Numbr LSB	ESN MSB	ESN	ESN	ESN	ESN	ESN LSB	Error Check Field 16 Bit CRC
00-63	42	12	34	01	23	45	67	89	AB	CRC

ESN stands for Vehicle Identification Number read from the Scully TIM

## 3.8.3. Example Response Message

The unit will echo the Write Company ID name message back to the bus master to indicate success. An exception response message will be returned if the unit's internal EEPROM fails to program (hardware failure).

Write Company ID Name "EXON" Normal Response Message

Addr	Func	First Char 'E'	'X'	'O'	'N'	Error Check Field 16 Bit CRC
00-63	45	45	58	4F	4E	CRC

## 3.9. Write Multiple Vehicles

### Function Code 46 Hex

Function Code 46 (hex), writes multiple elements into the Vehicle List. All numbers are in hex in the example below. In the VIP, bypass key serial numbers are stored within the Vehicle List. Note that write messages should not contain more than 9 vehicle numbers for a VIP rack controller.

### 3.9.1. Example Query Message

Write Multiple Vehicle List Elements AAAAAA555555 to 1234  
and element 0123456789AB to 1235 Query Message

Addr	Func	Elmnt No. MSB	Elmnt No. LSB	No of Elmnts MSB	No of Elmnts LSB	ESN 0 MSB	ESN 0	ESN 0	ESN 0	ESN 0	ESN 0 LSB
00-63	46	12	34	0	2	AA	AA	AA	55	55	55

ESN 1 MSB	ESN 1	ESN 1	ESN 1	ESN 1	ESN 1 LSB	Error Check Field 16 Bit CRC
01	23	45	67	89	AB	CRC

ESN stands for Electronic Serial Number read from the Scully Truck Identification Module

## 3.9.2. Example Response Message

Write Multiple Vehicle List Elements Normal Response Message

Addr	Func	Elmnt Nmbr MSB	Elmnt Nmbr LSB	Elmnt Count MSB	Elmnt Count LSB	Error Check Field 16 Bit CRC
00-63	46	12	34	00	02	CRC

The normal response returns the slave address, function code, starting address and number of vehicle list elements written.

## 3.10. Read Multiple Vehicles

### Function Code 47 Hex

Function Code 47 (hex), reads multiple elements from the Vehicle List. All numbers are in hex in the example below:

### 3.10.1. Example Query Message

Read Multiple Vehicle List Elements Query Message

Addr	Func	Elmnt Nmbr MSB	Elmnt Nmbr LSB	Elmnt Count MSB	Elmnt Count LSB	Error Check Field 16 Bit CRC
00-63	47	12	34	00	02	CRC

### 3.10.2. Example Response Message

Read Multiple Vehicle List Elements AAAAAA555555 from 1234  
and element 0123456789AB from 1235 Normal Response Message

Addr	Func	Elmnt No. MSB	Elmnt No. LSB	No of Elmnts MSB	No of Elmnts LSB	Byte Count	ESN 0 MSB	ESN 0	ESN 0	ESN 0	ESN 0	ESN 0 LSB
00-63	47	12	34	0	2	0C	AA	AA	AA	55	55	55

ESN 1 MSB	ESN 1	ESN 1	ESN 1	ESN 1	ESN 1 LSB	Error Check Field 16 Bit CRC
01	23	45	67	89	AB	CRC

ESN stands for Electronic Serial Number read from the Scully Truck Identification Module

## 3.11. CRC Vehicle List

## Function Code 4A Hex

The CRC Vehicle List command is used to obtain a rack controller-derived Modbus CRC-16 value for a logically contiguous subset of the Vehicle List. This CRC value can be used by the TAS to decide whether the rack controller's Vehicle List is correct and up to date.

### 3.11.1. Example Query Message

To, for example, obtain the unit's CRC of the 100 (decimal) vehicle serial numbers stored in the Vehicle List as index entries 500 to 599 (decimal):

CRC Multiple Vehicle List Elements Query Message

Addr	Func	Elmnt Nmbr MSB	Elmnt Nmbr LSB	Elmnt Count MSB	Elmnt Count LSB	Error Check Field 16 Bit CRC
00-63	4A	01	F4	00	64	CRC

### 3.11.2. Example Response Message

The unit will respond by echoing the requested starting index and length, followed by the Vehicle List CRC-16 value.

Return calculated Vehicle List "slice" CRC-16

Addr	Func	Start Index MSB	Start Index LSB	Count MSB	Count LSB	CRC value MSB	CRC value LSB	Error Check Field 16 Bit CRC
00-63	4A	01	F4	00	64	65	AA	CRC

As a point of reference, 1 blank serial number (000000000000 hex or 6 consecutive "00" bytes) CRC's as 1B00 (hex); 10 contiguous blank serial numbers (60 consecutive "00" bytes) CRC as DBFF, and 100 contiguous blank serial numbers (600 consecutive "00" bytes) CRC as 65AA. The example response above shows all serial number entries 500 to 599 in the Vehicle List to be blank (there are in fact many different non-blank combinations of bytes that can CRC to one particular value; the odds that randomly something other than the correct serial numbers would CRC to the desired value are fairly small).

## 3.12. Write Bypass Keys

## Function Code 4B Hex

The Write Bypass Keys message is used to write one or more Bypass Authorizer serial numbers to the control unit's onboard EEPROM-resident Bypass Authorizer List. The format and operation of Write Bypass Keys (function 4B) is identical in operation and construction to *Write Multiple Vehicles* (function 46).

The Intellitrol by default has room for up to 32 Bypass Authorizer serial numbers; to determine the actual size of the Bypass Authorizer List, read the *Bypass Key Block Size* register (0AC), and divide by the size of a stored bypass key element (8).

### 3.13. Read Bypass Keys

#### Function Code 4C Hex

The Read Bypass Keys message is used to read one or more Bypass Authorizer serial numbers from the control unit's onboard EEPROM-resident Bypass Authorizer List. The format and operation of Read Bypass Keys (function 4C) is identical in operation and construction to *Read Multiple Vehicles* (function 47).

The Intellitrol by default has room for up to 32 Bypass Authorizer serial numbers; to determine the actual size of the Bypass Authorizer List, read the *Bypass Key Block Size* register (0AC), and divide by the size of a stored bypass key element (8).

### 3.14. Write “Enable-Features” Password

#### Function Code 4D Hex

The Write “Enable-Features” Password Modbus command is used to supply the rack controller unit with the “magic” enable byte stream that in turn authorizes unbundled (extra-cost, optional) rack controller features.

For the Intellitrol, the “password” is 8 bytes long. The VIP does not support this Modbus command.

The password-generation algorithm is Scully-proprietary and is not in the scope of this document.

#### 3.14.1. Example Message Set Enable Password

Set Enable-Features Password Command

Addr	Func	Byte Count	Pass-word MSB	Pass-word #2	Pass-word #3	Pass-word #4	Pass-word #5	Pass-word #6	Pass-word #7	Pass-word LSB	Error Check Field 16 Bit CRC
00-63	4D	08	03	AF	3C	19	D9	32	CE	17	CRC

#### 3.14.2. Example Message Set Enable Password Response

Set Enable-Features Password Response

Addr	Func	Byte Count	Error Check Field 16 Bit CRC
00-63	48	08	CRC

### 3.15. Normal Modbus Message Format

Currently, all Modbus command messages start with an address byte, then follow with a command (function code) byte, optionally followed by data, and terminated with a two-byte CRC-16. All Modbus responses follow the same form, the address byte is the address of the slave unit responding and not the address of the master (the master doesn't even have an address per se, “it's just the Master”), and the function byte is just echoing back the command function byte. A response message with bit 7 of the function code byte set is an “exception” response.

## Intellitrol/VIP Modbus RTU Protocol Specification

---

BYTE	FIELD	MEANING
0	Address	Normal Modbus unit address selection byte
1	Function Code	Normal Modbus function code byte 00 - 7F (hex)
. . .	...data...	Normal Modbus message data, if any
n-1, n	CRC	Normal Modbus CRC-16 bytes

No Modbus command message ever sends a function code with bit 7 set.

### 3.15.1. Modbus Exception Response Messages

Except for broadcast messages, when a master device sends a query to a slave device, it expects a normal response. One of four possible events can occur from the master's query:

1. If the slave device receives the query without a communication error, and can handle the query normally, it returns a normal response.
2. If the slave does not receive the query due to a communication error, no response is returned. The master program will eventually process a time-out condition for the query.
3. If the slave receives the query but detects a communication error (parity or CRC), no response is returned. The master program will eventually process a time-out condition for the query.
4. If the slave receives the query without a communication error but cannot handle it (e.g. if the request is to read a non-existent register), the slave will return an exception response informing the master of the nature of the error.

### 3.16. Example Exception Response Message

When an error occurs, an exception response message is returned. The message is generated by setting the most significant bit (bit 7) in the function code byte. This byte is returned along with the address, the generated exception response code, and 16-bit CRC. The following is an example exception response message:

Example Exception Response Message

Addr	Func	Excep Resp Code	Error Check Field 16 Bit CRC
00-63	81	02	CRC

Function code 01 was sent with an illegal data address.

An exception response message was returned by setting bit 7 in the Function Code byte.

The exception response code returned was an 02, ILLEGAL DATA ADDRESS.

### 3.17. Exception Response Codes

The following exception response codes can be generated from VIP or Intellitrol Modbus commands.

RESPONSE #	RESPONSE NAME	DEFINITION
00	NO MODBUS ERROR	Command executed correctly. No exception response error code was returned.
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave.
04	SLAVE DEVICE FAULT	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a time-out error from occurring in the master.
06	SLAVE DEVICE BUSY	The slave is engaged in processing a long duration program command. The master should re-transmit the message later when the slave is free.
07	NEGATIVE ACKNOWLEDGE	The slave cannot perform the program function received in the query. This code is returned for an unsuccessful programming request using function code 13 or 14 decimal. The master should request diagnostic or error information from the slave.
08	MEMORY PARITY ERROR	
09	TIM COMMAND ERROR	
0A	TIM FAMILY ERROR	
0B	TIM NOT VALID	
0C	NUMBER OF COMPARTMENTS ERROR	
0D	SPI FAMILY ERROR	
0E	SPI WRITE ERROR	
0F	SPI READ ERROR	
10	TIM MEMORY AREA ERROR	
11	WRITE TO SCRATCH PAD ERROR	
12	VERIFY SCRATCHPAD ERROR	
13	COPY SCRATCHPAD ERROR	

14	TIM ENTRY NOT VALID	
15	READ SERIAL ERROR	
16	MEMORY ALLOCATION ERROR	
17	I2C ERROR	
18	READ CLOCK ERROR	
19	READ ONLY VALUE	
80	NO RESPONSE	



## 4. CRC Generation

The following procedure is used to generate the 16-bit CRC sent and received with every Modbus command:

1. Load a 16-bit register with FFFF hex (all 1's). Call this the CRC register.
2. Exclusive OR the first 8-bit byte of the message with the low-order byte of the 16-bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the LSB), zero-filling the MSB. Extract and examine the LSB.
4. If the LSB was 0: Repeat step 3 (another shift).
5. Otherwise, if the LSB was 1: Exclusive OR the CRC register with A001 hex.
6. Repeat steps 3 through 5 until 8 shifts have been performed. When this is done, a complete 8-bit byte will have been processed.
7. Repeat steps 2 through 6 for the next 8-bit byte of the message. Continue doing this until all bytes have been processed.
8. The final contents of the CRC register are the CRC value.

#### 4.1. “C” Implementation of CRC-16

Here are two sample CRC-16 implementations. The first is from the Intellitrol rack controller. It is big (takes a lot of statically defined firmware data space for its two tables) and fast. The second is from the VIP. It is small and slow.

#### 4.1.1. Big'n'Fast “C” Implementation of CRC-16

```
const static unsigned char hi_crc_table[] =
{
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
```

```
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40  
};
```

```
const static unsigned char lo_crc_table[] =  
{  
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2,  
    0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,  
    0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,  
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,  
    0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,  
    0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,  
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6,  
    0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,  
    0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,  
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,  
    0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE,  
    0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,  
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA,  
    0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,  
    0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,  
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,  
    0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62,  
    0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,  
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE,  
    0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,  
    0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,  
    0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,  
    0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76,  
    0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,  
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,  
    0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,  
    0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,  
    0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,  
    0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A,  
    0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,  
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86,  
    0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40  
};
```

```
/* Returns the 16-bit CRC value based on the buffer information. */
```

```
unsigned short modbus_CRC
(
    unsigned char *bufptr,      /* Starting address */
    unsigned short buflen,     /* CRC Segment length */
    unsigned short seed        /* Initial CRC seed Value */
)
{
    unsigned char index;
    unsigned char crc_hi;
    unsigned char crc_lo;

    crc_hi = (byte) (seed >> 8);
    crc_lo = (byte) (seed);

    while (buflen--)
    {
        index = (unsigned char) (crc_lo ^ *bufptr++);
        crc_lo = (unsigned char) (crc_hi ^ hi_crc_table[index]);
        crc_hi = lo_crc_table[index];
    }
    return((short) (crc_hi << 8) | crc_lo);
}
```

### 4.1.2. Small'n'Slow "C" Implementation of CRC-16

```
#define CRCPOLY 0x0A001
static unsigned int lrc (byte * inbuf, int nbyte, unsigned int seed)
/*****
 *      Purpose:      Compute MODBUS CRC I/A/W Modicon Manual      *
 *      Inputs:       pointer to buffer and buffer size            *
 *      Returns:      the CRC                                       *
 *      Side effects:  none.                                         *
 *****/
{
    unsigned int    crc = seed;
    unsigned int    newbyte;
    byte    i,j;
    byte    flag;

    for (i = 0; i < nbyte; i++)
    {
        newbyte = *inbuf++;
        newbyte &= 0xFF;
        crc ^= newbyte;
        for (j = 0; j < 8; j++)
        {
            flag = (byte) (crc & 1);
            crc >>= 1;
            if (flag)
                crc ^= CRCPOLY;
        }
    }
    return crc;
}
```

[End of      *Intellitrol/VIP Modbus RTU Protocol Specification*      Total of 68 pages]