

Contents

1.0	Introduction	2
2.0	Unidrive Installation	3
2.1	Connecting Modbus	4
2.2	Set Modbus plus address switches	6
2.3	Verify Modbus network is OK	9
3.0	Communicating with Unidrives	11
3.1	Read or Write to a drive parameter	11
3.2	Setting the Unidrive precision speed reference	13
3.3	Starting and stopping the Unidrive	16
3.4	Reading multiple parameters from the Unidrive	17
4.0	Modbus Fundamentals	18
5.0	Network performance	20
6.0	Frequently asked questions	20
	Index	24

1.0 INTRODUCTION

Modicon is one of the world's largest PLC manufacturers; in fact, a Modicon employee, Dick Morley, invented the PLC in 1968. Modicon is also the inventor of the Modbus Plus factory network, a 1.0 megabit token ring communications system that interfaces PLC's, drives, sensors, actuators and factory computer systems together. The Control Techniques **UD76 Modbus Plus Coprocessor Module** provides easy connection of Unidrive's to this popular industrial network.

The **UD76 Modbus Plus Coprocessor Module**, shown in Figure 1 below, is a 32 x 65 x 112 mm wedge-shaped plastic module that slides into the Unidrive's Large Application Module Slot.

Design of the **UD76 Modbus Plus Coprocessor Module** was accomplished under the auspices of the Modicon "ModConnect Partner's Program" of which Control Techniques Worldwide is a member. The module design includes some Modicon-licensed circuits and the operation of the module is certified by Modicon's testing labs in Andover, Massachusetts. This ensures that Control Techniques customers will have a Modbus Plus compatible drive that will function flawlessly with all other devices hooked up to the Modbus Plus network.

A token-ring network, such as Modbus Plus, prevents communications collisions by passing a special message called a "token" from node to node, eventually passing the token back to the first node. Only a node that has received a token message can transmit on the network - this prevents simultaneous transmission or collisions. This token passing feature also makes the network deterministic; any node can be guaranteed access to the network once per token rotation. Modbus Plus networks operate at 1.0 megabit with a maximum of 32 "nodes" permitted on a single token ring. The 32 node limit is simply a electrical limitation, repeaters can be used to extend the token ring to 64 nodes. If more than 64 nodes are required for a factory automation application, then special "bridges" may be employed to connect token rings together. In this case, Modbus Plus addressing protocol limits the network to four connected or bridged token rings. Modbus Plus uses a "Biphase-S" transmission technique requiring only a single twisted-pair shielded cable. Standard 9-pin IDC connectors make the daisy-chain wiring very simple and reliable.

The **UD76 Modbus Plus Coprocessor Module's** software is designed to look like a PC outfitted with a Modicon **SA85** network adapter. The drive appears as a set of holding registers, wherein each holding register is associated with a drive parameter. The operations supported are "read multiple registers", "write multiple registers" and "read single register". If you are controlling the drive from a Modicon 984 PLC, then the "**MSTR**" function can be used to read and write to one or several drive parameters.

Since the module employs a shared-memory interface, access to drive parameters is performed at memory bus speeds. This means that the **UD76 Modbus Plus Coprocessor Module** can update parameters inside the drive as fast as the Modbus Plus network can supply them. All in all, this is a convenient and cost effective way to connect Unidrives into factory automation systems that utilize Modicon PLCs.

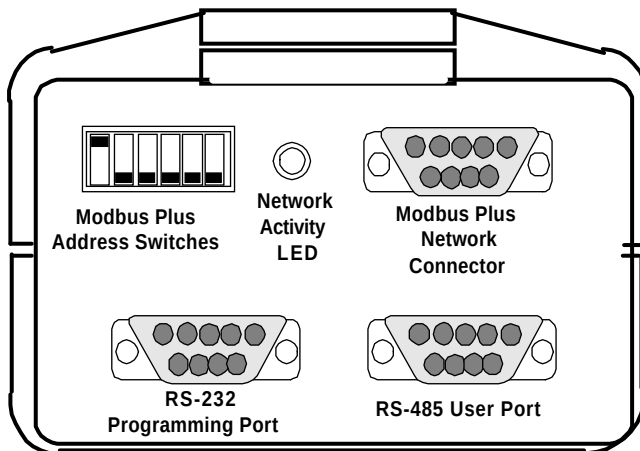


Figure 1. UD76 Modbus Plus Coprocessor Module

The **UD76 Modbus Plus Coprocessor Module** contains a mix of Modicon-licensed circuits and custom Control Techniques circuits. The Modicon section contains a 80C152 CPU with RAM and EPROM. A dual-port RAM interfaces the Modicon system to the Control Techniques section. The Modicon circuits drive a transformer-isolated RS-485 type twisted-pair network using a Biphase-S signaling. The Biphase-S signals are generated by a custom Modicon gate array circuit. The transceiver section employs extensive protective circuits to swamp out transients and bleed off static charges. The 6 position DIP-switch sets the Modbus Plus address which is read at power-up. A small green LED in the upper center of the Large Option Module indicates proper network activity. Under normal conditions it will blink 6 times a second if the network is exchanging tokens.

The **Control Techniques** circuits contain the **UD70** add-on controller module which includes a 16 MHz Intel 960SA 32-bit processor with 96 Kbytes of FLASH memory. The **UD70** can be programmed in a simple BASIC-like language called DPL (Drive Programming Language). The **UD70** includes a resident real-time operating system supporting a user-defined initialization task, background task, clock task, event task and encoder task. A robust debugger provides breakpoints, single stepping and watch windows. Modbus Plus support is embedded within the **UD70** operating system. This means that responses to network parameter queries are handled automatically.

2.0 UNIDRIVE INSTALLATION

If you are retrofitting a Unidrive to install the **UD76 Modbus Plus Coprocessor Module**, you must remove the terminal cover to allow you to insert the **UD76 Large Option Module**. Just follow the ensuing steps:

1. Turn off power to the Unidrive and wait 5 minutes.
2. Remove the terminal cover.
3. Position the **UD76 Large Option Module** as shown in Figure 2 below, slide it into place until it latches into position.
4. Insert the Modbus Plus 9-pin Cable Connector into the module's network connector and tighten the DB-9 lockdown screws.
5. Set the Address DIP-switches to the desired Modbus Plus Address (see Section XX.X following).
6. Apply power the Unidrive and observe that after about ten seconds, the green "network activity" LED will begin blinking at a rate of six blinks per second. This indicates that the module has joined the Modbus Plus network and is operating properly.
7. Refit the plastic terminal cover.

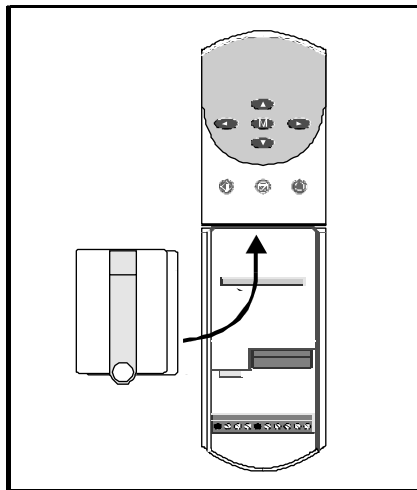


Figure 2. How to install the UD76 / Modbus Plus Large Option Module

This completes the installation of the **UD76 Modbus Plus Coprocessor Module** in Unidrive. Be sure you have set the DIP-switches and verified that you have the "network activity" LED blinking at six times a second before reattaching the terminal cover.

2.1 CONNECT MODBUS PLUS CABLE

Modbus Plus networks use **BELDEN 9841** twisted pair, shielded, with insulating outer jacket cable. The cable is daisy-chained to all **Modbus Plus** 9-pin connectors in a bus topology. The two nodes at the far ends of the daisy-chain require “terminating” connectors while nodes in the middle of the daisy-chain require “inline” connectors.

It's wise to buy these supplies from **Modicon**. **Modicon** sells convenient **IDC** connectors and a tool to assemble them. The “terminating” connectors also include resistive termination as required by the network transmission line design. The phone numbers to order these cables and connectors are:

MODICON CUSTOMER SERVICE ORDER ENTRY

North America	(800) 468 - 5342
International	+1- 508 - 794 - 0800

The parts needed to assemble **Modbus Plus** cable segments are:

Modicon Part #	Description
97 - 9841 - 100	MBPlus 100 Foot Reel
97 - 9841 - 500	MBPlus 500 Foot Reel
97 - 9841 - 01K	MBPlus 1000 Foot Reel
AS - MBKT - 085	Inline Connector Dark Gray (1 connector per pkg)
AS - MBKT - 185	Terminating Connector Light Gray (2 connectors per pkg)
AS - MBPL - 001	Modbus Plus Connector Installation Tool
GM - MBPL - 001	Modicon Modbus Plus Network Planning and Installation Manual (Chapter 5: Installing the Network Cable)

The two signal wires are “non-phasic” which means that you can connect them either way in the **Modicon Modbus Plus** connector with no ill effects.

Remember, if the drive is at the end of the cable daisy-chain, it must use a “terminating” connector. In all other cases, it must use an “inline” connector. The total length of the cable daisy-chain can be 1500 feet (450 meters) before having to resort to repeaters.

Modbus +

If you find yourself in the field installing one of these systems and you run out of **Modicon** connectors, you can still build the cables out of standard **DB-9** connectors. The signal wires are carried on pins 2 and 3 while the shield is connected to pin 1 as Figure 3 illustrates.

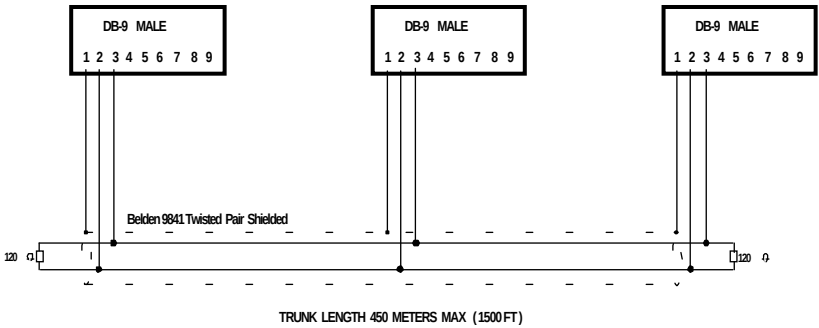


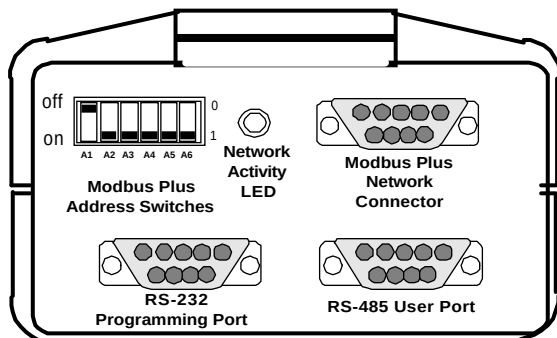
Figure 3. Modbus Plus Cable Details

Thanks to the “Biphase-S” signalling method, the connection of the two signal lines (pins 2 and 3) are non-phasic. In other words, you can connect them either way and it will still work. The illustration in Figure 6 is designed to show clearly how the pins are wired, but the **Modbus Plus** cable is a “daisy-chain” and stubs are not allowed.

It's important that the shields be connected to pin 1 as shown. Each **UD76 Modbus Plus Coprocessor Module** has protective circuits that use the shield or screen. The ends of the cable should be terminated with 120 ohms. You can see how convenient the **Modicon** parts are, you can buy “terminating” connectors that have this resistor built-in.

2.2 SET MODBUS PLUS ADDRESS SWITCHES

Each node must have a unique address from 1 to 64. It is recommended that you not set the module to address 1 since many installers like to give this address to the PLC. Likewise, it is recommended that you reserve address 64 for future maintenance. The following table shows the possible addresses and their respective switch settings.



ADD-RESS	A1	A2	A3	A4	A5	A6
1	1	1	1	1	1	1
2	0	1	1	1	1	1
3	1	0	1	1	1	1
4	0	0	1	1	1	1
5	1	1	0	1	1	1
6	0	1	0	1	1	1
7	1	0	0	1	1	1
8	0	0	0	1	1	1
9	1	1	1	0	1	1
10	0	1	1	0	1	1
11	1	0	1	0	1	1
12	0	0	1	0	1	1
13	1	1	0	0	1	1
14	0	1	0	0	1	1
15	1	0	0	0	1	1
16	0	0	0	0	1	1
17	1	1	1	1	0	1
18	0	1	1	1	0	1
19	1	0	1	1	0	1
20	0	0	1	1	0	1
21	1	1	0	1	0	1
22	0	1	0	1	0	1
23	1	0	0	1	0	1
24	0	0	0	1	0	1
25	1	1	1	0	0	1
26	0	1	1	0	0	1
27	1	0	1	0	0	1
28	0	0	1	0	0	1
29	1	1	0	0	0	1
30	0	1	0	0	0	1
31	1	0	0	0	0	1
32	0	0	0	0	0	1

Modbus +

ADD- RESS	A1	A2	A3	A4	A5	A6
33	1	1	1	1	1	0
34	0	1	1	1	1	0
35	1	0	1	1	1	0
36	0	0	1	1	1	0
37	1	1	0	1	1	0
38	0	1	0	1	1	0
39	1	0	0	1	1	0
40	0	0	0	1	1	0
41	1	1	1	0	1	0
42	0	1	1	0	1	0
43	1	0	1	0	1	0
44	0	0	1	0	1	0
45	1	1	0	0	1	0
46	0	1	0	0	1	0
47	1	0	0	0	1	0
48	0	0	0	0	1	0
49	1	1	1	1	0	0
50	0	1	1	1	0	0
51	1	0	1	1	0	0
52	0	0	1	1	0	0
53	1	1	0	1	0	0
54	0	1	0	1	0	0
55	1	0	0	1	0	0
56	0	0	0	1	0	0
57	1	1	1	0	0	0
58	0	1	1	0	0	0
59	1	0	1	0	0	0
60	0	0	1	0	0	0
61	1	1	0	0	0	0
62	0	1	0	0	0	0
63	1	0	0	0	0	0
64	0	0	0	0	0	0

Obviously, you should record the pod's **Modbus Plus** address since you will need it in your Ladder Logic programming.

2.3 VERIFY MODBUS PLUS NETWORK O.K.

The Network Activity LED flashes in distinctive patterns to indicate network status. The following table summarizes these status indications:

Indicator Pattern	Meaning
6 Flashes per Second	Normal Operation - token is being passed
1 Flash per Second	Node is powering up or reconfiguring (should last 5 seconds)
2 Flashes, OFF for 2 Seconds	Never Getting Token
3 Flashes, OFF for 1.7 Seconds	Sole Station, no other nodes responding
4 Flashes, OFF for 1.4 Seconds	Duplicate Station detected

If you cannot get Normal Operation, check all connectors for proper assembly and seating, look for open circuits and check for defective or missing terminating connectors. Of course, all nodes on the network must have unique ADDRESSES so double check the address switch settings.

3.0 COMMUNICATING WITH UNIDRIVES

In this manual, communication between Modicon PLCs and Control Techniques Unidrives is emphasized. It is possible to control a drive over the Modbus Plus network from an IBM-compatible PC using the Modicon **SA-85** Modbus Plus Adapter Card but this requires knowledge of Modicon NETBIOS software and is out of the scope of this manual.

The Unidrive appears to the PLC ladder logic programmer as a block of holding registers. Each holding register (register 40000 through 4xxxx) is mapped to a particular drive parameter by a very simple formula. The holding register number is derived by prefixing a 4 to the Unidrive menu and parameter number as shown below:

Unidrive Parameters are Registers!

Unidrive Register = 4MMPP

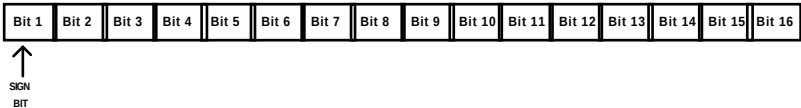
where: MM = menu number (0 .. 99)

PP = parameter number (00 .. 99)

For example, the Unidrive's Maximum Speed Clamp is Menu 01, Parameter 06. To the Modicon ladder logic programmer, this would be addressed as holding register 40106.

Modicon PLCs that support Modbus Plus networking have a MSTR function that allow the user to copy registers from the PLC to registers in the drive and vice versa. There can be many drives with similar sets of holding registers. The MSTR function requires a specific Modbus Plus address as well as a register specification, thus the MSTR operation talks to only one drive at a time.

Many drive parameters are bipolar. Older Modicon PLCs cannot express negative quantities. In fact, smaller Modicon PLCs can accept positive numbers no greater than 999 while the larger, 32-bit PLCs can accept positive numbers no greater than 9999. Knowing this, we can safely use the most significant bit (BIT 1) to be a sign bit in a "sign-magnitude" representation. In this motif, a logic ONE in BIT 1 denotes a negative quantity, a logic ZERO denotes a positive quantity.



For example, 300 in decimal is 0000000100101100 in binary. Using our sign-magnitude notation, -300 in decimal is 1000000100101100 in binary. The truth is, this is fairly simple to do in Modicon ladder logic. A **MBIT** (modify bit) function can be used to set or clear the sign bit (BIT 1).

RULE: SIGNED QUANTITIES

Set BIT 1 (left most significant bit) in any register to express a value as a negative quantity.

3.1 READ / WRITE TO A DRIVE PARAMETER

All drive parameters are accessed by a similar technique using ladder logic. As an example, the Unidrive speed reference may be set by writing into the first Preset Speed which is drive register 40121. Of course, you must select preset speed reference with parameter 01.14 and then specify Preset Speed #1 with parameter 01.15 ahead of time for this to take effect.

Modicon 984 PLCs that support Modbus Plus include a MSTR function to initiate network transactions from ladder logic. Essentially you can transfer registers from the PLC to the drive and vice versa. Keep in mind that a MSTR function may take several PLC scans to complete.

Details on using the MSTR function are delineated in several Modicon manuals; Chapter 17 of the **Modicon 984 Programmable Controller Systems Manual** gives a very complete description. As described in this manual, the MSTR function requires a few “control block” registers to be set up in advance. The following Figure 4 shows the the MSTR control block registers required to write one **PLC** register to drive #5, register 40121 (Preset Speed #1). In this example, we choose to locate these 9 MSTR control block registers at location 41000 through 41008 within the PLC. One quirk about addressing a drive is that the last non-zero routing byte must be a “one”, this is the task number (it's always a one).

MSTR CONTROL BLOCK REGISTERS

REGISTER	MSTR FUNCTION	REGISTER CONTENTS
41000	Operation Type	1 = write to drive 2 = read from drive
41001	Error status	See table that follows on page 14
41002	Length	1 = number of registers to be transferred
41003	Drive Registers	121 = start of drive register area (40121)
41004	Routing Byte 1	5 = drive address (single loop example)
41005	Routing Byte 2	1 = task number (single loop example)
41006	Routing Byte 3	0 (single loop example)
41007	Routing Byte 4	0 (single loop example)
41008	Routing Byte 5	0 (single loop example)

Figure 4 MSTR Control Block

The MSTR block has three internal nodes that must also be set up to do the network transmission. The top node specified within the MSTR block refers to the 9 register data block that sets up the transmission; this is at location 41000 in our example. The middle node specified within the MSTR block is the beginning of the PLC data area that will contain the register or registers to be written to the drive. In this example, we allocate PLC register 41009 (located conveniently right after the MSTR block registers) to hold the Preset Speed #1 value. The constant entered into the bottom node of the MSTR block is the maximum number of registers that will be read or written, in this case it will be a 1.

Our example ladder logic to write the Preset Speed #1 (drive register 40121) in drive #5 is shown in Figure 5 below. We used five SUB blocks to load the control block registers with values. The speed value, held in the PLC at register 41009, is set up by using *Modsoft Lite's* convenient "Reference Data" area where we can poke in a new value at anytime.

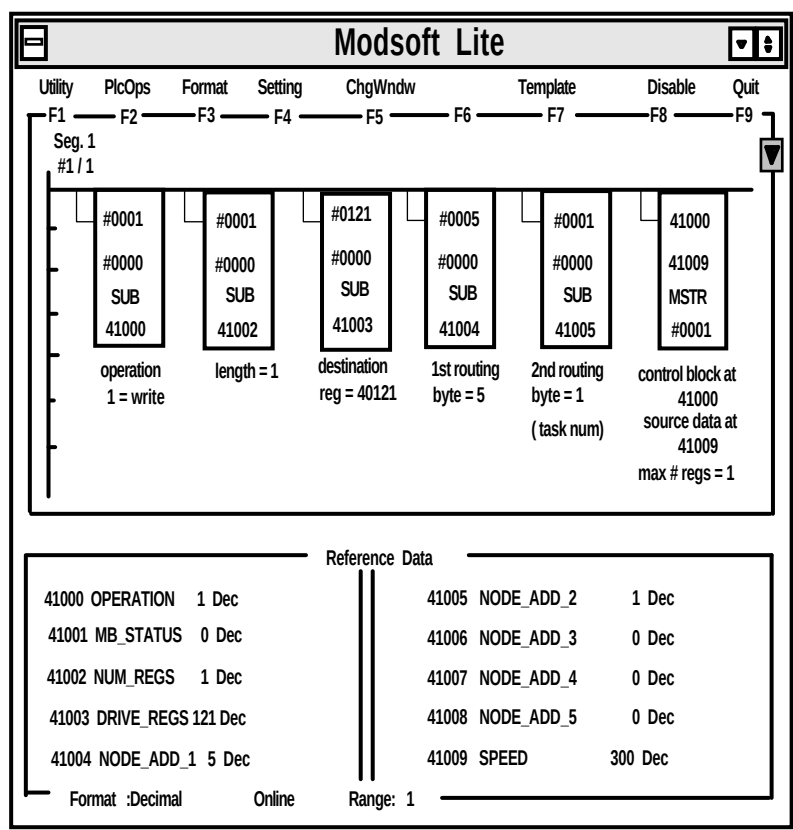


Figure 5. Ladder Logic To Set Drive Register 40121 To 300

If the PLC is set into **RUN** mode, this example ladder logic program will start scanning continuously. The Modbus Plus transaction will run repetitively and by inspecting the **PRESET SPEED #1 REFERENCE** parameter on the Unidrive's keypad and display, we can observe any updates occurring when we poke into register 41009 within the Reference Data area.

The following table summarizes some of the error codes returned in register 41001 by the **MSTR** function when the **Modbus Plus** transaction fails:

Hex Error Code	Meaning
1001	user abort
2001	invalid operation
2003	invalid length
2004	invalid offset
2006	invalid slave data area
2008	invalid network routing
2009	route = your own address
3001	unsupported function
3002	no such slave data area
4001	bad slave response
5001	bad network response
6Mss	routing failure
	ss = see Modbus+ manual
	M = loop number

The examples in this section show writing a register from the PLC to the drive. To read a drive register (or a block of registers), just change the operation code to 2 (read). The drive register will be read and deposited into the PLC register specified.

The rest of this manual is devoted to describing how to control the Unidrive's speed, start and stop the Unidrive and read from multiple parameters. Users can employ these examples as a springboard for their ladder logic programming.

3.2 SETTING THE UNIDRIVE PRECISION SPEED REFERENCE

The Unidrive's speed may be set digitally by writing to Parameter 01.21, Preset Speed #1 (mapped to register 40121) as described in the previous section.

However, it would be more interesting to utilize the Unidrive's "high resolution" feature to control the speed. This high resolution reference is selected by writing a "5" to Parameter 01.14 (register 40114) which is the Reference Selector. This selection can be made with the Unidrive's onboard keypad.

contain 1253 while register 40119 will contain 78 as shown in Figure 6.

The high resolution frequency reference is supplied to the drive in two parts. Parameter 01.18 (register 40118) holds the most significant part in times-10 format (XXX.X). Parameter 01.19 (register 40119) holds the least significant part (.0XX). The fact that they are contiguous allows us to transmit two successive registers using a single MSTR block.

For example, to set the drive speed to 125.378 Hz, register 40118 will

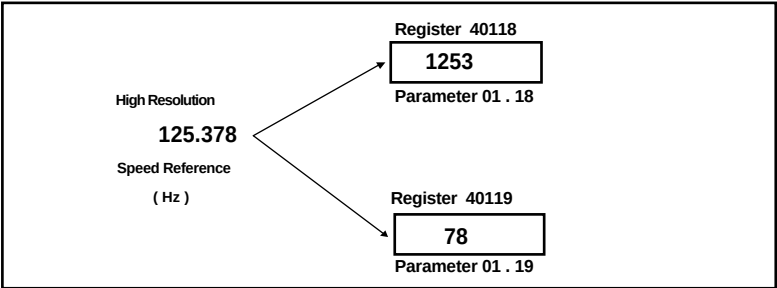


Figure 6. Decomposition of Precision Frequency Reference

Since these two high-resolution frequency control registers are adjacent, updating the speed via Modbus Plus can be done in one MSTR operation. The ladder logic to write to the high resolution frequency control parameters is shown in Figure 7.

Note that five SUBTRACT blocks are wired to set up the MSTR function's setup registers in the following configuration:

Register	Description	Value
41000	Operation Type	1 = write to drive
41001	Error Status Return	
41002	# of Registers to transfer	2 = send two registers
41003	Drive Register Area	118 = register 40118 (HI-RES FREQ)
41004	Routing Byte #1	5 = drive Modbus Plus Address
41005	Routing Byte #2	1 = drive task number (always 1)

A MSTR block is wired to do the network transmission. The top node specified within the MSTR block refers to the 9 register data block that sets up the transmission; this was previously set up by the SUB blocks. The middle node specified within the MSTR block is the beginning of the PLC data area that will contain the registers to be written to the drive. The constant entered into the bottom node of the MSTR block is the maximum number of registers that will be read or written.

In this example, we allocate PLC registers 41009 and 41010 (located conveniently right after the MSTR block registers) to hold the two hi-res frequency values. Here we have used SUB blocks to write the value 1253 into register 40048 and the value 78 into register 40049. This effectively sets the drive frequency to 125.378 Hz.

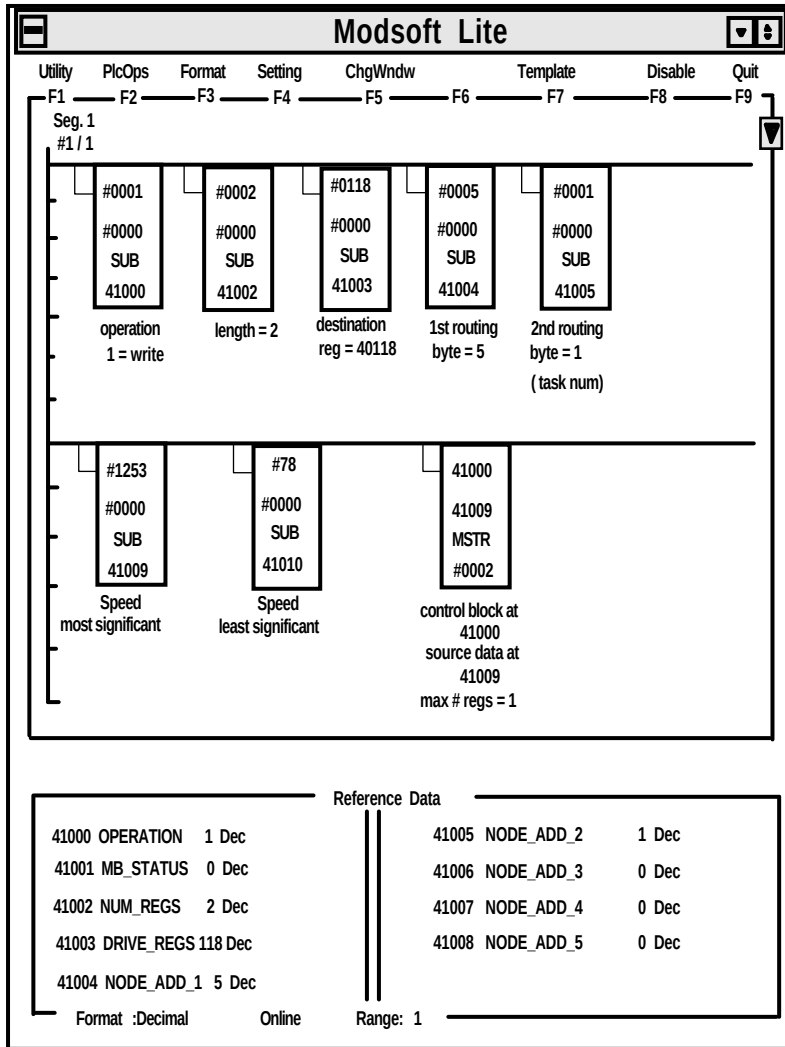


Figure 7. Ladder Logic To Set Unidrive High Resolution Frequency Reference

If the PLC is set into RUN mode, this example ladder logic program will start scanning continuously. The Modbus Plus transaction will run repetitively and by inspecting the high resolution reference parameters on the pod's keypad and display, we can observe any updates occurring when we poke into register 41009 within the Reference Data area.

3.3 STARTING AND STOPPING THE UNIDRIVE

If the Unidrive's "Sequencing Mode" (parameter 06.04) is set to 3 (PLC mode), then the drive can be started and stopped by controlling "Sequencing Bit 0" (parameter 06.30). Setting this bit will start the drive, conversely clearing this bit will stop it. The only proviso is that Unidrives, by default, direct terminal 27 (Run Forward pin) to parameter 06.30, so this connection has to be disabled. You can disable the input pin by adjusting parameter 08.19 (F4 Input Destination) to an unused or invalid parameter. Sample ladder logic to start or stop a Unidrive is given in Figure 8 below.

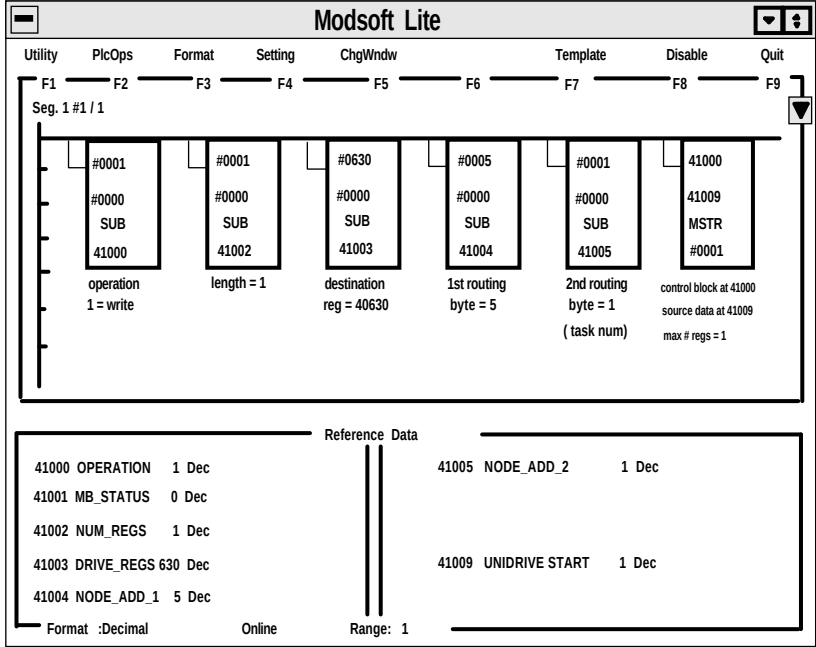


Figure 8. Starting and Stopping the Unidrive

Writing a one into register 41009 in the Modsoft Lite Reference Data Area will cause the Unidrive to start. Writing a zero will cause the Unidrive to stop. As can be seen from the ladder logic, register 41009 in the PLC will be transferred by the MSTR operation to register 40630 in the drive.

3.4 READING MULTIPLE PARAMETERS FROM THE UNIDRIVE

As an example of reading several parameters from the drive in a single Modbus Plus transaction, consider reading the entire Trip Log back from the drive. The last ten trips reside in drive parameters 10.20 through 10.29 (registers 41020 through 41029). Register 41020 is the most recent fault while register 4129 is the oldest fault. Check the Unidrive documentation for a complete list of trip codes.

The ladder logic to read the 10 trip log entries is shown in Figure 9 below.

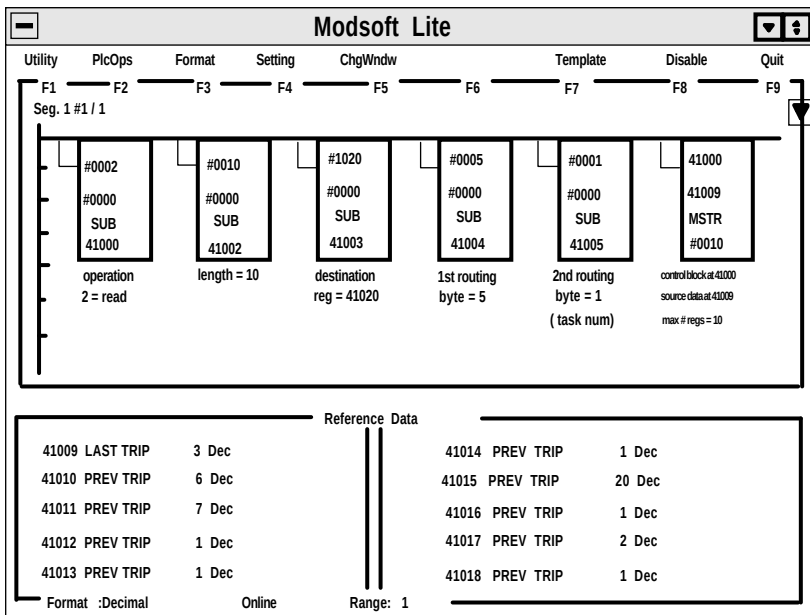


Figure 9. Reading the Unidrive's Trip Log

4.0 MODBUS FUNDAMENTALS

At the physical layer, the Modbus Plus network is a HDLC ring design permitting 32 nodes using standard isolated RS-485 signaling circuits operating in a Biphasic-S format. The speed is 1.0 Mbits/sec using shielded twisted-pair cabling. If more than 32 nodes are needed, repeaters are available from Modicon. In any event, the addressing methodology limits the number of nodes to 64. Larger networks can be created by bridging multiple 64 node rings via Modicon BP85 Bridge Plus adapters. The addressing system supports "routing" that allows a node to access a node in another ring. A Cyclic Redundancy Check (CRC) polynomial is used to verify correct reception of the HDLC frame. An illustration of a typical Modbus Plus system follows in Figure 10 below. As you can see, the cabling is a bus configuration but the system looks like a ring logically.

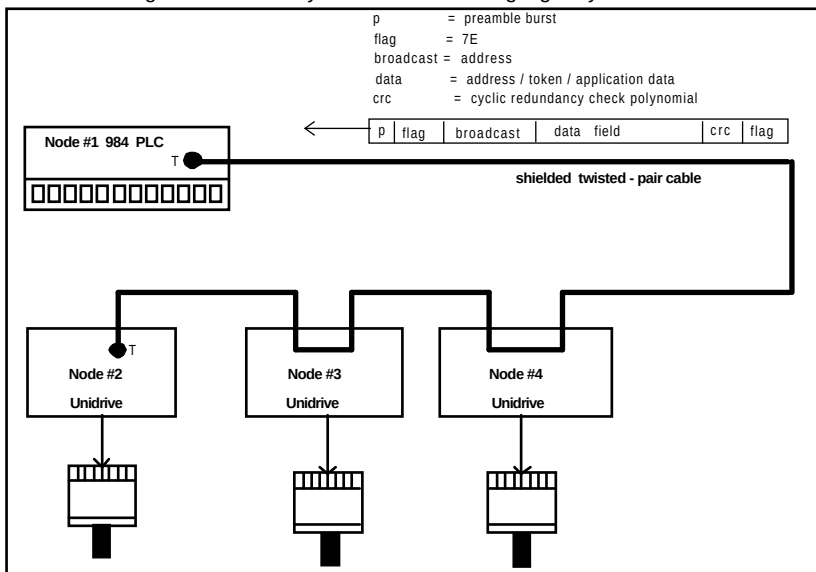


Figure 10. MODBUS PLUS NETWORK

The Modbus Plus HDLC message frame is fairly complex. One message pattern is the token frame used to pass "permission to transmit" from node to node. This message may contain up to 64 bytes of GLOBAL data. Of more interest to us is the more typical data frame wherein PLC register data is passed from node to node. In this case, the essential parts of the message frame are the five routing bytes and the MODBUS data frame.

To enable the Modbus Plus system to traverse multiple logical rings, five routing bytes are used to specify the path from the source to the destination. Multiple Modbus Plus rings must be connected with BP85 Bridges. The Figure 11 below shows how to specify the routing bytes to get from leftmost node 3 (PLC) to rightmost node 18 (Unidrive).

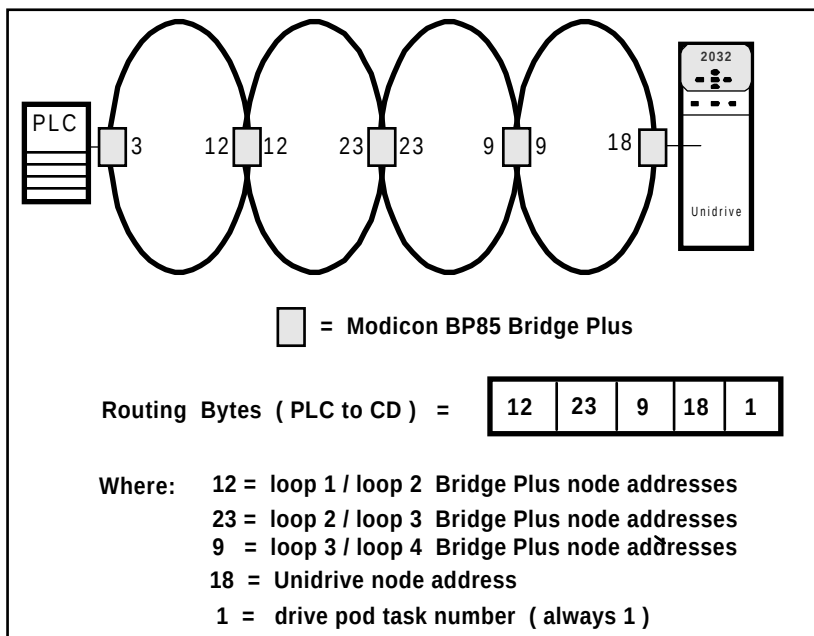


Figure 11. Multiple Ring Addressing

Node addresses on either side of the **BP85 Bridge Plus** don't have to be the same, but you will find it convenient to arrange them this way.

Since the drive is viewed as a collection of registers, only three **Modbus** commands need be supported - "read multiple registers", "write multiple registers" and "read single register". With these three commands, either a single register or a set of contiguous registers can be transferred.

For users just attempting to communicate to a **ControlTechniques** drive from a **Modicon 984 PLC**, the previous information is merely academic. The **MSTR** function automatically creates the "read multiple registers" and "write multiple registers" commands. Users attempting to use a **Modbus Plus SA-85** or a **DEC Q-Bus network adapter** must program their **NET BIOS** software to use only those two commands.

Future software revisions of the **UD76 Modbus Plus Coprocessor Module** will support the **Modicon GLOBAL DATA** feature. With this feature, any drive can put up to 32 registers into the token message. When this happens, all drives and **PLC's** on the network can read this data. Likewise, a **PLC** can broadcast 32 registers to all drives in the network. This can be very useful in assembling coordinated drive applications or in rapid status scanning. Please understand that the first generation of this **UD76 Modbus Plus Coprocessor Module** does not support this feature.

5.0 NETWORK PERFORMANCE

A Master's Thesis could be written attempting to analyze in detail the performance of a token ring network like Modbus Plus. If you're really interested, the **Modicon Modbus Plus Network Planning and Installation Guide (GM-MBPL-001 Rev. C)** has a detailed explanation of network performance.

The Modbus Plus network is capable of 20,000 registers per second throughput. Chapter 3 of the aforementioned manual gives the following insights.

A **Modbus Plus** network that is transmitting no data at all is still exchanging tokens. One complete traversal of the token around the ring is called the "Token Rotation Time". Each node takes 530 microsecs to transmit the token.

$TR = 0.53 (N)$	TR	$=$	Token Rotation Time (msec)
	N	$=$	# of nodes in the ring

Therefore, a 64 node Modbus Plus ring will require $(.53) (64) = 33.92$ msec to just pass the token around.

A Modbus Plus network in which a single node transmits 100 registers would have the following token rotation time:

$$TR = (2.08 + .016 (R)) + 0.53 (N)$$

TR	$=$	Token Rotation Time (msec)
R	$=$	# of registers to transmit
N	$=$	# of nodes in the ring

In the above example, a single node sending 100 registers would extend the token rotation time in a 64 node ring to 37.6 msec. The alert reader will notice two things. First, a 16-bit register transmits at 1 microsec per bit, just as you would expect for a 1 mbit network. Each register transmits in .016 msec or 16 microsecs. One hundred registers transmit in about 1.6 msec. Second, the "overhead" required to transmit these 100 registers is 2.08 msec. If you just send one register, the overhead is a very significant portion of the message. If you send 100, then the overhead is somewhat less significant. Obviously, it is more efficient to send multiple registers.

A Modbus Plus network in which 64 nodes transmit 100 registers each would have the following token rotation time:

$$TR = (2.08 + .016 (R))(N) + 0.53 (N)$$

TR	$=$	Token Rotation Time (msec)
R	$=$	# of registers to transmit
N	$=$	# of nodes in the ring

In this worst case, the token rotation time would be extended to 401.92 msec. Of course, having every node transmit 100 registers is a very unusual situation.

The message overhead seems, at first glance, to be fairly high in the Modbus Plus network. Remember that the typical message frame has command codes, source and destination addresses, routing bytes, CRC characters, broadcast addresses and the user's data. All transmissions require that the node wait for an acknowledgement from the receiving node. The overhead is high but you have a network that is fool-proof and operates peer-to-peer.

6.0 FREQUENTLY ASKED QUESTIONS

6.1 I can't get the module to work !

The DIP switch on the UD76 module is used to set the Modbus Plus address. The Modbus Plus address setting must be unique and not conflict with any other Modbus Plus address on the token ring.

If the network activity LED is blinking at six times a second, then the network is successfully exchanging tokens. The good technician always tries something simple first. Why not use the Modsoft Lite or Taylor PLC programming environment to try a simple example like those given earlier in this manual. If a simple example like this works, then the card is working properly.

6.2 The network isn't working !

If you have the network activity light blinking six times per second, the network is working and exchanging tokens.

If an error pattern is blinking, then the table given in Section 2.3 should give you a hint as to what's wrong. Each node must have a unique address so check the address switches carefully. The network is wired in a daisy-chain connection with terminating connectors on the far ends and inline connectors between. Check all connectors for proper assembly.

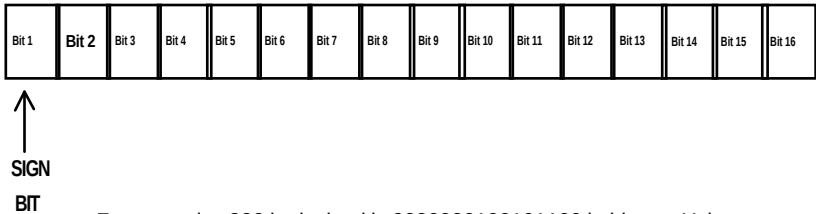
6.3 I can't change parameters !

Many drive parameters are read-only. These are denoted in the Appendices with the code RO.

Quite frankly, the Modbus Plus programmer has carte-blanche access to all parameters, regardless of the security setting. If a parameter is protected by a security setting, you may not be able to see it change on the display, but it will change nonetheless.

6.4 How do you send a negative number ?

Many drive parameters are bipolar. Modicon PLCs cannot express negative quantities. In fact, smaller Modicon PLCs can accept positive numbers no greater than 999 while the larger, 32-bit PLCs can accept positive numbers no greater than 9999. Knowing this, we can safely use the most significant bit (BIT 1) to be a sign bit in a “sign-magnitude” representation. In this motif, a logic ONE in BIT 1 denotes a negative quantity, a logic ZERO denotes a positive quantity.



For example, 300 in decimal is 0000000100101100 in binary. Using our sign-magnitude notation, -300 in decimal is 1000000100101100 in binary. The truth is, this is fairly simple to do in **Modicon ladder logic**. A **MBIT** (modify bit) function can be used to set or clear the sign bit (BIT 1) in Preset Speed #1 as shown in the sample ladder logic contained in Figure 12 below.

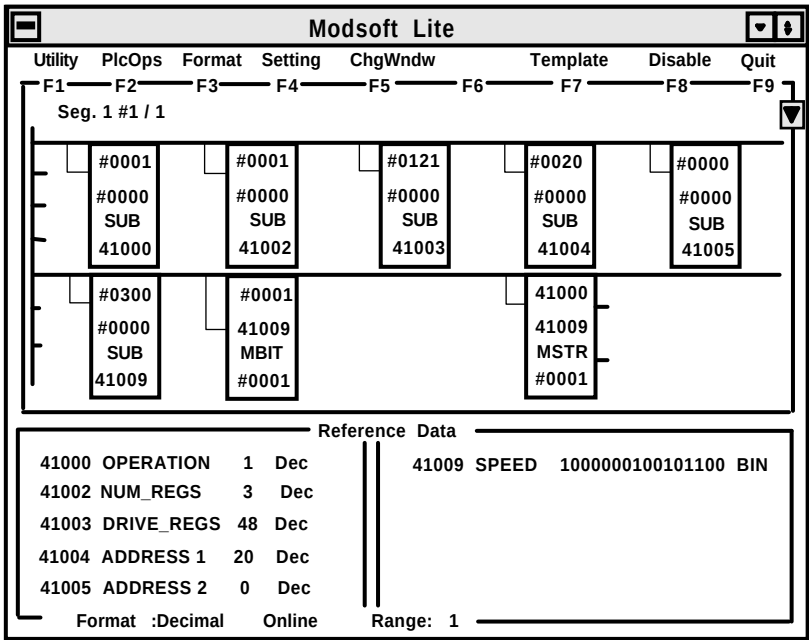


Figure 12. Ladder Logic to Set Preset Speed #1 to -300

6.5 The Modbus Plus module will not talk to my PLC?

The most common reason we've seen for the module refusing to communicate with the PLC is the proper specification of the routing bytes. The module looks like a SA-85 PC Modbus Plus adapter card. In this case, **many people forget that the last non-zero routing byte must be a one** (that's the task number).

For example, to communicate to a Unidrive drive at address 24 the PLC must specify the following routing bytes in the MSTR control registers:

REGISTER	MSTR FUNCTION	REGISTER CONTENTS
4XXXX + 4	Routing Byte 1	24 = drive address
4XXXX + 5	Routing Byte 2	1 = task number
4XXXX + 6	Routing Byte 3	0
4XXXX + 7	Routing Byte 4	0
4XXXX + 8	Routing Byte 5	0

6.6 Can I perform Modbus Plus operations from DPL?

Not in this release of the product. An upcoming release of the **UD76 Modbus Plus Coprocessor Module** will permit DPL programmers to read and write registers from other Modbus Plus nodes.

6.7 Your ladder logic examples don't work with my PLC ?

You may have a Modicon COMPACT PLC or an early-model 16-bit Modicon PLC that cannot accept entries into its functional blocks greater than 999. These PLCs have sixteen-bit registers and are capable of holding data up to 65535, it's just that the ladder logic editor will not permit specification of data larger than 999. Some ladder logic editors, such as the Taylor program, do permit entry of larger numeric values. Another way to get around this limitation is to specify a holding register rather than a constant in the SUB block and then use the "reference data area" to set the value of this register. This is not a problem in the 32-bit Modicon PLCs and the new Quantum-series PLCs.

6.8 I can't read Menu 0, Parameter 0 on the Unidrive ?

All right, you've got us on this one. Modicon registers are "one-based" so there isn't a register 40000 (the first register is 40001). Therefore, parameter 00.00 is inaccessible using the **UD76 Modbus Plus Coprocessor Module**. This is the only inaccessible parameter on the Unidrive. This restriction is inconsequential since parameter 00 on all menus is reserved for housekeeping duties such as "parameter save", "reset" and so on. You can access parameter 01.00 (register 40001) and perform the same operations.

Index

A

Address switches 3
Address table 7
Address table part 2 8

B

Biphase-S 6

C

Cable types 5
Communicating with Unidrive 10
Connections 5
Connectors 5

E

Error codes 13

F

FAQ's 21

I

Indicator pattern 9
Installation 4
Introduction 2

M

Modbus fundamentals 18
Modicon Help line 5
Modicon Manual 11
MSTR 11

N

Negative numbers 10

Network performance 20
Nodes 2

P

Parameters 10

R

Read & write to a parameter 11
Reading multiple parameters 17
Registers 10

S

Setting prec speed reference 13
Starting the Unidrive 16
Stopping the Unidrive 16

U

UD70 3
Unidrive comms 10
Updating a parameter 11

V

Verification of network activity 9

W

Writing to many parameters 17