Scully Rack Controller System

Intellitrol

Modbus RTU Protocol Specification

11/19/2019

FOR INTERNAL USE ONLY

Scully Signal Company
70 Industrial Way
Wilmington, MA, USA 01887

Drawing No. D-020-03
Revision 3.2
MODBUS32.DOC

Revision History

Revision	Who	Date	Changes Made
2.3	SFT	03/27/95	• Added Write Multiple Register command 10 hex & UNIX date registers
2.4	SFT	04/20/95	• 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.
2.5	SFT	04/26/94	• 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.
2.6	RDH	11/15/95	• Change all “Cybertrol” to Intellitrol; minor formatting changes and prettification; Relay out sections to “make it flow better”; Sectionize for page numbering (TOC in roman, etc.).
2.7	RDH	12/03/95	• 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
2.8	MR	12/29/97	• Added information to reflect changes in Version 1.05 Software
2.9	KLL	02/27/08	• Added information to reflect changes in Version 1.09 Software
2.10	KLL	09/23/10	• Added Intellitrol 2 Model Number
2.11	KLL	02/23/10	Added VIP mode 5 description Added Ground monitoring while trying to identify the truck probe Added modbus register 7E to blink the permit leds when the ground is good while trying to identify the truck probe type
3.0	KLL	08/16/11	Add modbus command 2D and 2E description

3.12	KLL	08/20/12	Added Modbus commands 44 and 45. Added byte count to command 45 example
3.2	HP	11/19/19	• Updated for version 1.7.0 • Added SuperTIM Support • Added Unload and Load mode • Added compartment count • Added version display on startup (x.y.z where x, y, z can be 0 – 8)

Table of Contents

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. Modbus command 48 (hex) has been added to support functionality of the backup processor. The main processor will not respond to this command, as collision between the main and backup processors would result. Except for this command, all other commands, registers, and status bits apply to the main processor only. This command is for in-house Scully use only and should not be implemented on any commercial TAS system.

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 it's 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). If a microprocessor should fail ("hang", "lock up", etc.), its permit relay drive circuit will automatically "decay" within a few hundred milliseconds and deactivate, thus forcing the unit non-permissive. Each microprocessor can independently monitor the *actual* state of both internal permit relays and can force the rack controller unit non-permissive if either microprocessor determines that the other's permit relay is shorted. 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.

The VIP system can also authorize loading based upon additional information stored in the TIM RAM other than the TIM laser serial number, called the Date Stamp. In this "Date Stamp" mode of operation the RAM in the TIM is organized like a tiny floppy diskette. A "file" containing the truck's expiration date is written into the TIM, like the expiration date on a credit card. The VIP checks the TIM (vehicle) expiration date against the onboard clock/calendar and allows the truck to load if the expiration date is good (in the future). To prevent forgery, the expiration date is encrypted (see the TRUCKID Touch Data Format Design Specification Rev 1.3). To allow the same truck to load at multiple suppliers, the date stamp for each supplier is kept in a unique TIM Date Stamp file. To allow the truck to load at some terminals but not at others, a list of terminal numbers can be placed in the TIM Date Stamp file. The maintenance of the truck's Date Stamp file(s) is beyond the scope of this document. Modbus commands *Write Company ID* and *Write Password* to set the Date Stamp file name and the encryption key are supported by both the Intellitrol and VIP rack controllers. The unit's "terminal number" is settable via the Modbus *Terminal Number* register.

Future Scully products will use this protocol with some additions to support additional features.

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.
4. PIDX Product code standard 04-101-15-45-2010

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. **Current rack controller units may transmit the response message as late as 1.0 second after the query message depending on the Modbus function and/or present rack controller task being executed.** The bus master should wait at least this long before processing a time-out. A bus collision and garbled data will occur if the bus master transmits its next query before 1.0 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 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 interpolate 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.

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. Non-Volatile Memory

Different rack controllers have need to store various amounts of information in such a manner that the information is preserved across possibly extensive power-off cycles. EEPROM (Electrically Erasable Programmable Read-Only Memory) is the current architecture of choice for "miscellaneous" non-volatile data storage, although FLASHRAM is rapidly gaining in popularity for bulk non-volatile data (e.g., "firmware") storage.

EEPROM is slow to write (relatively speaking; "typical" write times are on the order of 5 milliseconds) and has a limited lifetime (typically good for 100,000 write operations) but doesn't require power (even a battery) to remember its data.

FLASHRAM tends to be much faster to write (on the order of tens of microseconds), but can only be bulk-erased, rendering it unsuitable for the non-volatile store of "random little tidbits" like a single 16-bit value.

2.1.1. Intellitrol Non-Volatile Memory

The Intellitrol rack controller uses FLASHRAM for both static data and firmware storage, and EEPROM for other more dynamic non-volatile memory storage requirements.

The EEPROM is logically subdivided into several distinct blocks or "partitions". These partitions help segregate data and enhance the management of competing data subsystems' demands (for example, insulating the Event Log from the Vehicle List).

2.1.1.1. EEPROM Home Block

The first or "root" EEPROM Partition is the Home Block. The Home Block contains all the partitioning information used by the rack controller to identify and access the rest of the partitions in the EEPROM address space (much like a disk filesystem)

2.1.1.2. EEPROM Boot Block

The Boot Block is a block of non-volatile storage reserved for the firmware kernel's use and is not available to the firmware shell.

2.1.1.3. EEPROM Crash Block

The Crash Block is reserved for use in recording "system crash" information for later retrieval and analysis. Version 1.0 of the Intellitrol does not yet make use of this block; as such the data stored here is undefined.

2.1.1.4. EEPROM System Info Block

The System Info Block is where all the "miscellaneous" non-volatile data is stored. Various disjoint Modbus registers are grouped here (by the firmware), for example.

The Intellitrol further subdivides the System Info block into smaller and more manageable data records or structures for ease of manipulation and access via Modbus commands. These System Info data structures are generally on the order of 64 bytes in size, with a dedicated CRC-16 for each structure to validate the data stored within the structure. The separate structures are allotted directly in the System Info block at fixed offsets.

The Modbus functions *Write Special EEPROM Block* and *Read Special EEPROM Block* may be used to write and read the various System Info data structures. These commands are not intended for production TAS usage, they are for Scully manufacturing and service use only. Improper use of these functions can corrupt the EEPROM structure and result in a dysfunctional rack controller unit.

The "Special" blocks are:

VALUE	NAME	DEFINITION
0	E2SYS_PARM	SysParmNV general system parameters
1	E2SYS_DSTAMP	DateStampNV Date Stamp parameters
2	E2SYS_DIA5	SysDia5NV Five-Wire optic diagnostic voltages
3	E2SYS_VOLT	SysVoltNV self-test/diagnostic voltage thresholds
4	E2SYS_SET1	SysSet1NV self-test/operation parameters
5 - ...		Reserved

In general, first read the special block from the rack controller to get the current values, change the desired item's value, recalculate the block's CRC-16, then write the special block back to the rack controller. A system reset may be required before the new values take effect.

2.1.1.4.1. SysParmNV Structure [Special Block 0]

The SysParmNV block collects the assorted and miscellaneous data ("general system parameters") that don't really belong anywhere else. For example, the *Modbus Minimum Response Time* register value and the Features-Enable password mask both reside in the SysParmNV structure.

The SysParmNV block values should only be read and set via the distinct Modbus registers or other commands, since there can be subtle interactions in the order of setting parameters.

2.1.1.4.2. DateStampNV Structure [Special Block 1]

The DateStampNV block contains the TAS-directed "Date Stamp" company id (as set by the *Write Company ID Name* Modbus function) and the Date Stamp enabling password (as set by the *Write Password* Modbus function).

The DateStampNV block values should only be set via the normal Modbus Date Stamp functions, as the actual data is internally encrypted to protect the password.

2.1.1.4.3. SysDia5NV Structure [Special Block 2]

The SysDia5NV block contains the table of Five-Wire Optic probe “diagnostic wet voltages” used by the Intellitrol to determine which compartment (probe) is “wet”. This block is normally blank, directing the rack controller to utilize the default values from the shell firmware as the current operational values. This 16-entry table of voltages overrides (when set) the default table that resides in FLASHRAM as part of the shell firmware.

3. SysVoltNV Structure [Special Block 3]

The SysVoltNB structure contains voltage thresholds used in the system self-test diagnostics. This block is normally blank, directing the rack controller to utilize the default values from the shell firmware as the current operational values. These voltages override (when set) the default values that reside in FLASHRAM as part of the shell firmware

3.0.0.0.1. SysSet1NV Structure [Special Block 4]

The SysSet1NB structure contains system parameters and thresholds used in the system operations. This block is normally blank, directing the rack controller to utilize the default values from the shell firmware as the current operational values. These parameters override (when set) the default values that reside in FLASHRAM as part of the shell firmware

3.0.0.1. EEPROM Event Log

The Event Log is an internal circular list of recent “events” of interest that are stored in non-volatile memory. See *Event Log Data Structure* below.

3.0.0.2. EEPROM Bypass Key List

The Bypass Key List is an internal list of authorized bypass key serial numbers that are stored in non-volatile memory. See *Bypass Key List Data Structure* below.

3.0.0.3. EEPROM Vehicle List

The Vehicle List is an internal list of authorized vehicle serial numbers that are stored in non-volatile memory. See *Vehicle List Data Structure* below.

3.0.1. VIP Non-Volatile Memory

The VIP rack controller uses EEPROM for both data and firmware non-volatile storage and does not “export” any useful system structure regarding the EEPROM (except of course for the Vehicle List).

3.1. 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, with one important exception — all serial numbers will have the high-order two digits (8 bits or one byte) as zero. 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; both use the "high-order always-zero" byte of the serial number itself to store the CRC, and thus require that the CRC high byte be a priori knowable (i.e., "0").

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 FFFFFFFFFFFF (hex).

The Vehicle List is manipulated via Modbus functions *Write Single Vehicle*, *Read Single Vehicle*, *Write Multiple Vehicles*, *Read Multiple Vehicles* and *CRC Multiple Vehicles* as well as by *Force Single Bit* (*Erase Vehicle List*).

3.1.1. VIP Vehicle List Features and Operations

The VIP Vehicle List is used for both vehicles and bypass keys. Serial numbers of authorized bypass keys are entered the vehicle list just as if they were vehicles (and thus the high byte of VIP bypass key serial numbers must also be 00). The vehicle list also contains an 8 bit internally calculated and verified CRC for error detection.

The VIP maintains a "high water mark" Vehicle List pointer (see *Vehicle List Size* register below) indicating the highest-numbered Vehicle List entry in use. The TAS may read this register to decide if the Vehicle List is up to date. This is not an especially reliable "validation" as it says nothing about what is actually written in the Vehicle List, just "how much" is written.

The Current Truck Register (CTR) is a 48-bit pseudo-register that contains the serial number of the currently connected truck. The CTR 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. Access the Current Truck Register as a special element number (FFFF hex) of the Vehicle List. Write operations on the Current Truck Register will return the illegal data address exception message.
The VIP supports only the "single vehicle" Modbus functions *Read Single Vehicle* and *Write Single Vehicle*.

3.1.1.1. VIP Supported Features/Operations

1. VIP Vehicle List has a maximum 5,000 vehicle capacity.
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.
4. Delete (erase) old vehicles by over-writing them with vehicle number 0.
5. Verify vehicle list by reading back individual vehicle numbers with *Read Single Vehicle* Modbus function (or "guess" based on *Vehicle List Size* register).
6. Delete (erase) all vehicles by over-writing all of them with vehicle number 0.
7. Bypass keys reside in the Vehicle List and are indistinguishable from vehicles.

3.1.2. 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 Intellitrol does not support the *Vehicle List Size* register.

The Intellitrol supports the VIPish Current Truck Register as defined above. However, for the Intellitrol, it is preferred that the *Truck Serial Number* registers be used instead. 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*, *Write Single Vehicle*, *Insert Single Vehicle,* and *Remove 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 30 millisecond response to external safety issues ("Overfill" conditions), the Intellitrol limits the maximum number of vehicles to be CRC'ed to 100 (which takes approximately 25 milliseconds). Since the Modbus messages involved are small (8 - 10 bytes), the entire 5000-long Vehicle List can be "verified" by 50 *CRC Multiple Vehicle* messages much faster than the vehicles can be read by the TAS. For an Intellitrol on a 9600 baud line, the Intelliview program [a prototypical TAS-ish program] typically takes less than 3 - 5 seconds to validate the entire 5000-long Vehicle List.

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.

3.1.2.1. Intellitrol Supported Features/Operations

1. Intellitrol Vehicle List has a maximum 5,000 vehicle capacity standard, up to 10,000 vehicle capacity with optional 28C513 EEPROM installed.
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 either *Write Single Vehicle*, *Insert Single Vehicle*, or *Write Multiple Vehicles*.
4. Delete (erase) old vehicles with the *Remove Single Vehicle* function.
5. Verify vehicle list by reading back vehicle numbers with either *Read Single Vehicle* or *Read Multiple Vehicles* Modbus functions, or by *CRC Multiple Vehicles* Modbus function.
6. Delete (erase) all vehicles by using the *Force Single Bit (Erase Vehicle List)* Modbus function.
7. Bypass keys are kept in separately managed Bypass Key List.

3.2. 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. Unlike vehicle serial numbers, all six bytes of the bypass key serial number are available (the Bypass Key List stores serial numbers as 8-byte structures with 2 bytes of CRC-16 information to validate each entry).

The Intellitrol Bypass Key List is normally limited to 32 keys but can contain more with the optional 28C513 EEPROM. 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.

3.3. 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”).

3.3.1. 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.

3.4. 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.

3.4.1. 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

3.4.2. 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

3.4.3. 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

3.4.4. Hardware Error [Event Type 04]

Event Type code 04 (hex) records system hardware errors. Hardware errors are typically detected by the resident firmware's selftest 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

3.4.4.1. 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	KernelBad	Actual Firmware Kernel CRC-16 value
4 - 5	ShellGood	Proper Firmware Shell CRC-16 value
6 - 7	ShellBad	Actual Firmware Shell CRC-16 value
8 - 21		Reserved

3.4.4.2. 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
2 - 21		Reserved

3.4.5. Voltage Error [Event Type 05]

Event Type code 05 (hex) records erroneous voltages detected by the firmware selftest 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)

3.4.6. 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

3.4.7. 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

3.4.8. 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	Pin 5 High Resistance	Socket pin 5 resistance is higher than expected
2 - 21		Reserved

3.4.9. 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

3.4.10. Overfill Info 2 [Event Type 0A]

Event Type code 0A (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

3.5. 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

3.5.1. 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.

3.5.2. 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.

3.5.3. 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.

3.5.4. 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.

3.5.5. 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.

3.5.6. 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 FFFFFFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS "bypass key".

3.5.7. 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 FFFFFFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS "bypass key".

3.5.8. 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).

3.5.9. 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).

3.5.10. 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.

3.5.11. 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).

3.5.12. 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 FFFFFFFFFFFF (hex) in the *Bypass Key Serial Number* registers to indicate a bypass by TAS “bypass key”.

3.5.13. 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.

3.6. 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.

The 32 Input Status Bits are:

INPUT BIT	CONDITION	MEANING IF READ BACK AS 1
0	Fault	Fault (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	Rack controller is in debug mode and will ignore fault conditions. **This is an unsafe operating condition!**
9	Pin 5 High Resistance	Socket pin 5 resistance is higher than expected

INPUT BIT	CONDITION	MEANING IF READ BACK AS 1
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

3.6.1. 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 rack controller 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 (even if the main and/or Backup microprocessors are unable to respond to Modbus commands). The Service LED is a “Fail Safe” circuit that will blink even if the microprocessor(s) is(are) not running — only a properly running main and backup microprocessor pair can suppress the Service LED from blinking.

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.

3.6.2. 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. 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.)

3.6.3. Truck Talk [Input Status Bit 02]

The Truck Talk status bit indicates that the rack controller unit has successfully established communications with an active on-truck unit. This truck unit may be simply a SuperTIM, or a more intelligent on-truck management system.

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 FFFFFFFFFFFF (hex) would then indicate that the TIM or the communications was faulty). Simply sensing overfill probes does not set Truck Talk.

3.6.4. Truck Valid [Input Status Bit 03]

The Truck Valid bit means that a truck serial number (as reported in the *Truck Serial Number* register) is in the unit's Vehicle List.

3.6.5. Bypass [Input Status Bit 04]

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

For the Intellitrol, the *Bypass State* register contains the current active bypass information.

3.6.6. Idle [Input Status Bit 05]

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

3.6.7. 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" have been met. These criteria can include vehicle authorization by serial number, vehicle ground safety proving, etc. Alternatively, the rack controller unit may be "bypassed" and permitting despite failure to satisfy one or more of the configured criteria.

For the Intellitrol, if the green "Permit" LED 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.

3.6.8. 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 (assuming the Intellitrol is not "Idle").

3.6.9. Debug [Input Status Bit 08]

The Debug status bit indicates that the hardware Debug jumper is installed. This directs the rack controller to ignore many safety and fault conditions and permit anyway. Running with the Debug jumper installed is an inherently unsafe mode of operation and should not normally be done.

3.6.10. Pin 5 High Resistance [Input Status Bit 09]

The Pin 5 High Resistance status bit indicates that the socket pin #5 resistance is higher than expected. This is the 5-wire diagnostic line connected to TB4 pin 5.

3.6.11. 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 or engaged.

3.6.12. Diode GND [Input Status Bit 13]

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

3.6.13. Resistive GND [Input Status Bit 14]

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

3.6.14. Intellicheck [Input Status Bit 15]

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

3.6.15. 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.

3.6.16. ADC Time-Out [Input Status Bit 18]

The ADC Time-out status bit indicates that a problem has been detected with the on-board Analog-to-Digital converter.

3.6.17. 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.

3.6.18. 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. This is typically not a Fault condition, unless the rack controller is running in "DateStamp" mode, which required accurate Date/Time information.

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.

3.6.19. 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.

3.6.20. Truck [Input Status Bit 22]

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

3.6.21. Kernel CRC Error [Input Status Bit 23]

The Kernel CRC Error status bit means that the firmware has detected a bad firmware "kernel" image. This problem is not typically fixable in the field, 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.

3.6.22. 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.

3.6.23. TIM Data Line Fault [Input Status Bit 26]

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

3.6.24. 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.

3.6.25. 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 rejumpered (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.

3.6.26. 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 (e.g., by a TAS), and it is the responsibility of the TAS to inform the pertinent personnel of the Shutdown state, as well as to “Recover” the rack controller to clear the Shutdown state when appropriate.

3.6.27. Relay Error [Input Status Bit 31]

The Relay Error status bit indicates that the rack controller firmware has detected a problem with the permit relay(s). The relay(s) can be shorted (“permitting” when shouldn’t be) or broken (not “permitting” when should be).

For the Intellitrol, the current status of both the main and backup permit relays may be read via the *Relay State* register.

3.7. 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”
10 - 15		Reserved

3.7.1. 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.

3.7.2. Five-Wire-Optic Echo Pulse [Output Status Bits 06 - 07]

The Five-Wire-Optic Echo Pulse status bits indicate the that rack controller is actively receiving input "echo" pulses in response to transmitting Five-Wire optic probe output pulses. The STSO_5WIRE_ECHO1 bit indicates that the controller has "just now" received an echo 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 "polls" Five-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 echo 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. (The Intellitrol unit may actually be sporadically blinking with flaky probes, but in general no TAS will be able to poll the Intellitrol fast enough over the Modbus link to accurately track exactly the state of the front panel LED, ergo the one-second hysteresis built into these paired status bits.)

3.7.3. 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 electronics (other than the tank probes). 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.

The Intellitrol actively communicates with the attached truck on a sporadic basis. The 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" but that is hard for a TAS to represent, let alone determine accurately).

3.8. 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 VIP units 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/VIP unit act upon (for example, "date and time" registers).

The early VIPs (version 3.17 and earlier) had only a few values to provide via 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	Diagnostic Voltages	These registers provide a window into the Intellitrol's internal voltages. Some are "static" such as the "Rail" voltages determined by the periodic diagnostics; while others are dynamic and subject to change from one reading to the next.
0060 - 006F	Internal State/Status	The State/Status registers provide a more detailed view of the current internal state of many of the unit's sub-systems, finite state machines, and so forth.
0070 - 008F	Non-Volatile Parameters	This block of registers groups together assorted and disparate non-volatile unit settings (registers). This block is intended to allow the unit to group parameters together such that the TAS can set "most" normally-set parameters with one write operation (which also allows the unit to "optimize" EEPROM write sequences).
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	Error/Debug	This block of registers is something of a "wildcard" access to varied information of a "debugging" (or otherwise non-production intent). The definition of these registers is subject to change from release to release of the firmware. For "Production" TAS usage, these registers should be viewed as "undefined".
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	Registers for accessing data from a connected TIM Module.
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
000-01F	VIP COMPATIBILITY				Original VIP implementation; Some Intellitrol emulation

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
0000	VIP Year	R / W*	1992 - 2050	On-board Dallas™ TOD clock	Current year. (* Intellitrol: Read-only)
0001	VIP Month	R / W*	1 - 12	On-board Dallas™ TOD clock	Current month. (* Intellitrol: Read-only)
0002	VIP Day	R / W*	1 - 31	On-board Dallas™ TOD clock	Current day. (* Intellitrol: Read-only)
0003	VIP Hour	R / W*	0 - 23	On-board Dallas™ TOD clock	Current hour. (* Intellitrol: Read-only)
0004	VIP Minute	R / W*	0 - 59	On-board Dallas™ TOD clock	Current minute. (* Intellitrol: Read-only)
0005	Shell version	R	—	Permanent	Shell Version Number
0006	VIP Serial Number	R / W	—	Non-Volatile	VIP: Serial Number; Intellitrol: 0
0007	Latest Log pointer	R	0 - 31	Volatile	Last log entry (VIP-log only)
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 - 1024	Non-Volatile	Modbus command response wait (milliseconds)
000C	Shell Checksum	R	—	Permanent	Checksum of uploaded shell program (hex)
000E	Mode Control	R	0 - 5	Non-Volatile	Authorization mode control
000F	Kernel Version	R	—	Permanent	Kernel Version Number
0012	Scully type / model	R	—	Permanent	Rack controller model number (0 or error is undefined; assume VIP) 1. VIP 2. Intellitrol
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	Reserved/Future configuration bits
0028	Config-D	R	—	Non-Volatile	Reserved/Future configuration bits
0029	RAM size (in KB)	R	—	Hardware-Based	Size of Dynamic RAM available to Shell code (KB)
002A	FLASHRAM size (in KB)	R	—	Hardware-Based	Size of Kernel/Shell FLASHRAM storage (KB)
002B	EEPROM size (in KB)	R	—	Hardware-Based	Size of Non-Volatile EEPROM storage (KB)
002C	Max ModBus message size	R	—	Permanent	Maximum Modbus message size (bytes)

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
002D	Number of 5 wire probes	R	—	Dynamic	Number of probes on current truck. SIL CPU only
002E	Enable Modbus Features	R	—	Non-Volatile	Software enable features
0030	Reference volt	R	—	Static	Diagnostic/Calibration “Reference Volt”
0031	Raw Power Supply (13 V)	R	—	Static	110/220-V Line-based raw power supply voltage
0032	Probe Bias Voltage	R	—	Static	Internal probe bias voltage
0033	5-Wire Optic Pulse	R	—	Static	Five-wire-optic output pulse voltage
0038 - 003F	Channel Noise Voltage	R	—	Static	Channels 1 - 8 noise voltages
0040 - 0047	Channel 10-Volt Rail	R	—	Static	Channels 1-8 “10 Volt” rail voltages
0048 - 004F	Channel 20-Volt Rail	R	—	Static	Channels 1-8 “20 Volt” rail voltages
0050 - 0057	Channel Voltage	R	—	Dynamic	Channels 1-8 last-read voltages
0060	Read Clock Status	R	—	Dynamic	On-Board Dallas™ Clock state
0061	Relay Status	R	—	Dynamic	Backup relay (in high byte) and Main relay (in low byte) state
0062	EEPROM status	R	—	Dynamic	On-board EEPROM status
0064	"Acquire" state	R	—	Dynamic	“Acquire” state engine
0065	"Probe-Try" state	R	—	Dynamic	“Probe-Try” state engine
0066	Optic 5-wire state	R	—	Dynamic	“Five-Wire-Optic” state engine
0067	Optic/Thermal 2-wire state	R	—	Dynamic	“Two-Wire” state engine
0068	VIP status (DateStamp)	R	—	Dynamic	VIP Status (“DateStamp”)
0069	Service "A" flags	R	—	Dynamic	Service-A (“iambroke”) flags
006A	Service "B" flags	R	—	Dynamic	Service-B (“iamsuffering”) flags
006B	Service "C" flags	R	—	Dynamic	Service-C reserved for future
006C	Combined VIP Status	R	—	Dynamic	VIP/DateStamp status in high byte; VIP status in low byte
006D	Ground Status	R	—	Dynamic	Ground status in low byte
006E	VIP status (badvipflag)	R	—	Dynamic	VIP Status (“badvipflag”)
006F	VIP status (CERTIFICATE_ds_fails)	R	—	Dynamic	VIP Status (“CERTIFICATE_ds_fails”)
0070	Shorts test active	R / W	0 - 1	Non-Volatile	Shorts Test Enable/Disable flag
0071 - 0074	Debug Pulse on Code	R / W	0 - 255	Non-Volatile	Debug Pulse Enable Flag
0075 - 0078	Debug Pulse on Failed Code	R / W	0 - 255	Non-Volatile	Debug Pulse on Failure Enable Flag
0079 – 007A	Software Feature Enable Code	R / W	0 - 255	Non-Volatile	Software Enabled Features

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
007B	TIM read enable	R / W	0 - 1	Non-Volatile	When this register is set to 1 and the system has VIP disabled, the Intellitrol will allow any TIM serial number to be considered found in the list of valid serial numbers, although no serial numbers will need be stored in the controller VIP's list.
007C	Resistive Ground reference value	R / W	0 - 3	Non-Volatile	Allowed Threshold for ground resistance.
007E	Enable Ground Display	R / W	0 - 1	Non-Volatile	When this register is set to 1 the Intellitrol will blink the green permissive LEDs during the probe identification loop and a good ground.
007F	5-Wire Compartment Count Display Time	R / W	0 - 255	Non-Volatile	Length of time to display compartment count when connecting to truck.
0080	Active Deadman Enabled	R / W	0 - 255	Non-Volatile	Enable Active Deadman
0081	Active Deadman Max open time	R / W	0 - 30	Non-Volatile	Deadman max open time
0082	Active Deadman Max close time	R / W	0 - 600	Non-Volatile	Deadman max close time
0083	Active Deadman Warning time	R / W	0 - 60	Non-Volatile	Deadman warning time
0084	Software Feature Enable Unload Terminal	R / W	0 - 255	Non-Volatile	Enable unload terminal mode
0085	SuperTim max unload time	R / W	0 - 65535	Non-Volatile	Max unload time
0086	SuperTim Certificate date enable mask	R / W	0 - 31	Non-Volatile	Select which certificates are used
0087	Software Feature Enable compartment count check	R / W	0 - 255	Non-Volatile	Enable compartment count comparison between truck and TIM value
0088	SuperTim fuel type check mask	R / W	0 - 255	Non-Volatile	Select which compartments to compare to TIM value
0089	Software Feature Enable auto write fuel type flag	R / W	0 - 255	Non-Volatile	Enable writing fuel type to TIM on connect
008A - 008B	SuperTim default fuel type	R / W	0 - 65535	Non-Volatile	Default fuel type to write to TIM
00A0 - 00A1	EEPROM base address	R	—	Hardware-Based	32-bit "physical address" of on-board EEPROM memory block
00A2	Home block size	R	—	Permanent	Size of "Home" block (bytes)
00A3	Home block offset	R	—	Permanent	Offset from start of EEPROM

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
00A4	Boot block size	R	—	Non-Volatile	Size of “Boot” block (bytes)
00A5	Boot block offset	R	—	Non-Volatile	Offset from start of EEPROM
00A6	Crash block size	R	—	Non-Volatile	Size of “Crash” block (bytes)
00A7	Crash block offset	R	—	Non-Volatile	Offset from start of EEPROM
00A8	System block size	R	—	Non-Volatile	Size of Non-Volatile System Info block (bytes)
00A9	System block offset	R	—	Non-Volatile	Offset from start of EEPROM
00AA	Event log block size	R	—	Non-Volatile	Size of “Event Log” (bytes)
00AB	Event log block offset	R	—	Non-Volatile	Offset from start of EEPROM
00AC	Bypass key block size	R	—	Non-Volatile	Size of Bypass Key list (bytes)
00AD	Bypass key block offset	R	—	Non-Volatile	Offset from start of EEPROM
00AE	Truck ID block size	R	—	Non-Volatile	Size of Truck-ID (TIM) list (bytes)
00AF	Truck ID block offset	R	—	Non-Volatile	Offset from start of EEPROM
00E0	Soft COMM_ID ground errs	R	—	Dynamic	Ground-Check “soft” COMM_ID errors
00E1	Hard COMM_ID ground errs	R	—	Dynamic	Ground-Check “hard” COMM-ID errors
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
011A	Non-Permit Reasons	R	—	Dynamic	Flags indicating why non-idle unit is non-permissive

#	REGISTER NAME	DIR	RANGE	STORAGE	DESCRIPTION
011B	Latest Log pointer	R	—	Dynamic	Latest Event Log Pointer
011C	Optic 2 wire Threshold	R / W	0 - 4375	Non-Volatile	Optical Probe Upper Threshold
011D	Hysteresis	R / W	0 - 700	Non-Volatile	Hysteresis Value
011E	Thermistor Threshold	R / W	0 - 3675	Non-Volatile	Thermistor Probe Upper Threshold
0120	Number of truck compartments	R	—	Dynamic	Number of compartments on truck
0121	Stop logging dome out events	R / W	0 - 1	Dynamic	Disable dome out event logging
0122	TIM info logged	R	—	Dynamic	Logged TIM info
0123	TIM size	R	—	Dynamic	Size of TIM
0124	Super TIM code	R	—	Dynamic	Super TIM code
0125	log data state	R	—	Dynamic	State of data log
0126	compare volts	R	—	Dynamic	Socket pin 5 voltage from truck connection

3.8.1. Date and Time (VIP) [Registers 000-004]

These Read/Write registers allow setting of local date and time, day light saving, and verification of the battery powered real time clock/calendar. Setting the time and date starts the clock running. All the time and date registers should be set at once with one command. Unreasonable times and dates such as the 30th of February may draw an exception response. The registers are designated as follows:

REGISTER	DEFINITION
0	Year (1992 and after)
1	Month (1-12)
2	Day (1-31)
3	Hour (0-24)
4	Minute (0-59)

NOTE: Always verify time and date after any change.

NOTE: The Intellitrol only supports the VIP Date/Time registers as read-only; use the UNIX Date/Time registers to set the Intellitrol date and time.

3.8.2. 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.

For the VIP, the version number is in the format MajorVersion * 100 (decimal) plus MinorVersion. For example, version 12.34 would be stored as 4d2 hex (1234 decimal) and version 1.2 as 78 hex (120 decimal).

For the Intellitrol, 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).

3.8.3. Scully Unit Serial Number [Register 006]

This register returns a unique 16-bit identification number programmed at the factory. Installing the debug jumper allows a write to this register to change the VIP’s serial number. If the debug jumper is not installed, writes to this register will return the illegal data address exception message. This register always reads back as zero for the Intellitrol.

3.8.4. Latest Log Pointer [Register 007]

Reading the Latest Log Pointer register returns the index of the most current VIP bypass log entry. A program could also determine the latest log entry by reading all the entries and comparing their date stamps. This register always reads back as zero for the Intellitrol.

3.8.5. 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.

Newly manufactured VIPs and Intellitrols will be shipped with the wait for TAS delay set to zero seconds.

3.8.6. Bypass Active Time [Register 009]

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

For VIP units, bypassing will stay active for at least the specified time (120 to 3600 seconds). If a truck is detected, the VIP unit will stay in bypass until the truck departs. If the truck is undetectable, this timer will end bypass when the timer expires. The bypass active minimum time is two minutes to allow ample time for thermal overfill probes to warm up and oscillate in extremely cold weather. The VIP unit looks for a dry probe signal as a truck present indication, and the two-minute warm-up guarantees operation in cold weather. It may be necessary to set the active time longer for trucks lacking overfill probes. Newly manufactured VIP’s will be shipped with Bypass Active Time set to 120 seconds.

For Intellitrol units, this register determines the maximum amount of time the unit can remain bypassed from the presentation of a 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.

Newly manufactured Intellitrols will be shipped with Bypass Active Time set to 3600 seconds (1 hour).

3.8.7. Terminal Number [Register 00A]

This Read/Write register is the desired terminal identification number (0-9999). This register is only required for DateStamp systems (See Modbus functions Write Company ID Name and Write Password) but is allowed anytime.

Newly manufactured VIP's and Intellitrols will be shipped with the terminal number set to 0.

3.8.8. 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 (currently, typically 30 milliseconds and always within a second for the VIP). 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.

Newly manufactured Intellitrol rack controllers will be shipped with the minimum response time set to 100 milliseconds.

3.8.9. Shell Checksum [Register 00C]

Computed checksum of the shell program.

3.8.10. Vehicle List Size [Register 00D]

Total number of active vehicles in the vehicle list. If the count matches the number of vehicles written, the new vehicle list is assumed to have loaded error free. Count mismatch will occur if communications errors caused rejection of one or more vehicles. When the rack controller unit calculates the vehicle list size, it will return a count of 0 to indicate the vehicle list size is invalid. The count should become valid (non-zero) within 4 seconds after the truck departs, the unit has completed calculation of the vehicle list size, and the unit has completed performing its periodic diagnostics. If the vehicle list was broadcast to all rack controller units on the Modbus using address 128 (hex), the master should check the vehicle list size on each unit. Each rack controller unit checks its vehicle list from time to time. Stuck bits or other errors in EEPROM may cause the unit to remove one or more vehicles at that time. These errors can be detected by checking the vehicle list size register periodically.

The Intellitrol does not support the Vehicle List Size register (reads will return 0 always); use the vehicle list CRC facility instead to validate the onboard vehicle list.

3.8.11. 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. On

Intellitrol units, 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 (or DateStamp control, if enabled), until the TAS does finally changes the mode.

When the TAS is not involved in the validation process, but only for vehicle list maintenance, 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 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.

3.8.11.1. VIP Mode Control Bit Assignments

MODE	OPERATION	MEANING
0	LOCAL OPERATION	Normal operation without any Modbus override. Permits trucks from the vehicle list or date from TIMs or Bypass keys.
1	REMOTE BYPASS	Will cause the unit to bypass the VIP function and act as if a bypass key had just been touched to the bypass port on the unit. When the truck departs or 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	Passive ID Mode	Disabling the VIP authorization functionality, and enabling Passive ID Mode, the terminal can now read all SuperTIM vehicle ID's without interference from the Intellitrol.
6 - 65535		Reserved, return exception response message.

3.8.12. Kernel Version [Register 00F]

Reading the Kernel Version register returns the version number of the running firmware kernel. The format of the kernel version is the same as for the shell.

The Intellitrol does not currently support a separate kernel.

3.8.13. Model Number [Register 012]

This register contains the model number of the rack controller unit. Earlier VIP firmware revisions will generate an error response message (because this register is undefined), indicating a VIP (or some other unknown type) unit is attached at the Modbus address requested. The following table lists the model numbers:

MODEL #	MODEL NAME
0	Undefined (error)
1	VIP
2	Intellitrol
3	Intellitrol SIL
4	Intellitrol 2

3.8.14. Unit Serial Number [Registers 020 - 023]

The Unit Serial Number register(s) contain the rack controller unit's 64-bit serial number. The Intellitrol uses the onboard Dallas™ clock's 48-bit serial number as the Unit Serial Number (register 20 will always read back 0)

3.8.15. 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.

3.8.16. Config-A [Register 025]

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

Some of the hardware jumpers are further "jumpered" by software control. This means that the Intellitrol firmware may selectively disable the hardware jumper function. Further, some of these software-controllable features are master-controlled by a "Features Enable Password". 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 keys to the internal Bypass Key List stored in EEPROM.

MASK (hex)	NAME	MEANING
0020	ENA_ERASE_KEYS	Boot up in “Special Operations” mode and erase the internal EEPROM Bypass Key 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 channels.
0200	CFGA_EURO8VOLT	The “European” voltage-limiting jumper is installed (in particular, JumpStart is essentially defeated).
0400	CFGA_100OHMGND	The “European” Pin-9-as-ground jumper is installed. VIP functionality is implicitly disabled.
7800		Reserved.
8000	CFGA_DEBUG	The “DEBUG” jumper is (or was) in place — This is an UNSAFE CONDITION; many safety checks are defeated.

3.8.17. 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.

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 the VIP subsystem code.
0008	ENA_GROUND	Enable the Ground-Check subsystem code.
0010	ENA_ADD_1_KEY	Boot up in “Special Operations” mode for manually adding bypass keys to the internal Bypass Key List stored in EEPROM.
0020	ENA_ERASE_KEYS	Boot up in “Special Operations” mode and erase the internal EEPROM Bypass Key List.
0040	ENA_DEADMAN	Enable the Deadman Switch subsystem code.
0080	ENA_VAPOR_FLOW	Enable the Vapor Flow sensing subsystem.
0100		Reserved
0200	STSB_ERR_EEPROM	EEPROM error
0400	STSB_ADC_TIMEOUT	ADC has timed-out
0800	STSB_CRC_SHELL	Shell CRC error
1000	STSB_BAD_CLOCK	Clock error
2000	STSB_BAD_CPU	CPU error
4000		Reserved

MASK (hex)	NAME	MEANING
8000	STSB_CRC_KERNEL	Kernel CRC error

3.8.18. Config-C [Register 027]

The Config-C more "Software System Configuration" flags.

MASK (hex)	NAME	MEANING
0001		Reserved
0002	ENA_INTL_SHORT	Enable the "Shorts" tests.
F800		Reserved.

3.8.19. Config-D [Register 028]

The Config-D register is reserved for even more future "Configuration" flags; all reads will return 0.

3.8.20. RAM Size [Register 029]

The RAM Size register returns the size (in "KB") of the unit's onboard RAM space.

3.8.21. FLASHRAM Size [Register 02A]

The FLASHRAM Size register returns the size (in "KB") of the unit's onboard FLASHRAM. The FLASHRAM holds the unit's firmware ("Kernel" and "Shell" executable code).

3.8.22. EEPROM Size [Register 02B]

The EEPROM Size register returns the size (in "KB") of the unit's onboard EEPROM non-volatile memory storage. The EEPROM is divided up into multiple "partitions", and used to store assorted semi-permanent information, such as the Modbus Minimum Delay size (see register 00B above), the unit's DateStamp code ("name" and "password"), the Vehicle List, and so on. The EEPROM partition information (sizes) can be obtained via the EEPROM Information block of registers (see registers 0A0-0BF below).

3.8.23. Modbus Message Size [Register 02C]

The Modbus Message Size register contains the unit's maximum supported Modbus Protocol Message size (in bytes). The unit will always support at least 64-byte messages. (This value can only be changed by rebuilding the unit's firmware and has an implicit maximum supportable size of 255 bytes.)

3.8.24. Number of probes [Register 02D]

This register is valid only on the SIL Intellitrol

This register contains the number of compartments in the current truck. If the truck uses two wire probes this register will report the number of compartments based on the condition of jumper J3 in the Intellitrol.
For 5 wire the Intellitrol will attempt to calculate the number of compartments by send a pulse and waiting for it to return. After it returns the diagnostic line is read. This should indicate the last probe +1 has a failure. So, the number of the compartments is the number of probes - 1. If the pulse does not return it assumes that truck is wet and this register reports 0x55.
IntelliCheck setup as a 5-wire optical does not have a diagnostic line so it also may report back as a 0x55. However, if the Super T.I.M. was setup by the truck builder as having an IntelliCheck this register will contain 0xAA.

3.8.25. Factory Enable Features [Register 02E]

This refers to the 026 register. That register displays the features that are currently enabled. To be enabled they must first be enabled at the factory. Then with them being enable and the jumpers in the enable position register 26 will indicate which features are in use. This register displays the features that are enabled at the factory.

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-Check subsystem code.
0010	ENA_ADD_1_KEY	Boot up in "Special Operations" mode for manually adding bypass keys to the internal Bypass Key List stored in EEPROM.
0020	ENA_ERASE_KEYS	Boot up in "Special Operations" mode and erase the internal EEPROM Bypass Key List.
0040	ENA_DEADMAN	Enable the Deadman Switch subsystem code.
0080	ENA_VAPOR_FLOW	Enable the Vapor Flow sensing subsystem.

3.8.26. Reference Volt [Register 030]

The Reference Volt register contains the last selftest value of the unit's internal reference voltage (in millivolts). Ideally, this value should be 1000 (1.000 volts); the actual value read is the unit's internal A/D converter calibration, used to self-calibrate the unit, especially for Five Wire Optic diagnostic readings (i.e., using the "diag" wire to determine the "wet" tank by measuring the voltage drop due to paralleled dropping resisters on the "probe bus").

The Reference Volt register is one of the "Diagnostic Voltage Registers" and is not intended for production TAS usage. The returned value is a semi-static one reflecting the last time the internal selftest diagnostics executed.

3.8.27. Raw 13V Supply Voltage [Register 031]

This register contains the A/D measured voltage of the raw 13 Volt supply (in millivolts).

The Raw 13V Supply Voltage register is one of the "Diagnostic Voltage Registers" and is not intended for production TAS usage. The returned value is a semi-static one reflecting the last time the internal selftest diagnostics executed.

3.8.28. Probe Bias [Register 032]

The Probe Bias register contains the internal “probe bias” switching voltage used by the comparators when driving Two-Wire Thermistor probes (in millivolts). This internal operating voltage switch point is determined by the bias jumper and is nominally 3.5 volts (register value 3500) or 3.8 volts (register value 3800) depending on the jumper position.

The Probe Bias register is one of the “Diagnostic Voltage Registers” and is not intended for production TAS usage. The returned value is a semi-static one reflecting the last time the internal selftest diagnostics executed.

3.8.29. Optic-Output Pulse [Register 033]

The Optic-Output Pulse register contains the measured voltage level (in millivolts) presented as the “optic output” pulse for Five-Wire Optic probe sets.

The Optic-Output Pulse register is one of the “Diagnostic Voltage Registers” and is not intended for production TAS usage. The returned value is a semi-static one reflecting the last time the internal selftest diagnostics executed.

3.8.30. Channel Noise [Register 038-03F]

The Channel Noise registers contain the measured open-circuit (no probes connected) channel noise levels (in millivolts) relative to the channel’s 10-Volt Rail voltage. The Intellitrol measures both noise levels above and below the nominal channel rail voltages. The Channel Noise register returns the larger of the two noise magnitudes -- a positive value is a noise level above the rail, while a negative (treating the register value as a signed 16-bit integer) value indicates a noise level below nominal rail voltage.

There is one Channel Noise register for each of the Intellitrol’s eight probe channels, register 038 is for Channel 1, ..., register 03F is for Channel 8. The value of the first two Channel registers is undefined if the unit is configured for “USA Trucks” 6-compartment operation.

The Channel Noise registers are part of the “Diagnostic Voltage Registers” and are not intended for production TAS usage. The returned values are semi-static ones reflecting the last time the internal selftest diagnostics executed.

3.8.31. Channel 10-Volt Rail [Register 040 - 047]

The Channel 10-Volt Rail registers contain the measured open-circuit (no probes connected) 10-volt channel voltages (in millivolts) with the “JumpStart” 20-volt power disabled. The “10-volt” power supply is the normal operational power supplied to a connected probe.

There is one Channel 10-Volt Rail register for each of the Intellitrol’s eight probe channels, register 040 is for Channel 1, ..., register 047 is for Channel 8. The value of the first two Channel registers is undefined if the unit is configured for “USA Trucks” 6-compartment operation.

The Channel 10-Volt Rail registers are part of the “Diagnostic Voltage Registers” and are not intended for production TAS usage. The returned values are semi-static ones reflecting the last time the internal selftest diagnostics executed.

3.8.32. Channel 20-Volt Rail [Register 048 - 04F]

The Channel 20-Volt Rail registers contain the measured open-circuit (no probes connected) 20-volt channel voltages (in millivolts) with the "JumpStart" 20-volt power enabled. The "20-volt" power supply is the "JumpStart" power provided as the initial cold-thermistor fast-warm-up power provided to a connected probe for the first 20 seconds or so of probe activity. The JumpStart power is automatically turned off as soon as the probes start oscillating, the probes are identified as either 2-wire or 5-wire optic probes, or 20 seconds have elapsed since the probes were first connected.

There is one Channel 20-Volt Rail register for each of the Intellitrol's eight probe channels, register 048 is for Channel 1, ..., register 04F is for Channel 8. The value of the first two Channel 20-Volt Rail registers is undefined if the unit is configured for "USA Trucks" 6-compartment operation.

The Channel 20-Volt Rail registers are part of the "Diagnostic Voltage Registers" and are not intended for production TAS usage. The returned values are semi-static ones reflecting the last time the internal selftest diagnostics executed.

3.8.33. Channel Last Voltage [Register 050 - 057]

The Channel Last Voltage registers contain the last measured channel voltage levels (in millivolts) for each of the 8 probe channels.

There is one Channel Last Voltage register for each of the Intellitrol's eight probe channels, register 050 is for Channel 1, ..., register 057 is for Channel 8. The value of the first two Channel Last Voltage registers is undefined if the unit is configured for "USA Trucks" 6-compartment operation.

The Channel Last Voltage registers are part of the "Diagnostic Voltage Registers" and are not intended for production TAS usage. The returned values are dynamic ones reflecting the last time the rack controller read the Analog-to-Digital converted voltages of the eight channels. In general, these values depend on the exact state of the rack controller, what type of probes are connected, various active and passive self-tests, and a host of other factors.

3.8.34. Clock Status [Register 060]

This register contains the current state of attempts or a successful reading of the on-board Dallas clock (updated at power on and truck departure). This register can be one of the following:

VALUE	NAME	DEFINITION
0	CLOCK OK	Clock read OK
1	CLOCK ABSENT	Chip not present
2	CLOCK CRC	CRC-8 read error
3	CLOCK FAMILY	Not a clock chip
4	CLOCK RANGE	Time out of range
5	CLOCK NOT RUNNING	Clock not running
6	CLOCK NOISY	Clock unreadable
7	CLOCK DEFAULT	Clock Default 1970
8 - 65535		Reserved

3.8.35. Relay State [Register 061]

The Relay State register contains the rack controller's relay state byte(s).

For the Intellitrol, the Main processor's relay state is in the low-order byte of the register, and the Backup processor's relay state (as interpreted by the Main processor) is in the high-order byte of the register. The Intellitrol relay state bytes are bitmapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
01	RELAY_CLOSED	The relay contacts are closed.
02	RELAY_WANTED	The relay contacts *should be* closed.
10	RELAY_NOTSURE	The relay contacts are probably in a transition between the closed and open states. Alternatively, the contact sensor circuitry is broken.
20	RELAY_BROKEN	The relay is broken — the relay contacts are not closed, and they should be.
40	RELAY_SHORTED	The relay is shorted — the relay contacts are closed, and they should be open.
8C		Reserved

3.8.36. EEPROM State [Register 062]

The EEPROM State register returns the rack controller's current EEPROM (non-volatile store) unit status.

The Intellitrol's EEPROM status register is split into two bytes. The low-order byte is the active EEPROM status, and the high-order byte is the valid-partition flags byte. The EEPROM status is bitmapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
01	EE_DATAERROR	The EEPROM has suffered one or more read or write byte data errors.
02	EE_WRITETIMEOUT	The EEPROM has suffered one or more time-out errors writing a data byte.
04	EE_FORMAT	The EEPROM needs "formatting" (the "Home" block is invalid).
08	EE_BADHOME	The EEPROM "Home" block is invalid (the pattern bytes appear correct, but the Home Block CRC-16 is invalid).
40	EE_ACTIVE	Reserved.
80	EE_BUSY	Reserved.
30		Reserved

The Intellitrol's EEPROM valid-partition flags byte is bitmapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
01	EEV_HOMEBLK	Home block/partition is valid/accessible.
02	EEV_BOOTBLK	Boot block/partition is valid/accessible.
04	EEV_CRASHBLK	Crash block/partition is valid/accessible.
08	EEV_SYSBLK	System Info partition is valid/accessible
10	EEV_LOGBLK	Event Log partition is valid/accessible.
20	EEV_KEYBLK	Bypass Key list partition is valid/accessible.
40	EEV_TIMBLK	Truck Id Module List partition is valid/accessible.
80		Reserved.

3.8.37. Acquire State [Register 064]

The Acquire State real-time register contains the current state of the finite state machine used to determine which type of probe is present. This register is only valid when the Main State is in the Acquire State. This register is for Scully use only, and can be one of the following:

VALUE	**NAME**	**DEFINITION**
0	Idle	Waiting for something to happen
1	Optic 5-Wire	Detecting truck with 5-wire optic probes
2	Optic 2-Wire	Detecting truck with 2-wire optic probes
3	Thermistor	Detecting truck with Thermistor probes
4	Gone	Truck gone, perform periodic diagnostics
5 - 65535		Reserved

3.8.38. Probe Try State [Register 065]

The Probe Try State real-time register contains the current state of the finite state machine used to determine which type of probe is present. This register is for Scully use only, and can be one of the following:

VALUE	**NAME**	**DEFINITION**
0	No Type	No probe test currently in progress
1	Optic 5-Wire	Currently testing for optic 5-wire probes
2	Optic 2-Wire	Currently testing for optic 2-wire probes
3	Thermistor	Currently testing for thermistor probes
4 - 65535		Reserved

3.8.39. Optic 5-Wire State [Register 066]

The Optic 5-Wire State real-time register contains the current state of the finite state machine used to keep track of the progress of pulsing 5-wire optic probes. This register is for Scully use only, and can be one of the following:

VALUE	NAME	DEFINITION
0	No5 Test	No test in progress
1	Pulsed	Pulse emitted, awaiting return pulse
2	Echoed	Echo received
3	Diag	Performing diagnostic, green wire present
4 - 65535		Reserved

3.8.40. Two Wire State [Register 067]

The Two Wire State real-time register contains the current state of the finite state machine used to keep track of the progress of 2-wire probes. This register is for Scully use only, and can be one of the following:

VALUE	NAME	DEFINITION
0	No Test 2W	No test in progress
1	ShortChk 2W	Checking probes for shorts, opens, etc.
2	ShortFail 2W	Probes failed test, waiting for truck disconnect
3	Pulsing 2W	Pulses detected, determining how many
2 - 65535	Reserved	

3.8.41. Service-A [Register 069]

The Service-A register (also known as the "iambroke" flags) contains additional error/status flags. Any Service-A flag bit set results in the rack controller entering a "Fault" state. The Service-A register is bitmapped as follows:

MASK (hex)	NAME	DEFINITION
0001	RAW_FAULT	The power-supply "Raw 13 volts" is out of spec.
0002	REF_VOLT_FAULT	The "Reference" volt is out of spec.
0004	PROBE_BIAS_ERROR	The "Probe Bias" voltage is out of spec.
0008	NOISE_FAULT	One or more probe channel's noise is out of spec.
0010	TOL10V_FAULT	One or more probe channel's 10-volt Rail levels is out of spec.
0020	TOL20V_FAULT	One or more probe channel's 20-volt Rail levels is out of spec.
0040	TOL5WO_FAULT	Five-wire-optic pulse voltage is out of spec.
0080		Reserved.
0100		Reserved.

MASK (hex)	NAME	DEFINITION
0200		Reserved.
0400		Reserved.
0800		Reserved.
1000	JUMPER_CHANGE	One or more hardware jumpers have changed.
2000	CABLE_SHORT	Channel-to-channel short in cable.
4000	ADC_FAULT	Can’t init the 68HC16’s A/D converters.
8000	BAD_TPUvsADC	Reserved.

3.8.42. Service-B [Register 06A]

The Service-B register (also known as the “iamsuffering” flags) contain further error/status flags. Any Service-B flag bit set results in the rack controller entering a “Fault” state. The Service-B register is bitmapped as follows:

MASK (hex)	NAME	DEFINITION
0001		Reserved.
0002		Reserved.
0004		Reserved.
0008		Reserved.
0010		Reserved.
0020		Reserved.
0040		Reserved.
0080		Reserved.
0100		Reserved.
0200		Reserved.
0400		Reserved.
0800		Reserved.
1000	HARD_WIRE_FAULT	All relays’ contacts are shorted; the rack control is effectively shorted and out-of-active-control. (Quite likely, the short is external to the rack controller.)
2000		Reserved.
4000		Reserved.
8000		Reserved.

3.8.43. Service-C [Register 06B]

The Service-C register is reserved for future error/status information.

3.8.44. Combined VIP Status [Register 06C]

The Combined VIP Status register contains additional information of the VIP subsystem operation. The low-order byte is the “VIP” status, and the high-order byte is the DateStamp status code. The VIP status byte is bitmapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
01	BVF_UNAUTH	TIM serial number is unauthorized.
02	BVF_TIMCRC	Can’t read the TIM serial number — CRC error.
04	BVF_TIMFAMILY	The TIM family is not the DS1993 family code.
08	BVF_TIMABSENT	The TIM is not responding to “Presence” polls.
10	BVF_TASDENY	TIM unauthorized by TAS control.
20	BVF_TASDELAY	Waiting for TAS to authorize/unauthorize the TIM.
40	BVF_DSNOAUTH	DateStamp did not authorize TIM.
80	BVF_DSERROR	Error reading DateStamp info from TIM.

The DateStamp status code byte may contain values as follows:

VALUE	**NAME**	**DEFINITION**
5	Dwfail	Dallas™ general write failure.
6	Drnopresence	No “Presence” pulse on Dallas™ read.
10	Dramcrc	Dallas™ RAM CRC read error.
21	Filenotfnd	Touch chip file not found.
22	Tchipformat	Dallas™ Touch-Chip not formatted — no “AA” code.
28	Filenotopen	Read/Write error — file not open.
30	Tchiplenbyt	Dallas™ Touch-Chip length byte in error (over 29).
31	Tchipwritelock	Dallas™ Touch-Chip not writable (software or hardware).
32	TchipEOF	End of File reading Dallas™ Touch-Chip file.
33	Dsbadchar	Invalid character in DateStamp file.
34	Dsexpired	DateStamp file expired.

3.8.45. Ground Status [Register 06D]

The Ground Status register contains the rack controller’s Ground Fault Detection subsystem’s current status in the low-order byte; the high-order byte is reserved and reads back as 0. The Ground status byte is bitmapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
01	GND_FAIL	Ground Fault detected.
02		Reserved.

MASK (hex)	NAME	DEFINITION
04		Reserved.
08		Reserved.
10	GND_NO_TEST	No test performed (Ground fault detection may be disabled, or COMM_ID may be in use by other subsystems).
20	GND_SHORTED	The Ground Bolt sense line (Pin 9) is shorted to ground.
40	GND_HARD_FAULT	The Ground Fault Detection circuitry failed.
80	GND_NO_TRIAL	Ground Fault test aborted.

3.8.46. Shorts Test Enable/Disable flag [Register 070]

This software selectable flag allows the user to enable (1) or disable (0) the test performed to find the short condition between the probes. The default is ON.

3.8.47. Debug Pulse Enable Flag [Register 071 - 74]

These registers allow the user to output specific code to the debug connector. The code is pertinent to a function performed by the software. The default is OFF.

3.8.48. Debug Pulse on Failure Enable Flag [Register 075 - 78]

These registers allow the user to output specific code to the debug connector. The code is pertinent to a failure of a function performed by the software. The default is OFF.

3.8.49. Software Feature Enable [Register 079 – 7A]

These registers are for enabling software features.

3.8.50. VIP Option Modbus register [Register 07B]

When this register is clear (0), the Intellitrol functions as old version (1.0.5). When this register is set to 1 and the system has VIP disabled, the Intellitrol will allow any TIM serial number to be considered found in the list of valid serial numbers, although no serial numbers will need be stored in the controller VIP's list.

- To enable this feature, send the following Modbus command:
  0106007B0001

- To disable this feature, send the following Modbus command:
  0106007B0000

- To read back the register value, use the following Modbus command:
  0103007B0002

The Intellitrol will not include the VIP in the PERMIT decision but the VIP leds AUTHERIZED, UNAUTHORIZED and STANDBY will function as normal.

The following describes the controller's operation for various situations.

1) Prior to truck connection, the VIP STANDBY led is ON, NON-PERMISSIVE (big red light is ON). Modbus TIM serial number registers (10A-10C) will contain 0.
2) When Intellitrol connects to truck, the controller attempts to read the TIM.

   a) If the serial number can be read, the VIP AUTHORIZED led is ON, and the Controller will be PERMISSIVE (big green light ON), if the overfill sensors are dry and grounding is ok. The serial number is stored in Modbus TIM serial number registers.

   b) If the serial number cannot be read correctly, possibly a bad noisy connection, the VIP UNAUTHORIZED led will be ON, and the Controller will be PERMISSIVE (big green light on) if the overfill sensors are dry and grounding is ok. The Modbus TIM serial number registers will contain FFFFFFFFFFFF. Driver should reconnect plug.

   c) If the TIM cannot be detected, possibly a broken wire, the VIP STANDBY led will be ON, and the Controller will be PERMISSIVE (big green light on) if the overfill sensors are dry and grounding is ok. Modbus TIM serial number registers will contain 0. Driver should reconnect plug.

3) In situations b & c above, a VIP Bypass operation will not be required to PERMIT the Intellitrol.

4) The cable will be considered disconnected, when the Intellitrol can no longer detect a truck signal of any type. After a 5 seconds disconnection, the Modbus TIM serial number register will be written 0.

3.8.51. Resistive Ground reference value [Register 7C]

This provides a method for setting the highest resistance level acceptable for a good vehicle ground connection when automatic ground type detection is selected via J8 and a ground bolt is not detected. The accepted register values and corresponding resistance levels are shown below.

Value	Resistance level
0	Near 2k, default value which matches Intellitrol
1	Near 100, recommended by API
2	Near 5k
3	Near 10k, specified by EN 13922

3.8.52. Enable Good Ground display register [Register 7E]

Register 7E was added to Intellitrol firmware 1.14. When this register is clear (0), the Intellitrol will function as the same as previous versions of the firmware. This register is only valid on Intellitrols with the ground proofing feature enabled. When this register is set to a 1, while the Intellitrol is trying to determine the Truck probe type, the PERMIT leds will flash at a ½ second rate if the ground is good.

- To enable this feature, send the following Modbus command:
  0106007E0001

- To disable this feature, send the following Modbus command:
  0106007E0000

- To read back the register value, use the following Modbus command:
  0103007E0002

3.8.53. 5-Wire Compartment Count Display Time [Registers 07F]

Register 7F was added to support Intellitrol2's 5-wire compartment count. This register is only valid on Intellitrol2. This register holds the time in seconds which the compartment count should be flashed on initial 5-wire vehicle connection. Only values between 0 and 31 and 0xFF are allowed. A value of 0xFF disables the flashing of the display.

3.8.54. Active Deadman Enabled [Registers 080]

Enable Active Deadman 0: OFF, 1: ON Registers 80 - 83 were added with Intellitrol firmware version 1.6.35 to provide an active deadman function. Register 80 enables the function with any non-zero value, a zero disables the function.

3.8.55. Active Deadman Max Open Time [Registers 081]

Register 81 holds the maximum number of seconds the deadman switch can be open prior to a fault being reported. This time is used for both the active and standard deadman functions. System permit, when the deadman function is active, is not allowed with a deadman fault. Only values 1 - 30 are allowed; the default is 3.

3.8.56. Active Deadman Max Close Time [Registers 082]

Register 82 holds the maximum number of seconds the deadman switch can be closed prior to a fault being reported. This time is used only for the active deadman function. System permit, when the deadman function is active, is not allowed with a deadman fault. Only values 10 - 600 are allowed; the default is 120.

3.8.57. Active Deadman Warning time [Registers 083]

Register 83 holds the number of seconds prior to a deadman fault being reported that a warning will be issued. This time is used only for the active deadman function. This value is used with register 82 to warn the deadman user that the switch must be opened to prevent a fault. This register value must be at least 15 less than register 82. If the register 82 value is less than 20 the register 83 value is ignored. Only values 10 - 60 are allowed; the default is 15.

3.8.58. Software Feature Enable Unload Terminal [Registers 084]

Register 84 enables unload terminal mode. This allows the Intellitrol to check the time and date stamp of when the truck was loaded and not allow a permit if the fuel has been in the truck too long. To configure the unit for unload mode a 1 needs to be written to Modbus register 0x0084. The maximum time to allow for an unload is set by writing the time, in minutes, to Modbus register 0x0085. Both values only need to be written once and will be persistent through power cycles. If the unload maximum time is exceeded the unit will not permit without being bypassed. The unload time exceeded can be detected by bit 9 (0x0200) of Modbus register 0x68 being set.

The unload terminal can also be programmed to not permit if the loaded fuel type does not match the allowable fuel type. To enable this feature, a bit mask of the compartments to check is written to Modbus register 0x88. Bit 0 is set if compartment 1 is to be checked, bit 1 for compartment 2 and so on. This value is also persistent through power cycles. If any compartment fuel type doesn't match bit 11 (0x0800) of register 0x0068 will be set. This is a bypassable condition. To determine which compartment type does not match a bit mask of miss matched types can be read from Modbus register 0x6B. If compartment 1 type does not match bit 0 will be set, if compartment 2 type does not match bit 1 will be set and so on.

3.8.59. SuperTim Max Unload Time [Registers 085]

Register 85 holds the maximum time to allow for an unload. If the unload maximum time is exceeded while in Unload Mode, the unit will not permit without being bypassed. The unload time exceeded can be detected by bit 9 (0x0200) of Modbus register 0x68 being set.

3.8.60. SuperTim Certificate Date Enable Mask [Registers 086]

Register 86 holds the value to select which certificates to compare with the SuperTIM data. The SuperTIM can store 5 certificate expiration dates. The Intellitrol will not permit if any of the selected certificates are expired. This fault is bypassable. Certificate 1 is the vapor tightness certificate, certificate 2 is the safe loading pass certificate. Certificates 3 to 5 are open for user definition. To enable this feature, a bit mask of the certificates to check is written to Modbus register 0x86. Bit 0 is set if certificate 1 is to be checked, bit 1 for certificate 2 and so on. This value is persistent through power cycles. The selected certificates expiration dates are compared to the current date to assure the certificate has not expired. If any have expired the unit will be non-permissive. If any selected certificate has expired bit 6 (0x0040) of Modbus register 0x68 will be set. To determine which certificate has expired a bit mask of expired certificates can be read from Modbus register 0x6F. If certificate 1 is expired bit 0 will be set, if certificate 2 is expired bit 1 will be set and so on.

3.8.61. Software Feature Enable Compartment Count Check [Registers 087]

The SuperTIM memory has the number of compartments stored in the builder data. The number of sensors detected by the Intellitrol can be compared to the number of compartments in the truck and permit will not be allowed if the counts don’t match. This is a bypassable fault. To enable this feature, 1 should be written to Modbus register 0x87. This value is persistent through power cycles. If this fault is detected bit 10 of Modbus register 0x68 will be set. This check is bypassed if the SuperTIM data indicates that an Intellicheck is installed on the truck.

3.8.62. SuperTim Fuel Type Check Mask [Registers 088]

Register 88 holds the value to select which compartments to compare the fuel type with the SuperTIM data. Bit 0 is set if compartment 1 is to be checked, bit 1 for compartment 2 and so on. This value is persistent through power cycles.

3.8.63. Software Feature Enable Auto Write Fuel Type Flag [Registers 089]

Register 89 enables automatically writing the fuel type to the SuperTIM data on connection with a truck.

3.8.64. SuperTim Default Fuel Type [Registers 08A – 08B]

Register 8A - 8B hold the value of the default fuel type to write to the SuperTIM data. Fuel types are saved as product family codes as defined by PIDX.

3.8.65. EEPROM Base Address [Registers 0A0 - 0A1]

The EEPROM Base Address registers return the 32-bit address of the first byte of the EEPROM non-volatile memory storage.

3.8.66. Home Block Size [Register 0A2]

The Home Block Size register returns the size (in bytes) of the EEPROM "Home" block or partition. If the rack controller's EEPROM is inaccessible, then this register is undefined.

3.8.67. Home Block Offset [Register 0A3]

The Home Block Offset register contains the EEPROM-relative address (or "offset") from the start of the EEPROM to the start of the Home Block. The Home Block always starts at offset 0.

3.8.68. Boot Block Size [Register 0A4]

The Boot Block Size register returns the size (in bytes) of the EEPROM "Boot" block or partition. If the rack controller's EEPROM is inaccessible, or the EEPROM "Home" block is invalid, then this register is undefined.

3.8.69. Boot Block Offset [Register 0A5]

The Boot Block Offset register contains the EEPROM-relative address (or "offset") from the start of the EEPROM to the start of the Boot Block. If the rack controller's EEPROM is inaccessible, or the EEPROM "Home" block is invalid, then this register is undefined.

3.8.70. Crash Block Size [Register 0A6]

The Crash Block Size register returns the size (in bytes) of the EEPROM "Crash" block or partition. If the rack controller's EEPROM is inaccessible, or the EEPROM "Home" block is invalid, then this register is undefined.

3.8.71. Crash Block Offset [Register 0A7]

The Crash Block Offset register contains the EEPROM-relative address (or "offset") from the start of the EEPROM to the start of the Crash block. If the rack controller's EEPROM is inaccessible, or the EEPROM "Home" block is invalid, then this register is undefined.

3.8.72. System Info Block Size [Register 0A8]

The System Info Block Size register returns the size (in bytes) of the EEPROM "System Information" block or partition. If the rack controller's EEPROM is inaccessible, or the EEPROM "Home" block is invalid, then this register is undefined.

3.8.73. System Info Block Offset [Register 0A9]

The System Info Block Offset register contains the EEPROM-relative address (or “offset”) from the start of the EEPROM to the start of the System Information Block. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

3.8.74. Event Log Block Size [Register 0AA]

The Event Log Block Size register returns the size (in bytes) of the EEPROM “Event Log” block or partition. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

The number of entries configured/supported in the Event Log can be determined by dividing the size of the Event Log Block partition by the size of an Event Log Entry (32 decimal).

3.8.75. Event Log Block Offset [Register 0AB]

The Event Log Block Offset register contains the EEPROM-relative address (or “offset”) from the start of the EEPROM to the start of the Event Log partition. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

3.8.76. Bypass Key List Size [Register 0AC]

The Bypass Key List Size register returns the size (in bytes) of the EEPROM “Bypass Key List” partition. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

The number of entries configured/supported in the Bypass Key List can be determined by dividing the size of the Bypass Key List partition by the size of a Bypass Key Entry (8 decimal).

3.8.77. Bypass Key List Offset [Register 0AD]

The Bypass Key List Offset register contains the EEPROM-relative address (or “offset”) from the start of the EEPROM to the start of the Bypass Key List partition. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

3.8.78. Vehicle List Size [Register 0AE]

The Vehicle List Size register returns the size (in bytes) of the EEPROM “Vehicle List” partition. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

The number of TIMs (Truck Identification Modules) configured/supported in the Vehicle List can be determined by dividing the size of the Vehicle List by the size of a Vehicle List Entry (6 decimal).

3.8.79. Vehicle List Offset [Register 0AF]

The Vehicle List Offset register contains the EEPROM-relative address (or “offset”) from the start of the EEPROM to the start of the Vehicle List Offset. If the rack controller’s EEPROM is inaccessible, or the EEPROM “Home” block is invalid, then this register is undefined.

3.8.80. “gsoft” Count [Register 0E0]

The “gsoft” Count register returns the total count of “Soft” GRNDCHCK COMM_ID failures. It has been empirically observed that “occasionally” the COMM_ID bit in the PORTMC register fails to set when the software sets the bit. COMM_ID usually sets on the second try. The “gsoft” counter in incremented when the first try fails but the second try succeeds. As of this writing, this problem is fervently believed to be a problem with the Motorola MC68HC16Y1 microprocessor, and not a problem with the software or board hardware.

The “gsoft” register applies to the Intellitrol version 0.5.D0 and later Beta releases, and to the version 1.x production releases.

3.8.81. “ghard” Count [Register 0E1]

The “ghard” Count register returns the total count of “Hard” GRNDCHCK COMM_ID failures. It has been empirically observed that “occasionally” the COMM_ID bit in the PORTMC register fails to set when the software sets the bit. COMM_ID usually sets on the second try. The “ghard” counter in incremented when the both the first and second tries fail. As of this writing, this problem is fervently believed to be a problem with the Motorola MC68HC16Y1 microprocessor, and not a problem with the software or board hardware.

The “ghard” register applies to the Intellitrol version 0.5.D0 and later Beta releases, and to the version 1.x production releases.

3.8.82. 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 command. 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 rack controller units are shipped from the factory pre-initialized with a running (battery-backed where allowed) clock to maintain the time. The TAS system may choose to implement a “different” time base (e.g., local time), as long as it is consistent in the usage and interpretation of the time base, and as long as the chosen values appear correct to the rack controller firmware (which “sanity-checks” the date to appear between years 1992 and 2050 GMT). GMT-based is strongly encouraged. The Intelliview prototypical TAS provided by Scully Signal 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.

3.8.83. 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

3.8.84. 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, Intellitrol, or whatever.
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	The "Debug" jumper is installed.
0200		Reserved.

MASK (hex)	NAME	DEFINITION
0400	STSA_VAPOR_FLOW	Reserved.
0800		Reserved.
1000	STSA_DED_CLOSED	The Deadman Switch is closed.
2000		Reserved.
4000		Reserved.
8000	STSA_INTELLICHECK	Reserved.

3.8.85. 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, and the STSB_ERR_VOLTAGE bit is set, then the TAS might examine the Service-A flags to see which voltage error(s) is(are) set. If, for further example, the Service-A flag NOISE_FAULT was set, the TAS might then examine the Channel Noise registers to see which channel or channels are suffering noisy power levels.

The Intellitrol Status-B register is bit-mapped as follows:

MASK (hex)	NAME	DEFINITION
0001		Reserved.
0002	STSB_ERR_EEPROM	One or more errors have occurred with the EEPROM.
0004	STSB_ADC_TIMEOUT	One or more A/D Converter errors have occurred.
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	Reserved.
0040		Reserved.
0080	STSB_CRC_KERNEL	The Kernel CRC-16 value is bad.
0100	STSB_ERR_VOLTAGE	One or more voltages are at fault. See Service-A.
0200		Reserved.
0400	STSB_TIMDATA	Reserved.
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.

3.8.86. Status-O [Register 106]

The Status-O register "maps" the first sixteen Output 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.

The Intellitrol Status-O register is bit-mapped as follows:

MASK (hex)	**NAME**	**DEFINITION**
0001		Reserved.
0002		Reserved.
0004		Reserved.
0008		Reserved.
0010	STSO_5WIRE_PULSE1	"Optic Output" was just pulsed.
0020	STSO_5WIRE_PULSE2	"Optic Output" was recently pulsed (within past second)
0040	STSO_5WIRE_ECHO1	"Optic Input" pulse just received.
0080	STSO_5WIRE_ECHO2	"Optic input" pulse recently received.
0100	STSO_COMM_VEHICLE1	Truck communications just occurred.
0200	STS0_COMM_VEHICLE2	Truck communications recently occurred.
0400		Reserved.
0800		Reserved.
1000		Reserved.
2000		Reserved.
4000		Reserved.
8000		Reserved.

3.8.87. Status-P [Register 107]

The Status-P register "maps" the second sixteen Output 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.

The Intellitrol release 1.0 shell does not use the Status-P bits; reads return 0.

3.8.88. Main State [Register 108]

The Main State real-time register contains the current state of the main finite state machine used to determine whether a truck is present. This register is for Scully use only, and can be one of the following:

VALUE	**NAME**	**DEFINITION**
0	Idle	Waiting for something to happen, periodic background self-test diagnostics
1	Acquire	Deciding which probe type is present
2	Active	Active probes or bypass key detected

3	Gone	Truck has disconnected, perform self-test diagnostics
4	Fini	Final post-truck clean-up, transit to “Idle” state
4 - 65535		Reserved

3.8.89. Truck Type State [Register 109]

This real-time register contains the current state of the finite state machine used to determine the type of truck connected. This register is for Scully use only, and can be one of the following:

VALUE	NAME	DEFINITION
0	Unknown	A truck is out there, but do not know what kind
1	Thermistor	Thermistor probe type is present
2	Optic 2-Wire	2-wire optic probe type is present
3	Optic 5-Wire	5-wire optic probe type is present
4	Departed	5 –wire Truck gone, performing periodic diagnostics
5	GONE	2–wire Truck gone, performing periodic diagnostics
6 - 65535		Reserved

NOTE: In the case of a mixed thermistor/optic 2-wire truck, the Truck Type State will be Thermistor.

3.8.90. 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 serial number, a value of FFFFFFFFFFFF 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	Truck S/N	Truck S/N	Truck S/N	Truck S/N LSB

3.8.91. 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

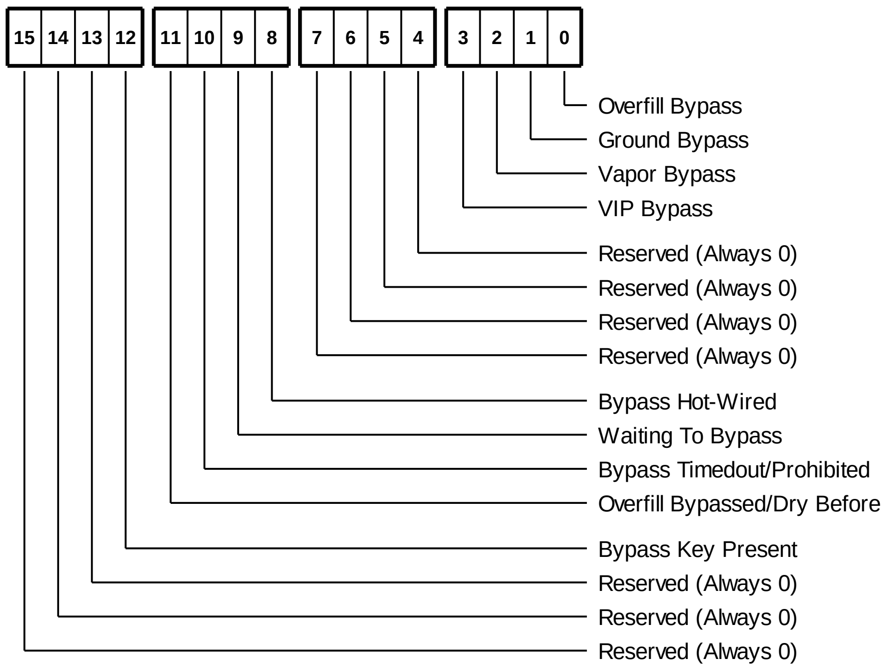
3.8.91.1. Probe States

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	Cold	Cold thermistor probe (not oscillating).
4	Hot	Warm thermistor probe (not oscillating).
5	Open	Bad probe on truck, channel/probe open
10	Fault	Random or unspecified fault (e.g., oscillations around 7 volts)
11	Grounded	Channel/Probe shorted to ground
12	Shorted	Channel/Probe shorted to another probe or other power source
13 - 65535		Reserved

3.8.92. 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 Timedout/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 "dry" for long enough that the unit will not allow an overfill bypass. This is predicated upon the idea that overfill bypass is to "get around" faulty probes, and that once the probes are believed to work (are "dry" for "long enough" to think they work properly), that any further overfill bypass will result in a fuel spill. The truck must disconnect.
- The "Bypass Key Present" flag indicates that a bypass key is currently present and being read.

3.8.93. 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:

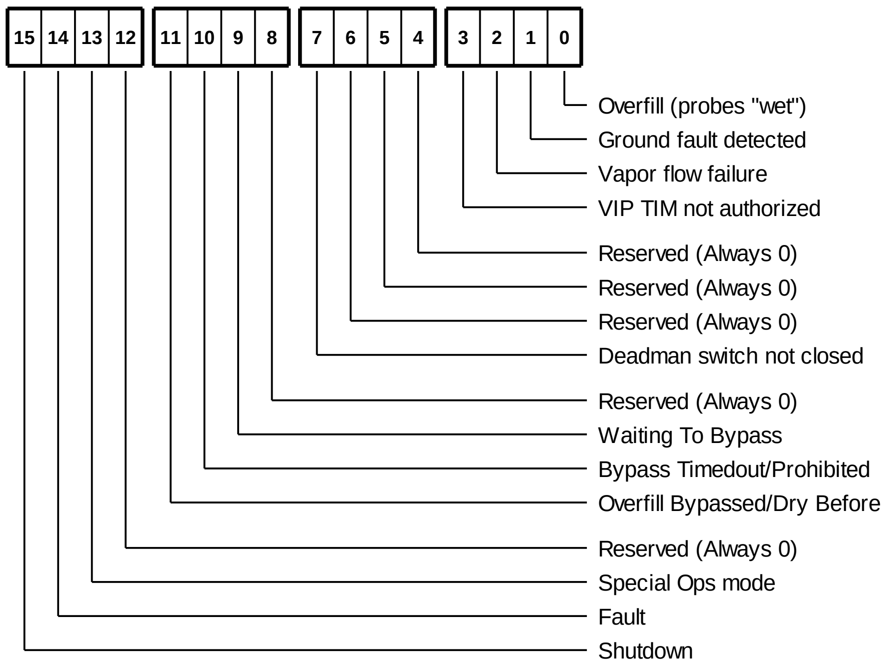
MSB	LSB	MSB	LSB	MSB	LSB
Register 116		Register117		Register 118	
Bypass S/N MSB	Bypass S/N	Bypass S/N	Bypass S/N	Bypass S/N	Bypass S/N LSB

3.8.94. 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.

3.8.95. 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.
- The “Bypass Timedout/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 “dry” for long enough that the unit will not allow an overfill bypass. This is predicated upon the idea that overfill bypass is to “get around” faulty probes, and that once the probes are believed to work (are “dry” for “long enough” to think they work properly), that any further overfill bypass will result in a fuel spill. 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 (and, seems to respond to trucks connecting, probes oscillating, etc.)

3.8.96. Latest Log pointer [Register 11B]

Pointer value to the newest log.

3.8.97. Optic 2 wire Threshold [Register 11C]

Optical Probe Upper Threshold.

3.8.98. Hysteresis [Register 11D]

Hysteresis Value.

3.8.99. Thermistor Threshold [Register 11E]

Thermistor Probe Upper Threshold.

3.8.100. Number of truck compartments [Register 120]

Number of compartments on truck.

3.8.101. Stop logging dome out events [Register 121]

Disable dome out event logging. 0: dome out logging, 1: No dome out logging.

3.8.102. TIM info logged [Register 122]

Logged TIM info.

3.8.103. TIM size [Register 123]

Size of TIM.

3.8.104. Super TIM code [Register 124]

Super TIM code.

3.8.105. Log Data State [Register 125]

State of data log.

3.8.106. Compare Volts [Register 126]

Pin 5 voltage from truck connection (Diagnostic line). Used to calculate number of connected probes.

3.9. SuperTIM Data List

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
01	Carrier Name	R	20	300 - 309	Name of the carrier company
02	Carrier Address	R	40	30A - 31D	Address of the carrier company
03	Contract Number	R	10	31E - 322	Carriers contract number
04	Operating Service	R	20	323 - 32C	Operating service’s name
05	Driver ID	R	10	32D - 331	ID of the driver
06 - 15	Allowable Volume Compartment 1 - 16	R	2	332 - 341	Allowable volume of compartment 1 - 16
16	Vapor Tight Certificate Type	R	1	342 MSB	Certificate type for the vapor tightness certificate. For vapor tightness type this value will always be 1.

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
17	Vapor Tight Certificate Date	R	3	342 LSB - 343	Expiration date for the vapor tightness certificate
18	Vapor Tight Certificate Number	R	20	344 - 34D	Vapor tightness certificate number
19	Safe Pass Certificate Type	R	1	34E MSB	Safe loading pass certificate type. For safe loading pass type this value will always be 2.
1A	Safe Pass Certificate Date	R	3	34E LSB - 34F	Safe loading pass certificate expiration date
1B	Safe Pass Certificate Number	R	20	350 - 359	Safe loading pass certificate number
1C	Certificate 3 Type	R	1	35A MSB	Certificate 3 type
1D	Certificate 3 Date	R	3	35A LSB - 35B	Certificate 3 expiration date
1E	Certificate 3 Number	R	20	35C - 365	Certificate 3 number
1F	Certificate 4 Type	R	1	366 MSB	Certificate 4 type
20	Certificate 4 Date	R	3	366 LSB - 367	Certificate 4 expiration date
21	Certificate 4 Number	R	20	368 - 371	Certificate 4 number
22	Certificate 5 Type	R	1	372 MSB	Certificate 5 type
23	Certificate 5 Date	R	3	372 LSB - 373	Certificate 5 expiration date
24	Certificate 5 Number	R	20	374 - 37D	Certificate 5 number
25	Table Valid	R	2	37E	A flag to indicate if the TIM data is valid
26	Table Revision	R	2	37F	The revision number of the TIM data
27	Alternate TIM ID Valid	R	1	380 LSB	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	6	381 - 383	Alternate TIM ID
29	Number of Compartments	R	1	420 LSB	Number of compartments on the truck
2A	Compartment Volume Units	R	1	384 LSB	Volume units for the compartments 1: Gallons, 2: Imperial Gallons, 3: Liters

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
2B	Trailer ID Number	R	20	385 - 38E	Trailer ID number
2C	Compartment Config	R	1	38F LSB	Current configuration of the compartments 0: Front to Rear, 1: Rear to Front
2D	Vapor Interlock Type	R	1	390 LSB	Trucks vapor interlock type 0: None, 1: In Ground Line, 2: As Last Sensor
2E - 3D	Compartment 1 - 16 Types Allowed	R	3	391 - 3B0	Product types allowed in compartment 1 - 16
3E	Max Loading Temperature	R	1	421 MSB	Maximum loading temperature
3F	Temperature Units	R	1	421 LSB	Temperature units 1: Celsius, 2: Fahrenheit
40	Compartment 1 Type Loaded	R / W	3	3B1 LSB - 3B2	Fuel type loaded into compartment 1
41	Compartment 1 Batch ID Loaded	R / W	3	3B3 LSB - 3B4	Fuel batch date code loaded into compartment 1
42	Compartment 1 Volume Loaded	R / W	2	3B5	Volume of fuel loaded into compartment 1
43	Compartment 2 Type Loaded	R / W	3	3B6 LSB - 3B7	Fuel type loaded into compartment 2
44	Compartment 2 Batch ID Loaded	R / W	3	3B8 LSB - 3B9	Fuel batch date code loaded into compartment 2
45	Compartment 2 Volume Loaded	R / W	2	3BA	Volume of fuel loaded into compartment 2
46	Compartment 3 Type Loaded	R / W	3	3BB LSB - 3BC	Fuel type loaded into compartment 3
47	Compartment 3 Batch ID Loaded	R / W	3	3BD LSB - 3BE	Fuel batch date code loaded into compartment 3
48	Compartment 3 Volume Loaded	R / W	2	3BF	Volume of fuel loaded into compartment 3
49	Compartment 4 Type Loaded	R / W	3	3C0 LSB - 3C1	Fuel type loaded into compartment 4

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
4A	Compartment 4 Batch ID Loaded	R / W	3	3C2 LSB - 3C3	Fuel batch date code loaded into compartment 4
4B	Compartment 4 Volume Loaded	R / W	2	3D4	Volume of fuel loaded into compartment 4
4C	Compartment 5 Type Loaded	R / W	3	3C5 LSB - 3C6	Fuel type loaded into compartment 5
4D	Compartment 5 Batch ID Loaded	R / W	3	3C7 LSB - 3C8	Fuel batch date code loaded into compartment 5
4E	Compartment 5 Volume Loaded	R / W	2	3C9	Volume of fuel loaded into compartment 5
4F	Compartment 6 Type Loaded	R / W	3	3CA LSB - 3CB	Fuel type loaded into compartment 6
50	Compartment 6 Batch ID Loaded	R / W	3	3CC LSB - 3CD	Fuel batch date code loaded into compartment 6
51	Compartment 6 Volume Loaded	R / W	2	3CE	Volume of fuel loaded into compartment 6
52	Compartment 7 Type Loaded	R / W	3	3CF LSB - 3D0	Fuel type loaded into compartment 7
53	Compartment 7 Batch ID Loaded	R / W	3	3D1 LSB - 3D2	Fuel batch date code loaded into compartment 7
54	Compartment 7 Volume Loaded	R / W	2	3D3	Volume of fuel loaded into compartment 7
55	Compartment 8 Type Loaded	R / W	3	3D4 LSB - 3D5	Fuel type loaded into compartment 8
56	Compartment 8 Batch ID Loaded	R / W	3	3D6 LSB - 3D7	Fuel batch date code loaded into compartment 8
57	Compartment 8 Volume Loaded	R / W	2	3E8	Volume of fuel loaded into compartment 8
58	Compartment 9 Type Loaded	R / W	3	3D9 LSB - 3DA	Fuel type loaded into compartment 9

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
59	Compartment 9 Batch ID Loaded	R / W	3	3DB LSB - 3DC	Fuel batch date code loaded into compartment 9
5A	Compartment 9 Volume Loaded	R / W	2	3DD	Volume of fuel loaded into compartment 9
5B	Compartment 10 Type Loaded	R / W	3	3DE LSB - 3DF	Fuel type loaded into compartment 10
5C	Compartment 10 Batch ID Loaded	R / W	3	3E0 LSB - 3E1	Fuel batch date code loaded into compartment 10
5D	Compartment 10 Volume Loaded	R / W	2	3E2	Volume of fuel loaded into compartment 10
5E	Compartment 11 Type Loaded	R / W	3	3E3 LSB - 3E4	Fuel type loaded into compartment 11
5F	Compartment 11 Batch ID Loaded	R / W	3	3E5 LSB - 3E6	Fuel batch date code loaded into compartment 11
60	Compartment 11 Volume Loaded	R / W	2	3E7	Volume of fuel loaded into compartment 11
61	Compartment 12 Type Loaded	R / W	3	3E8 LSB - 3E9	Fuel type loaded into compartment 12
62	Compartment 12 Batch ID Loaded	R / W	3	3EA LSB - 3EB	Fuel batch date code loaded into compartment 12
63	Compartment 12 Volume Loaded	R / W	2	3EC	Volume of fuel loaded into compartment 12
64	Compartment 13 Type Loaded	R / W	3	3ED LSB - 3EE	Fuel type loaded into compartment 13
65	Compartment 13 Batch ID Loaded	R / W	3	3EF LSB - 3F0	Fuel batch date code loaded into compartment 13
66	Compartment 13 Volume Loaded	R / W	2	3F1	Volume of fuel loaded into compartment 13
67	Compartment 14 Type Loaded	R / W	3	3F2 LSB - 3F3	Fuel type loaded into compartment 14

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
68	Compartment 14 Batch ID Loaded	R / W	3	3F4 LSB - 3F5	Fuel batch date code loaded into compartment 14
69	Compartment 14 Volume Loaded	R / W	2	3F6	Volume of fuel loaded into compartment 14
6A	Compartment 15 Type Loaded	R / W	3	3F7 LSB - 3F8	Fuel type loaded into compartment 15
6B	Compartment 15 Batch ID Loaded	R / W	3	3F9 LSB - 3FA	Fuel batch date code loaded into compartment 15
6C	Compartment 15 Volume Loaded	R / W	2	3FB	Volume of fuel loaded into compartment 15
6D	Compartment 16 Type Loaded	R / W	3	3FC LSB - 3FD	Fuel type loaded into compartment 16
6E	Compartment 16 Batch ID Loaded	R / W	3	3FE LSB - 3FF	Fuel batch date code loaded into compartment 16
6F	Compartment 16 Volume Loaded	R / W	2	400	Volume of fuel loaded into compartment 16
70	Terminal Name	R / W	20	401 - 40A	Name of the terminal for the last load
71	Terminal Address	R / W	40	40B - 41E	Address of the terminal for the last load
72	Terminal Gantry Number	R / W	1	41F	Gantry number for the last load
73 - 77	Fault Log	R	6		Fault entry
78	Service Center Name	R	20		Name of the last service center
79	Service Center Address	R	40		Address of the last service center
7A	Builder Name	R	20		Name of the company that built the truck
7B	Builder Address	R	40		Address of the company that built the truck
7C	Truck Serial Number	R	10		Trucks serial number
7D	Truck VIN	R	20		Trucks vehicle identification number
7E	Truck Build Date	R	3		Date the truck was built (Hex GMT Epoch Format)

Subfunction	Register Name	Read / Write	Data Length (Bytes)	Holding Registers	Description
7F	Truck Weight Units	R	1		Weight unit for the vehicle weight 1: Pounds, 2: Kilograms
80	Truck Gross Vehicle Weight	R	4		Weight of the vehicle
81	Intellicheck Type	R	1		Is an Intellicheck being used
82	Overfill Sensor Type	R	1		Are overfill sensors being used
83	Retained Sensor Type	R	1		Are retained sensors being used
84 - 93	Compartment 1 - 16 Build Volume	R	2		Compartment 1 - 16 tank volume
94	Scully Sensors	R	1		Are Scully sensors installed
95	Tank Model Number	R	20		Model number of the tank
96	Max Working Pressure	R	2		Maximum tank pressure
97	Allowable Working Pressure	R	2		Working tank pressure
98	Pressure Units	R	1		Pressure units 1: PSI, 2: Bar, 3: kPa
99	Bulkheads	R	1		Number of bulkheads on a truck
9A	Tank Profile	R	20		Profile description of truck tanks
9B - B2	Overfill Sensor 1 - 24 Length	R	1		Length of sensor 1 - 24

4. 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
10	Write Multiple 16-bit Registers
41	Write Single Vehicle
42	Read Single Vehicle
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 SuperTIM Data
54	Write SuperTIM Data
55	Read Third Party
56	Write Third Party
57	Read SuperTIM Data Area
58	Write SuperTIM Data 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

4.1. 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.

4.1.1. 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

4.1.2. 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.

4.1.3. 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

4.2. 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.

4.2.1. Example Query Message Reading Input Status Bits 0 - 15

Read Input Statis 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

4.2.2. 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 w ith VIP, in vehicle list, and authorized.

4.3. 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.

4.3.1. 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

4.3.2. Example Response Message Read Registers 3 and 4

Read Regisers 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.

4.4. 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.

4.4.1. 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

4.4.2. 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

4.5. 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.

4.5.1. Example Query Message Write Register 0E

Write Single Register Query Message

Addr	Func	Reg Numbr MSB	Reg Numbr 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.

4.5.2. Example Response Message Write Register 0

Write Single Register Normal Response Message

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

4.6. 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.

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

Write Regisers 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.

4.6.2. 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

4.7. 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**:

4.7.1. 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

4.7.2. 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

4.8. 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:**

4.8.1. 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

4.8.2. 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

4.8.3. 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

4.9. Write Password — Function Code 44 Hex

Writes a password to the unit for use in date-stamp VIP operation.

4.9.1. Example Query Message

Write Password "FRODO" Query Message

Addr	Func	Byte Count	First Char 'F'	'R'	'O'	'D'	Fifth Char 'O'	Error Check Field 16 Bit CRC
00-63	44	5	46	52	4F	44	4F	CRC

Byte Count is 9 characters maximum

4.9.2. Example Response Message

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

Write Password "FRODO" Normal Response Message

Addr	Func	Byte Count	First Char 'F'	'R'	'O'	'D'	Fifth Char 'O'	Error Check Field 16 Bit CRC
00-63	44	5	46	52	4F	44	4F	CRC

4.10. Write Company ID Name Function Code 45 Hex

Writes a company ID name to the unit for use in date-stamp VIP operation.

4.10.1. Example Query Message

Write Company ID Name "EXON" Query Message

Addr	Func	Byte Count	First Char 'E'	'X'	'O'	'N'	Error Check Field 16 Bit CRC
00-63	45	04	45	58	4F	4E	CRC

If the company ID name is less than 4 characters, it must be null padded to fill out four characters.

4.10.2. 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

4.11. 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.

4.11.1. 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

4.11.2. 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.

4.12. 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:

4.12.1. 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

4.12.2. 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

4.13. Backup Functions — Function Code 48 Hex

Function Code 48 (hex), interfaces with the backup processor (Intellitrol only). All backup functions are in hex in the examples below. The following table summarizes the backup processor functions. Broadcast messages will be ignored except for force bit (Modbus function code 05):

Note: The Intellitrol release 1.0 firmware does not implement or support Backup processor Modbus communications.

#	FUNCTION	DESCRIPTION
00	Read Backup Status	Reads the backup status word
01	Read Backup Revision	Reads the backup firmware version
02-03	Reserved	
04	Read Bypass Key Serial Number	Reads the serial number of the bypass key if the backup processor is in bypass
05	Read Jumper Settings	Reads the jumper settings for baud rate, parity, data bits, address, ground, debug, and mixed comm
06	Read Log Entry	Reads an entry from the backup processor's log
07-FF	Reserved	

4.13.1. Read Backup Status — Backup Function Code 00 Hex

4.13.1.1. Example Query Message

Read Backup Status Query Message

Addr	Func	Backup Func	Error Check Field 16 Bit CRC
00-63	48	00	CRC

4.13.1.2. Example Response Message

Read Backup Status Normal Response

Addr	Func	Backup Func.	Status Word MSB	Status Word LSB	Jumper No.	Clock Error	Error Check Field 16 Bit CRC
00-63	48	00	20	00	04	00	CRC

This example show s a baud rate jumper error

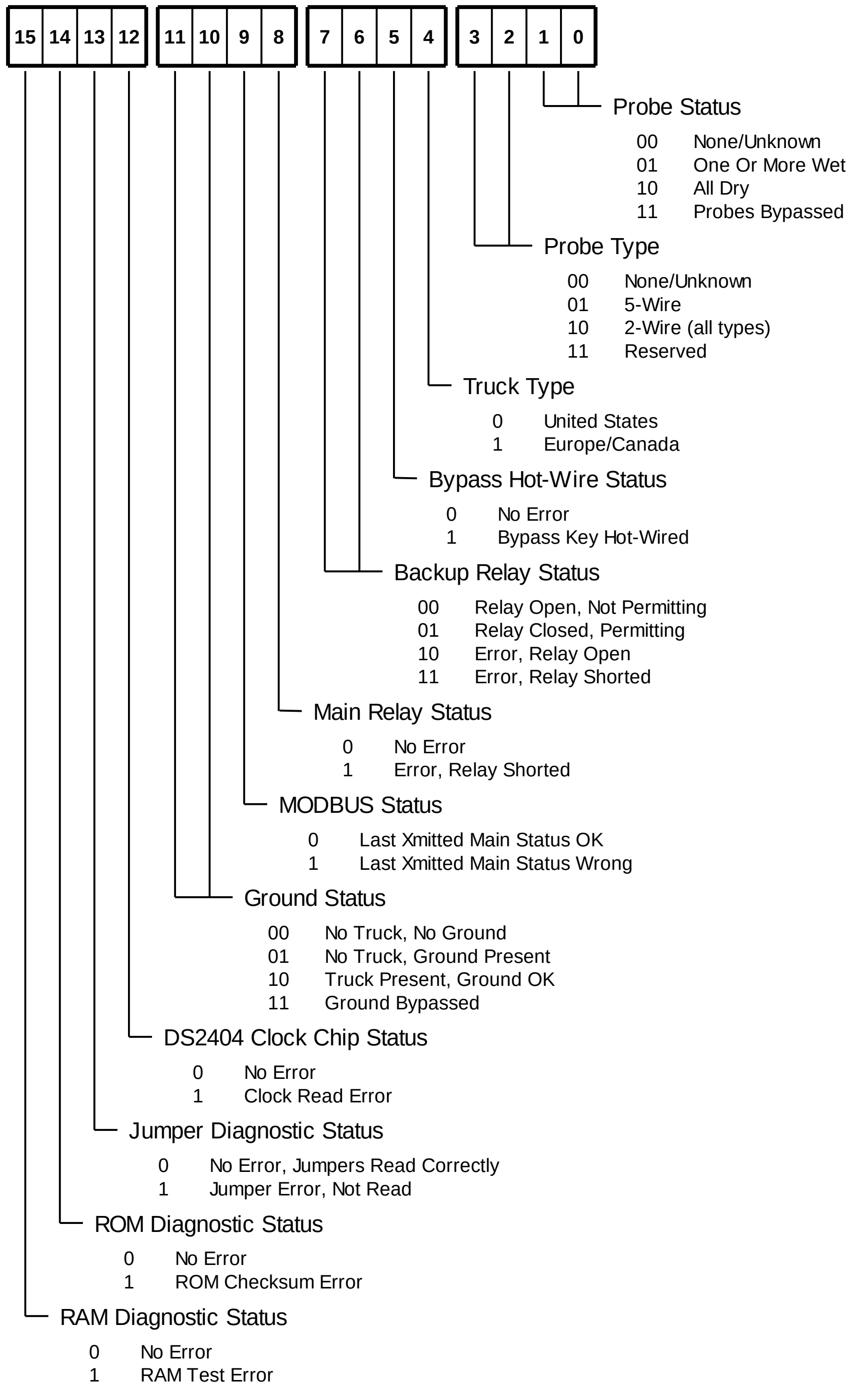
4.13.1.3. Jumper and Clock Errors

The following table lists the Jumper No. and Clock Error fields if there are any errors with either diagnostic:

#	JUMPER	CLOCK
00	No Error	No Error
01	Ground Enable	Chip Not Present
02	Mixed Communications	Checksum Error
03	Debug	Wrong Chip Present
04	Baud Rate	Noisy Clock Reading
05	Parity	Clock Not Set
06	Data Bits	Clock Not Running
07	Address 1	Reserved
08	Address 16	Reserved
09	6/8 Jumper	Reserved
0A-FF	Reserved	Reserved

4.13.1.4. Backup Status Word Bits

The following illustration lists the possible values and meanings for the real-time Backup Status word bits:



4.13.2. Read Backup Revision Backup Function Code 01 Hex

4.13.2.1. Example Query Message

Read Backup Revision Query Message

Addr	Func	Backup Func	Error Check Field 16 Bit CRC
00-63	48	01	CRC

4.13.2.2. Example Response Message

Read Backup Revision Normal Response

Addr	Func	Backup Func	Major Rev	Minor Rev	Date MSB	Date	Date	Date LSB	Error Check Field 16 Bit CRC
00-63	48	01	01	02	2F	43	6A	50	CRC

The Date field is the UNIX style time and date expressed as seconds since 1/1/70
This example show s the Backup Revison is Rev 1.02 at 2/16/95 15:08

4.13.3. Read Bypass Serial Number Backup Function Code 04 Hex

4.13.3.1. Example Query Message

Read Bypass Serial Number Query Message

Addr	Func	Backup Func	Error Check Field 16 Bit CRC
00-63	48	04	CRC

4.13.3.2. Example Response Message

Read Bypass Serial Number Normal Response

Addr	Func	Backup Func.	Serial No. MSB	Serial No.	Serial No.	Serial No.	Serial No.	Serial No. LSB	Error Check Field 16 Bit CRC
00-63	48	04	00	00	00	01	F2	E3	CRC

This example show s the backup processor is currently in overfill bypass by the key 000001F2E3H
NOTE: If the unit is not being bypassed, this field w ill read 000000000000H
If the unit is being bypassed by the TAS, this field w ill read FFFFFFFFFFFFH

4.13.4. Read Jumper Settings Backup Function Code 05 Hex

4.13.4.1. Example Query Message

Read Jumper Settings Query Message

Addr	Func	Backup Func	Error Check Field 16 Bit CRC
00-63	48	05	CRC

4.13.4.2. Example Response Message

Read Jumper Settings Normal Response

Addr	Func	Backup Func	Baud Rate	Parity	Data Bits	Addr	GND Enabl	Debug	Mixed Comm	Error Check Field 16 Bit CRC
00-63	48	05	04	00	08	01	00	00	01	CRC

This example shows 9600 baud, no parity, 8 bits, address 01H, no ground, not in debug, and mixed communication

4.13.4.3. Jumper Settings

#	BAUD	PARITY	DATA	GND ENA	DEBUG	MIXED
00	Default	None	—	Not Enabled	Not Enabled	Not Enabled
01	1200	Odd	—	Enabled	Enabled	Enabled
02	2400	Even	—	—	—	—
03	4800	—	—	—	—	—
04	9600	—	—	—	—	—
05	19.2k	—	—	—	—	—
06	—	—	—	—	—	—
07	—	—	7 Bits	—	—	—
08	—	—	8 Bits	—	—	—
09-FF	—	—	—	—	—	—

4.13.5. Read Log Entry Backup Function Code 06 Hex

The backup processor will not repeat error entries if they've already happened within the last 3 days.

4.13.5.1. Example Query Message

The backup log is a circular buffer stored on board the 512-byte EEPROM in 8-byte records for a maximum of 64 log entries.

Read Log Entry Query Message

Addr	Func	Backup Func	Log Entry	Error Check Field 16 Bit CRC
00-63	48	06	15	CRC

Read backup log entry 21

4.13.5.2. Example Response Message

Read Log Entry Normal Response

Addr	Func	Backup Func	Log Entry	Action	Date MSB	Date	Date	Date LSB	Status Word MSB	Status Word LSB	Error Check Field 16 Bit CRC
00-63	48	06	15	02	2D	1C	5C	78	20	00	CRC

Log entry 21 shows a self check error occured at 3:30 PM Christmas 1993 from the jumpers being in error
Upon call, the Status Input Data field contains the log entry to be read.
The Date field is a UNIX style time and date expressed as seconds since 1/1/70
The Status Word field uses the same format as the read Backup Status command

4.13.5.3. Example Response Message

The action field tells what action occurred to generate the log entry, and the possible entries are:

#	ACTION
00	Reset
01	Watchdog
02	Self-Check
03	Bypass Start
04	Bypass End
05-FF	Reserved

4.14. Read Event Log — Function Code 49 Hex

The Event Log is a generalization of the VIP Bypass Log; accordingly, this function is a generalization of, and works like the Read VIP Bypass Log (function code 43 hex), with the following differences:

- Nominally the Event Log is 32 entries in size, but the actual size may vary, and can be determined from the *Event Log Block Size* register.
- There is no *Latest Log Pointer* register. All records must be downloaded, and times compared to determine the latest.

The VIP rack controller does not support the Event Log.

4.14.1. Example Query Message

Read Log Element 12 Query Message

Addr	Func	Elmnt Numbr MSB	Elmnt Numbr LSB	Error Check Field 16 Bit CRC
00-63	49	00	12	CRC

4.14.2. Example Response Message—Reset Occurred (Type Code = 02)

Read Log Element 0012 Normal Response (Reset Occurred)

Addr	Func	Elmnt No. MSB	Elmnt No. LSB	Type	Sub-type	Rep Mask MSB	Rep Mask LSB	Start Time MSB	Start Time	Start Time	Start Time LSB	Hdw Rev MSB	Hdw l Rev LSB	Kernel Ver MSB
00-63	49	00	12	02	20	FF	FC	30	DE	19	B5	12	00	00

Kernel Ver LSB	Shell Ver MSB	Shell Ver LSB	Jum-pers MSB	Jum-pers LSB	Con-fig2 MSB	Con-fig2 LSB	Rsrvd	Rsrvd	. . .	. . .	Rsrvd	Rsrvd	Error Check Field 16 Bit CRC
00	05	E3	81	4C	01	44	00	00	. . .	. . .	00	00	CRC

Log Element 12 Shows that a reset first occured at 3:25:41 AM Christmas 1995

This example response shows:

- Type code 02 — System Reset;
- Subtype 20 (hex) — "Watchdog" reset;
- Time code 30DE19B5 — the [first] "Reset" event occurred at 3:25:41 25-Dec-95;

- RepMask FFFC (hex) — a total of 3 resets occurred;
- Info block for “Reset” shows:
    - the Hardware is Revision 1.2.0;
    - the firmware kernel is version 0.0.0 (The first Intellitrol release has no separate kernel);
    - the firmware shell is version 0.5.E3 (a late Intellitrol Beta test for the first production release);
    - the unit is hardware-jumpered for 8-compartment, Deadman-Switch, Ground-Fault Detection, and VIP operations; The DEBUG jumper is installed (or was at the time the reset occurred) — indicating an UNSAFE CONDITION in which to operate the Intellitrol rack controller unit since DEBUG defeats most safety and fault tests;
    - the unit is software-configured for Deadman-Switch and VIP operations; Ground-Fault Detection is disabled, either deliberately by TAS command, or not Features-Password-enabled; (the software has no control over the 6/8 compartment operation and so will run as jumpered, in this case 8-compartment mode)

4.15. 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.

4.15.1. 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

4.15.2. 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).

4.16. Write Bypass Keys Function Code 4B Hex

The Write Bypass Keys command 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).

Address	Function	Element Number		Number of Elements		Bypass Key 1					
01 - 63	4B	12	34	00	02	00	11	22	33	44	AA

Bypass Key 2						16 Bit CRC
00	11	22	33	44	AA	CRC

Write Bypass Keys Query Message

Address	Function	Element Number		Number of Elements		16 Bit CRC
01 - 63	4B	12	34	00	02	CRC

Write Bypass Keys Response Message

4.17. Read Bypass Keys Function Code 4C Hex

The Read Bypass Keys command 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).

Address	Function	Element Number		Number of Elements		16 Bit CRC
01 - 63	4C	12	34	00	02	CRC

Read Multiple Vehicles Query Message

Address	Function	Element Number	Number of Elements		Byte Count	Bypass Key 1					
01 - 63	4C	12	34	00	02	00	11	22	33	44	AA

Bypass Key 2						16 Bit CRC
00	11	22	33	44	AA	CRC

Read Multiple Vehicles Response Message

4.18. 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.

4.18.1. Example Message Set Enable Password

Set Enable-Features Password Command

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

4.18.2. 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

4.19. Read Special EEPROM Block Function Code 4E Hex

The Read Special EEPROM Block Modbus command is used to read one of the special configurations- or diagnostic-parameters blocks from the rack controller. This information is highly unit type-specific (and firmware version-specific as well) and is intended solely for Scully manufacturing and servicing purposes.

Address	Function	EEPROM Block Code	Byte Count	16 Bit CRC
01 - 63	4E	02	02	CRC

Read Special EEPROM Block Query Message

Address	Function	EEPROM Block Code	Byte Count	Data		16 Bit CRC
01 - 63	4E	02	02	C0	1A	CRC

Read Special EEPROM Block Response Message

4.20. Write Special EEPROM Block Function Code 4F Hex

The Write Special EEPROM Block Modbus command is used to write one of the special configurations- or diagnostic-parameters blocks to the rack controller. This information is highly unit type-specific (and firmware version-specific as well) and is intended solely for Scully manufacturing and servicing purposes.

Address	Function	EEPROM Block Code	Byte Count	Data		16 Bit CRC
01 - 63	4E	02	02	C0	1A	CRC

Write Special EEPROM Block Query Message

Address	Function	EEPROM Block Code	Byte Count	16 Bit CRC
01 - 63	4E	02	02	CRC

Write Special EEPROM Block Response Message

4.21. Report Compartment Volume — Function Code 50 Hex

This command will send back the volume of the compartment requested from the SuperTIM.

Address	Function	Compartment Number	16 Bit CRC
01 - 63	50	0C	CRC

Report Compartment Volume Query Message

Address	Function	Compartment Number	Compartment Volume				16 Bit CRC
01 - 63	50	0C	00	00	00	00	CRC

Report Compartment Volume Response Message

4.22. Read SuperTIM Area — Function Code 51 Hex

This command will read data from the SuperTIM.

4.23. Write SuperTIM Area — Function Code 52 Hex

This command will write data to the SuperTIM.

4.24. Read SuperTIM Data — Function Code 53 Hex

This command will read data from the SuperTIM. Refer to section 3.9 for subcommands.

Address	Function	SuperTIM Data Subcommand	16 Bit CRC
01 - 63	53	05	CRC

Read SuperTIM Data Query Message

Address	Function	SuperTIM Data Subcommand	SuperTIM Data	16 Bit CRC
01 - 63	53	05	……………………………	CRC

Read SuperTIM Data Response Message

4.25. Write SuperTIM Data — Function Code 54 Hex

This command will write data to the SuperTIM. Refer to section 3.9 for subcommands.

Address	Function	SuperTIM Data Subcommand	SuperTIM Data	16 Bit CRC
01 - 63	54	40	……………………………	CRC

Write SuperTIM Data Query Message

Address	Function	SuperTIM Data Subcommand	16 Bit CRC
01 - 63	54	40	CRC

Write SuperTIM Data Response Message

4.26. SuperTIM Third Party commands Function Code 55 - 56 Hex

These commands are available for Intellitrol Model 3 or greater. Several companies have requested the ability to put proprietary information into the SuperTIM. This way the information will follow the truck. One example could be an electronic copy of the last invoice. There is 5K bytes reserved for this. The Intellitrol will allow access to this through the Modbus. The Intellitrol software will not perform any action on the information stored here.

The Third-Party area Modbus Read command:

Address	Function	Memory Address		Data Length	16 Bit CRC
01 - 63	55	0C	00	01	CRC

Read Third Party Data Query Message

Address	Function	Memory Address		Data Length	Third Party Data	16 Bit CRC
01 - 63	55	0C	00	01	………………………………	CRC

Read Third Party Data Response Message

The data field must not exceed 70 bytes.
The Third-Party area Modbus Write command:

Address	Function	Memory Address		Data Length	Third Party Data	16 Bit CRC
01 - 63	56	0C	00	01	………………………………	CRC

Write Third Party Data Query Message

Address	Function	Memory Address		Data Length	16 Bit CRC
01 - 63	56	0C	00	01	CRC

Write Third Party Data Response Message

4.27. Read SuperTIM Data Area Function Code 57 Hex

This command reads data from the SuperTIM data area.

4.28. Write SuperTIM Data Area Function Code 58 Hex

This command writes data to the SuperTIM data area.

4.29. Insert Vehicle Function Code 59 Hex

This command is an updated version of the insert vehicle function. Also works with alternate TIM ID's.

Address	Function	Vehicle ID Number						16 Bit CRC
01 - 63	59	00	00	01	21	39	EB	CRC

Insert Vehicle Query Message

Address	Function	Index		Vehicle ID Number						16 Bit CRC
01 - 63	59	00	00	00	00	01	21	39	EB	CRC

Insert Vehicle Response Message

4.30. Remove Vehicle Function Code 5A Hex

This command is an updated version of the remove vehicle function. Also works with alternate TIM ID's.

Address	Function	Vehicle ID Number						16 Bit CRC
01 - 63	5A	00	00	01	21	39	EB	CRC

Remove Vehicle Query Message

Address	Function	Confirmation		16 Bit CRC
01 - 63	5A	FF	FF	CRC

Remove Vehicle Response Message

4.31. Read Number of Probes Function Code 5B Hex

This command reads the number of probes connected to the Intellitrol.

Address	Function	16 Bit CRC
01 - 63	5B	CRC

Read Number of Probes Query Message

Address	Function	Number of Probes		16 Bit CRC
01 - 63	5B	00	0C	CRC

Read Number of Probes Response Message

4.32. Alternate between ADC Tables Function Code 5C Hex

This command alternates between updated ADC table added in version 1.7.0 and the original pre 1.7.0 ADC table. The ADC table is used for determining the number of connected probes. The command will return 01 if new table has been selected and it will return 00 if the old table has been selected. This command allows users to fall back to the old ADC table if they encounter issues with the updated table.

Address	Function	16 Bit CRC
01 - 63	5C	CRC

Alternate between ADC Tables Query Message

Address	Function	Selected ADC Table	16 Bit CRC
01 - 63	5C	01	CRC

Alternate between ADC Tables Response Message

4.33. Get current ADC Table Function Code 5D Hex

This command will return the value of the ADC table currently selected. The command will return 01 if new table is currently selected and it will return 00 if the old table is currently selected.

Address	Function	16 Bit CRC
01 - 63	5D	CRC

Get current ADC Table Query Message

Address	Function	Selected ADC Table	16 Bit CRC
01 - 63	5D	01	CRC

Get current ADC Table Response Message

4.34. 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.

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.

5. 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 form 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.

5.1. 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 w as sent w ith an illegal data address.
An exception response message w as returned by setting bit 7 in the Function Code byte.
The exception response code returned w as an 02, ILLEGAL DATA ADDRESS.

5.2. 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 date 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	The slave attempted to read extended memory but detected a parity error in the memory. The master can retry the request, but service may be required on the slave device
09	TIM Command Error	Defective or missing TIM
0A	Not a Super TIM	The Dallas chip in the TIM is not a DS1996
0B	Valid Error	The Truck Builder Table Valid location did not contain 0x55AA
0C	Invalid Compartment Number	Number of compartments exceed the maximum number of 16
0D	SPI Loader Family Error	The SPI EEPROM memory ID in the program loader puck is not 0x13 or 0x14
0E	SPI Write Error	Error trying to write to the Program Loader Puck
0F	SPI Read Error	Error trying to read the Program Loader Puck
10	TIM Memory Size or Data Length Error	Invalid address or the data length is greater than 70 bytes
11	TIM Write to Scratch Pad Error	Error occur when writing to the scratch pad area in the DS1996
12	TIM Verify Scratch Pad Error	Error occur when verifying the data written to the scratch pad area in the DS1996
13	TIM Copy Scratch Pad Error	Error occur when transferring the data from the scratch pad to the memory in the DS1996

14	Valid Flag Not Valid for This Entry	The Super TIM entry is not valid
15	Reading Intellitrol Serial Number Error	Error trying to read the Dallas Serial number in the Super TIM
16	Error Allocating Memory	Error occurred when trying to allocate memory
17	I2C Bus Error	An error was detected when trying to access an I2C device
18	Read Real Time Clock Error	Error when trying to read the DS1371 real time clock
19	Read Only	Register value is read only and cannot be written to.
80	No Response	Did not receive a response

6. 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.

6.1. "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.

6.1.1. 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,
   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,
   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);
}
```

6.1.2. 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 114 pages]