

The Wrong Way to Link Serial I/O Motor Protection Relays (MPRs) to the DeltaV System

This document describes Wiring Schematic.

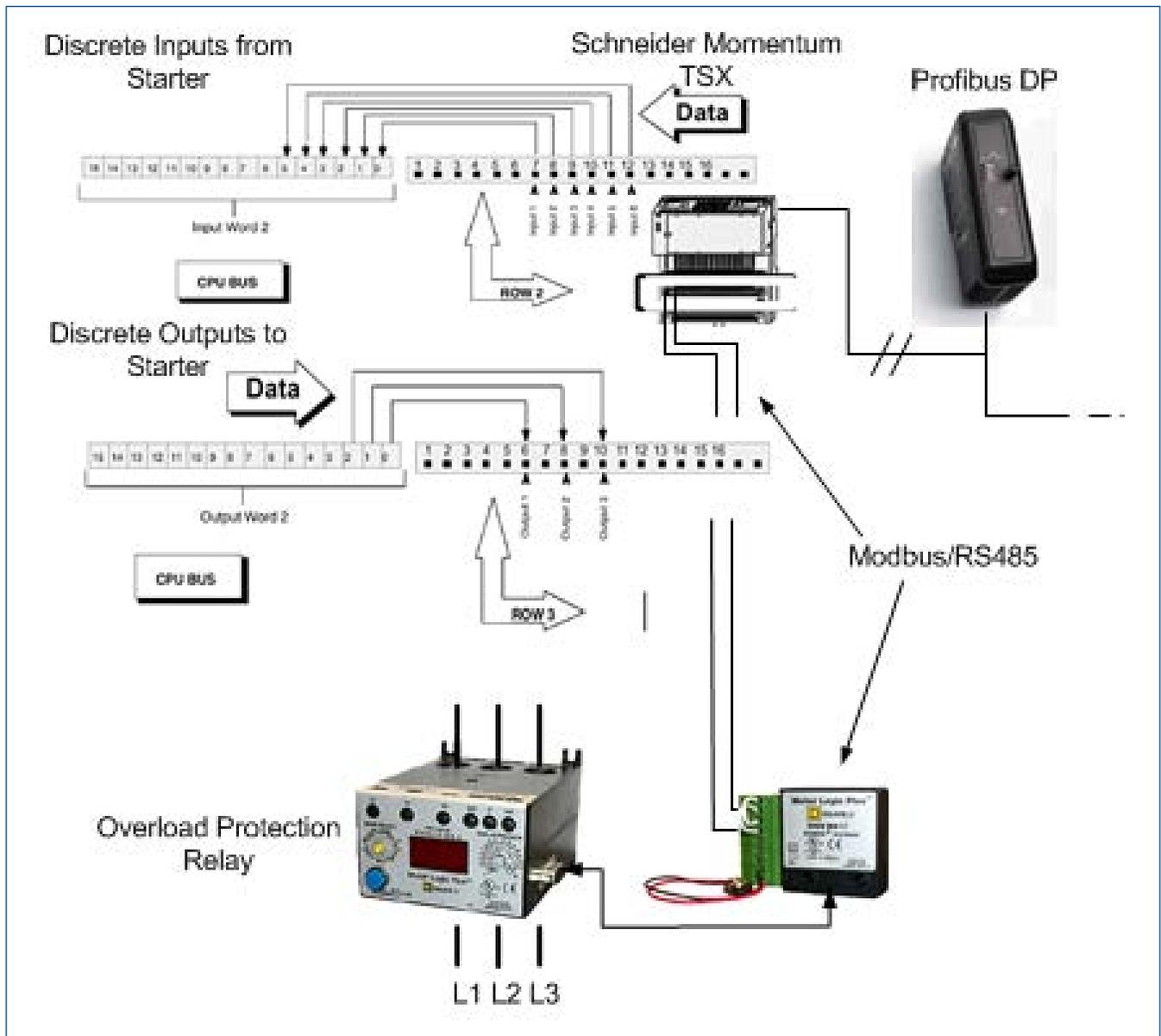


Table of Contents

Introduction	3
Schneider Momentum TSX/Modbus-Profibus Gateway 170 DNT 110 00 - Configuration and Working Basics	4
DST Consuming Serial Communication Management of a Momentum TSX	6
Tested Performance	10
Scaleup and Expected Performance	10
Integrated Profibus Solutions versus Gateways	11

Introduction

It is common practice to scale control systems according to the number of analog and discrete I/O. In the case of the DeltaV system, that approach implies the allocation of one DST license per I/O signal. This is pertinent, since system resources (controller CPU and memory) are tied to handling signals on the I/O bus.

When it comes to serial I/O or bussed technology, nothing changes from the perspective of a controller's resources. What changes are the field installation requirements, which are usually minimized. Thus, to a controller, the original vehicle a signal used to land on the I/O bus must not make a difference if the integrity and capacity of system resources are not to be corrupted.

In the case of making use of Schneider Momentum TSX/Profibus-Modbus gateway modules (170 ADM 110 00) it is theoretically possible to interconnect a number of 32 Modbus devices on one Profibus DP segment with DeltaV. Potentially large numbers of signals are configurable and the limit of 512 DSTs per controller might quickly interfere with a planned configuration for an anticipated number of controllers.

Extreme care must be taken when estimating the number of configured DSTs based on mere I/O counts. As soon as the Modbus parameterization functionality of a Schneider Momentum/Profibus-Modbus gateway is considered, the rule implying one DST per signal is no longer sustainable.

The serial port settings, the Modbus data register addresses, the data lengths and the Modbus query functions must be defined as references to 16 bit unsigned integer words stored in a Profibus DP data table. DeltaV interprets such a word as a signal that enters or exits the system (controller) on the I/O bus. Thus, it does not distinguish it from any conventional I/O signal. DeltaV 6.3 counts a word of 16 bit length as an analog DST and not a discrete DST.

A DCS/MCC interface connecting up to 20 Momentum modules per Profibus DP segment has been designed. Serial communication to each of the Momentum modules on the basis of Modbus is achieved with MPRs for M.V. and L.V. motors. Depending on the starter type, the number of configured analog DSTs per motor varies between 13 and 21. This is entirely due to the requirement to manage the Modbus communications through a Profibus gateway via DeltaV system I/O.

All efforts to circumvent system I/O managed by controllers for static Modbus parameterization will fail. In the DeltaV system, there are no direct links to the system I/O subsystem from workstations. System I/O must be referenced from control modules assigned to a controller for writing. One reference costs a DST license.

Performance testing has been successfully accomplished making use of three simulated motor types and three Momentum modules attached to MPRs mounted on test bench devices. Large-scale testing involved a number of 20 such devices on a Profibus segment. The supplier accomplished it during staging of the MCC. Initially, large time delays could be rectified, according to the MCC supplier, with active bus termination installed.

It could be that initial controller hardware was sized based on conventional I/O counts. If the decision were made to implement the above DCS/MCC architecture on the basis of Momentum gateways, controller hardware and DST licenses would have to be supplemented to avoid the violation of the 512 DST limit per controller. This, however, is far from being an optimal economic solution. The processing power of the number of controllers needed would greatly exceed the requirements to handle static Modbus parameterization and data formatting.

Integrated Profibus MPRs can be found on the market. The elimination of Modbus and the gateway to Profibus eliminates the need to assign system I/O to manage static Modbus parameters. This fact supports the rule of one DST per signal. This is not specific to Profibus but applies to all integrated bus-based systems in the automation market.

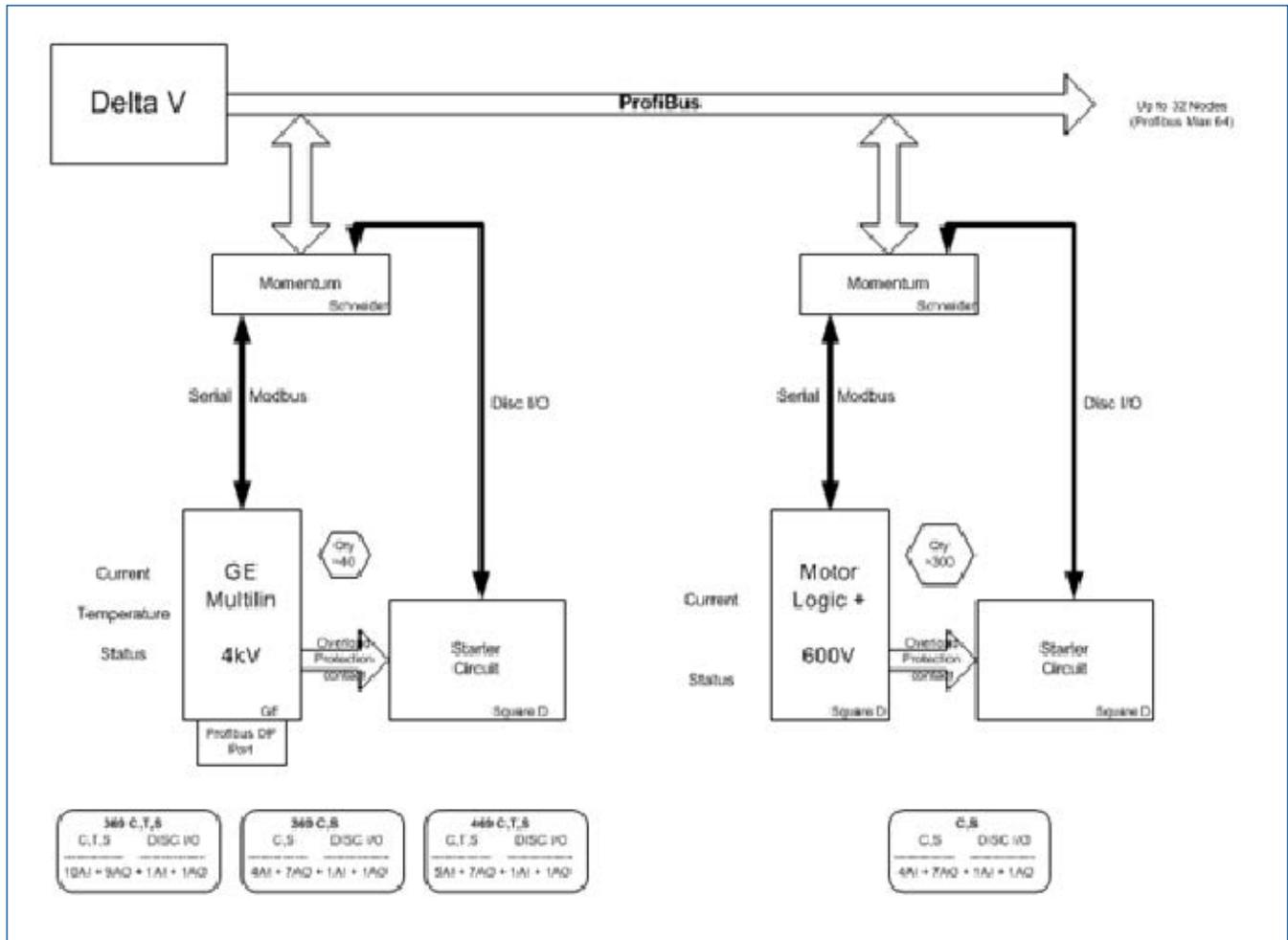


Figure 1 — MCC-DeltaV interconnection comprising M.V. and L.V. motors with C:= Current S:= Status T:= Temperature.

Schneider Momentum TSX/Modbus-Profibus Gateway 170 DNT 110 00 - Configuration and Working Basics

All configuration parameters and a Momentum TSX/Modbus-Profibus Gateway/DeltaV data transfer rely on Profibus DP communication. The GSD file that ships with Schneider I/O devices for Profibus defines 32 input and 32 output bytes available for communication with a Modbus-Profibus Gateway 170 DNT 110 00. These bytes can be seen to be arranged in a table with two columns comprising each 16 rows of 16 bit unsigned integers (words).

The high and low bytes in the output words define the bit pattern for a Momentum’s discrete outputs and store the configuration parameters for Modbus serial I/O with a connected motor protection relay or other device.

The high and low bytes in the input words store the bit pattern for a Momentum’s discrete inputs and its device status. Also, both the serial data and the status of a connected motor protection relay or other device are held.

A sequence number must be written to the first output word to drive the master-slave communications, according to Modbus. Every time the sequence number undergoes a change from its previous value, another Modbus query-response cycle is triggered.

The first input word, on the other hand, reads back the sequence number written a few instances ago. It ought to be logically verified that the sequence number and Modbus responses coincide with the sequence number and Modbus queries.

In a scenario of minimal complexity, the parameters for serial asynchronous transmission and Modbus queries are static such that the values of the words affected never change. The sequence number, however, must always be forced to change.

The discrete I/O base of a Momentum is independent of the sequence number change and the Modbus parametrization. It could be controlled via Profibus and shut down or started up with a zero or non-zero sequence number, respectively.

A MCC/DCS design concept could be based on the application of Momentum modules interfaced with the DeltaV system as indicated in the above figures. A Momentum discrete I/O base would be configured for motor command contacts and status feedback. Readings of M.V. and L.V. motor current, as well as winding/bearing temperatures from a motor protection relay together with fault alarms and prealarms, would exploit a Momentum Modbus interface.

The Profibus byte order and signal assignment within DeltaV Explorer must be verified to match the preset byte definition as explained above. With Big Endian byte order, it was found that the input sequence number high byte is the first byte (offset zero) and the low byte is the second byte (offset 1) and so forth up to the 32nd byte.

The impact on configured DST licenses, which are of the analog DST category for Profibus (DeltaV 6.3), can be halved, if the I/O mapping is accomplished according to 16 bit unsigned integers instead of individual bytes.

This is equally important when considering the configured DST limit of 512 maximum per controller (DeltaV 6.3 and earlier).

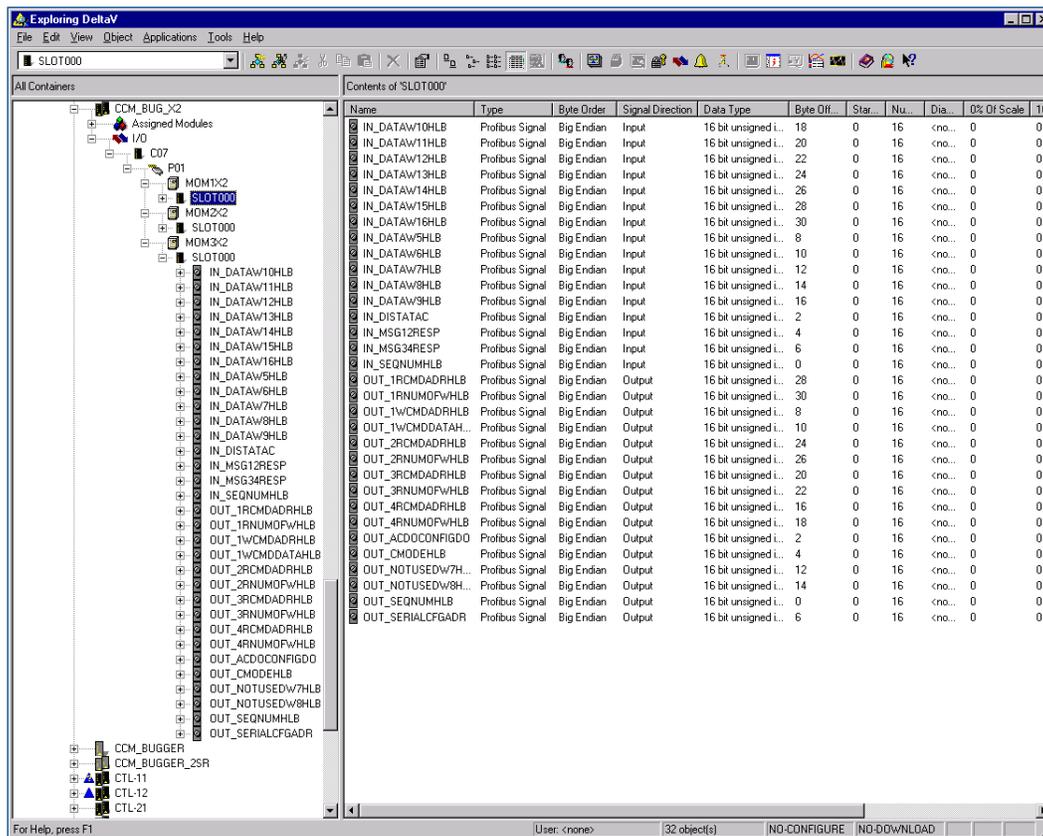


Figure 2 — Momentum Profibus I/O mapping within the DeltaV system.

DST Consuming Serial Communication Management of a Momentum TSX

As verified with DeltaV Explorer, each Profibus I/O reference related to a Schneider Momentum TSX is counted as one analog DST. The configuration of a Momentum discrete I/O base, including shutdown states and status feedback, is compact with Profibus and requires three DSTs with a Momentum 170 ADM 540 80 discrete I/O base. It is suitable for device control requiring a maximum of six DI and three DO per device.

In contrast to discrete I/O, the parameterization of serial communications (e.g. RS485) based on the Modbus protocol requires a sequence number driver, UART parameterization, Modbus function selection, data length definition and starting address setup. Good engineering practice also requires the handling of Modbus error messages. Each of the previously mentioned logical elements requires references to one or more Profibus words and counts as either a configured analog DST license for input or for output.

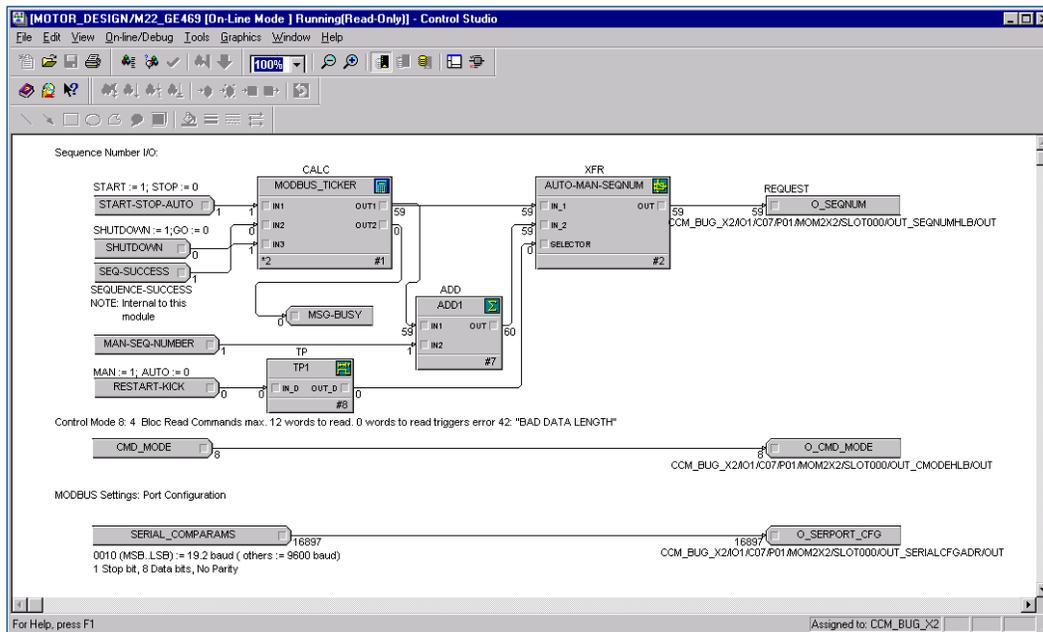


Figure 3 — Sequence number driver.

The above cutout from a DeltaV control logic module depicts three Profibus output parameters. This results in three configured AO DSTs. The first is the sequence number reference, the second specifies the Modbus command mode (a combination of up to four single or block read and writes), and the third represents the word defining the serial port configuration and Modbus device address.

Modbus data register addressing and data length definition also require a reference to a Profibus parameter each, such as depicted in Figure 4.

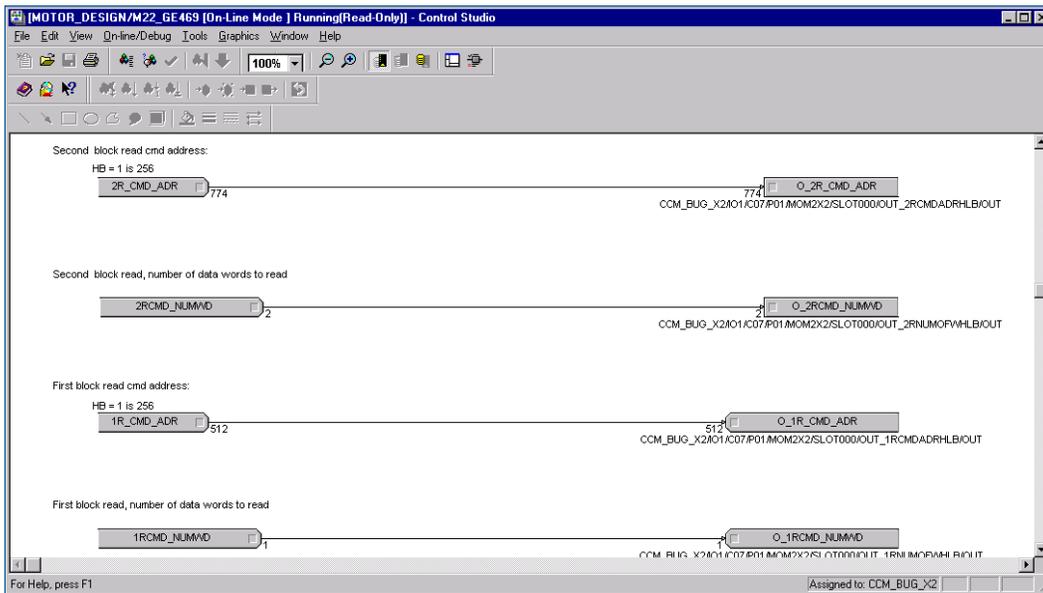


Figure 4 – Modbus register and data length specification.

The above cutout from the same DeltaV control logic module shown previously depicts four Profibus output parameters, starting addresses one and two, and data lengths for queries one and two. This accounts for four configured AO DSTs.

Up to four such Modbus “messages” could be defined. The constraints on the address values depend on the address space defined within the connected Modbus slave device. The constraint on the data length is given by a maximum of 12 words that could be read back via Modbus into the Profibus data table of the Momentum Profibus-Modbus gateway. If more than twelve parameters were to be read from a device, then sequence numbers would assume the role of segregating Modbus queries and responses written into the twelve words of the Profibus data table, as mentioned above.

Message input to the Profibus data table would have to be configured in DeltaV, as Figure 5 depicts below:

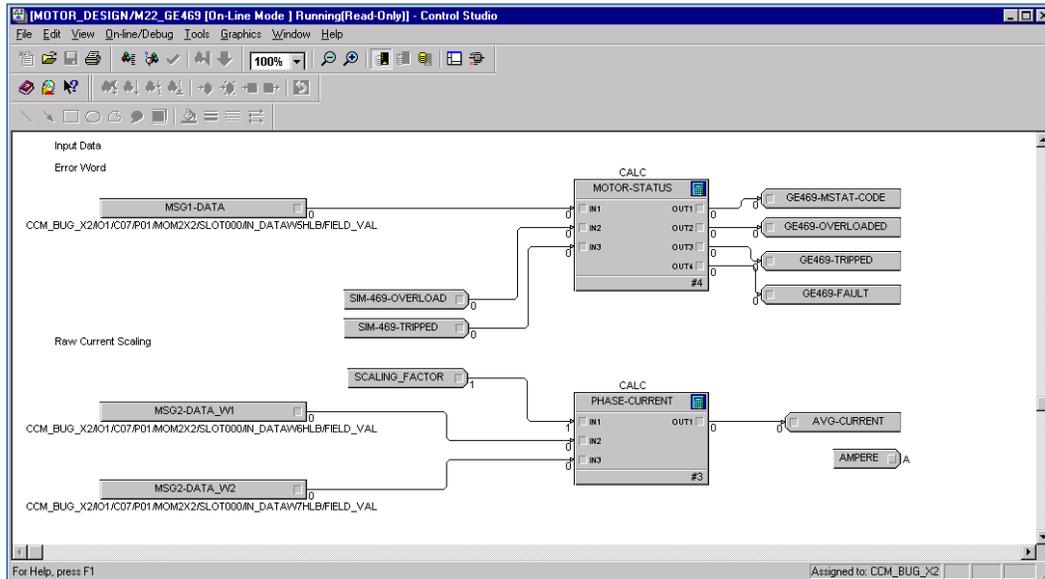


Figure 5 — Data written from a Modbus device (e.g. MPR) into the Profibus data table.

In the example above, the motor status and average phase current are written from an MPR to DeltaV across the Profibus-Modbus gateway. For the two device parameters shown, three AI DSTs are configured. This is due to the fact that the device at hand transfers motor current as two consecutive words. There is some small room for conservation of one DST. With the help of DeltaV Explorer, the current data for this particular device could be mapped according to a long, unsigned integer (32-bit) on the controller I/O level, leaving the library definition based on 16-bit unit unchanged. The latter is required to keep the I/O map generic to all combinations of Modbus messages.

Combined Modbus error message checking could be realized within DeltaV as shown below:

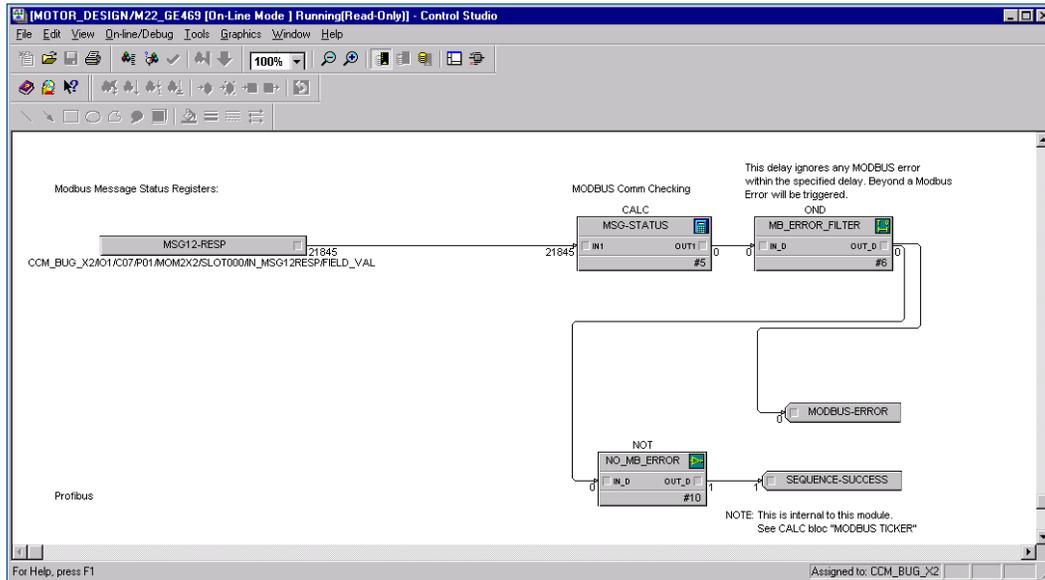


Figure 6 – Modbus message error checking and filter.

Each Modbus query of a pair writes a result code into one byte of one and the same Profibus data word. Combined into a 16 bit unsigned integer, the error status could be determined for both non-distinctively. This mechanism could be used as well if the total of four messages were defined. It would yield a 32-bit unsigned integer to be mapped on the controller I/O level for a specific device, in order not to corrupt the generic I/O map in the DeltaV Explorer library.

All the above figures were cut out of the same logic module opened with DeltaV Control Studio online. It is configured to interface with a GE469 Multilin MPR for a 4kV two-speed motor. There are 11 configured DSTs, 7AO and 4 AI, explained so far. This contrasts with the fact that only two device parameters are actually read: motor average phase current and MPR status.

The addition of one analog DST for Momentum-Profibus error checking and two analog DSTs for DI/DO yield an analog DST count of 14 altogether.

Tested Performance

On one Profibus segment, three Schneider Momentum TSX devices have been tested. Each was mounted with the 170 ADM 540 80 I/O base (including Modbus) and the Profibus Gateway 170 DNT 110 00. The DeltaV 6.3 system deployed for testing made use of an MD controller. One MPR per Momentum in a bench test device was connected to its serial port.

The length of the Profibus trunk cable was about 12 feet. The end-of-segment terminator was in the default IN position. The baud rate was set at 1.5 Mbit/s.

Separate DeltaV control modules were configured: one module for motor device control, one for discrete IO simulation, one for motor starter and control circuit simulation, and one for Modbus communication. A handful of custom alarms were defined and activated for the device control module and the communications module.

In order to minimize CPU loading, the execution rates of the three modules were set at 500 ms for the device control module, 200 ms for the discrete I/O control module, and at 1s for the communications module. CALC blocks for data conversion and alarm filtering were set to run at slower scan rates than the execution rates of the underlying control modules.

By visual inspection of the output diodes on the Momentum discrete I/O base the response time on the Profibus could be assessed. Between setting of a command using a motor faceplate within DV Operate and a state change of an LED was no noticeable difference.

Alarms were triggered with the help of phase current, phase voltage and RTD simulators integrated with the bench test devices. Related time delays between the occurrence of events and the display of alarms on the operator interface were within the expected magnitudes of the configured alarm filters.

The motor types simulated were two-speed full-voltage reversible and nonreversible. MPR parameters read were current and status and sporadically temperature. The minimum number of DSTs configured for all three motors was 39. The maximum number was 47. CPU loading was checked using DeltaV diagnostics. The MD free processor time was 92% of its capacity.

Investigations were initiated to find methods for writing static Modbus parameterization data once with the goal to save configured DSTs. The assignment of such static I/O to Application Stations or to the ProfessionalPLUS Station failed due to the isolation between controller assigned I/O and other nodes on the DeltaV control network. Writing static parameter data to the Profibus database from within DeltaV Operate running VBA scripts or using data links failed as well.

Strictly discrete I/O performance was tested attaching 20 Momentum modules to one and the same Profibus segment and each Momentum involved to an MPR of a staged MCC. Initially, long delays were observed. Improvements to shorten delays were successful, according to the MCC supplier. This implied the use of an active bus termination and a Schneider Quantum PLC.

Scaleup and Expected Performance

Depending on a project's constraints, there might not be a choice but to design Profibus segments connecting 20 Momentum devices and MPRs located in an MCC within a 100m distance from DeltaV cabinets. This is especially true when front-end engineering efforts fail to identify the issues discussed in this paper and contractual commitments are already made.

The discrete portion of I/O within less than one-second delay has been tested and reported feasible by the MCC supplier as described above.

The delays concerning data sent across Modbus past the Profibus gateway to DeltaV are expected to increase as the Profibus loading increases. The delays expected with 20 devices on the segment are supposed to match the delays observed when staging the MCC at the supplier's site. Profibus sends the entire data frame as configured regardless of whether the values of individual bytes changed or not. To these already observed delays, the DeltaV Operate update delays will have to be added.

The most severe impact on cost will be the requirement to acquire additional controller hardware to accommodate all configured DSTs within the 512 DST limit per controller and the total license fees for AI and AO.

Integrated Profibus Solutions versus Gateways

Without the need to manage gateways, integrated Profibus solutions would require analog DST licenses on a per (device) parameter basis. Motors—including current, status and temperature reading—as well as device control based on bit patterns within one and the same word, would consume five DSTs.

The above argument applies to a variety of suppliers and bus systems and is thus not restricted to Profibus. Gateways such as the one described do not lend themselves to new large-scale installation for the economic reasons mentioned in this paper. The situation could favor gateways regarding integrating devices featuring recent bus technologies and existing devices built on the basis of Modbus serial communications.

Emerson
North America, Latin America:
☎ +1 800 833 8314 or
☎ +1 512 832 3774

Asia Pacific:
☎ 65 6777 8211

Europe, Middle East:
☎ +41 41 768 6111

🌐 www.emerson.com/deltav

©2016, Emerson Process Management. All rights reserved.

The Emerson logo is a trademark and service mark of Emerson Electric Co. The DeltaV logo is a mark of one of the Emerson Process Management family of companies. All other marks are the property of their respective owners.

The contents of this publication are presented for informational purposes only, and while every effort has been made to ensure their accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. All sales are governed by our terms and conditions, which are available on request. We reserve the right to modify or improve the designs or specifications of our products at any time without notice.