ACS 500™ AC Drive connection to SLC 500 PLC™

ABB Drives has variable frequency drives from 1 Hp up to 400 Hp. This application note demonstrates the ProSoft Technologies 1150-MBM Modbus Master product to connect Allen-Bradley SLC 500 PLCs using serial communication to ACS 500 standard AC drives.

Application Description

To address the need to connect ABB standard AC drives to SLC 500 PLCs, ABB Drives and ProSoft Technologies, Inc. have introduced a serial Modbus solution for the 1746 platform.

In this note we will concentrate on a very common application: interfacing the A-B SLC 500 PLC to 4 ACS 500 drives using a RS-485 serial interface.

In most of the serial interface connections to the ACS 500 drive, the application wants to control the drive speed, and to start and stop the drive remotely. Most of the data being received back from the drive is contained in the operational data area, which includes information like frequency actual, current actual, and torque actual. In some cases drive setup variables also need to be modified. The most typical one being acceleration or deceleration rate.

In this application note, the following information is sent down to the drive:

- Drive Frequency reference (40,013)
- Drive Start and Stop commands (45,102)

The following information is read back from the drive:

- Drive Output frequency (40,001)
- Motor current (40,003)
- % Rated torque (40,004)
- Drive status, including drive ready, running, and faulted information (45,101)

The read data transfer is pre-built into a fixed polling list. The frequency reference and the drive start and stop commands are also written to the drives using the poll list.

Performance

Typically there can be on an average 9.5 messages per second on one Modbus serial link to ACS 500 Drives.

In this example we have two reads and two writes for each one of the drives, with a total number of four drives. This means that our poll-list has 4*(2+2) = 16 messages. With 9.5 messages per second, it will take 1.7 seconds to complete the whole list.

Example Applications

Applications involving the Modbus connection can be found in many industrial sectors. This solution is not intended for building mechanically connected systems, but is suited for applications where

- Discrete wiring cost would be reduced.
- Increased information from the drives is needed beyond the 2 Analog and 3 Digital signals available on the drive.
- Performance is suitable for the application.

Control performance can be improved by mixing discrete-I/O control with serial communication.

Solution Overview

This solution is done using the standard ACS 500 drives with the CRU03E firmware revision, and insertion of the 1150-MBM firmware chip into a 1746-BAS module. In a typical implementation, the following steps are needed for a successful implementation:

- Install firmware and configure the 1746-BAS hardware
- Decide the use of memory in the basic module
- Decide the use of memory in the PLC
- Identify the command list
- Identify the module configuration parameters
- Develop ladder logic
- Program the ACS 500 drives
- Wire the drives to the PLC

Install and Configure Hardware

The 1150-MBM is shipped as an EPROM chip which needs to be placed into the U18 User Socket. Align the notches on the EPROM plastic carrier with the notches in the User socket. Make sure the EPROM is well seated.

Set the jumpers on the 1746-BAS module

- JW1 Across 7-8 and 9-10 (RS-485)
- JW3 Across 3-5 and 4-6 (Enable software)
- JW4 Across 1-3 and 2-4

Memory on Basic Module

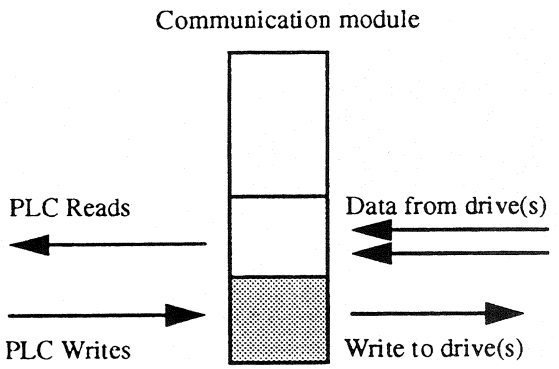
Communication between the SLC 500 and the protocol module occur through the backplane through COP instructions. Data is transferred between the module and the PLC in blocks, with the first word in a block being a Block Identifier followed by data. Every block has its own use.

The Blocks are classified as follows:

- Data [0 – 79]
- Command [80 – 99]
- Event Initiated Writes [100 – 119]
- Configuration [255]
- Error codes [253, 254]

The data blocks [0 – 79] are especially important. There are a total of 80 data blocks, each with a length of 50 words, totalling 4000 registers.

This is the area where data from the PLC is written for a subsequent transmission to the drives, and where the data received from the drives is stored for later transmission to the PLC.



The data is transferred independently from the communication module to the drives, and from the communication module to the PLC. The addresses on the communication module buffer do not match the actual drive addressees.

Memory on PLC

The PLC has all the data needed stored into its integer files. This data includes the configuration information for the communication module, as well as the actual control and feedback data. To simplify, the different uses are divided into separate integer files for clarity.

Configuration data N7:0 – N7:10

This is explained in the following section.

Write data buffer N7:50 – N7:101

This is used for storing the data to be written to the communication module.

Command status N7:110

These words are used to store the return statuses received from the module.

Read data buffer N7:150 – N7:201

This is used to receive the data from the communication module.

Write data to slave N10:0 – N10:49

These 50 words are set for the Block ID #0 for the data to be written to the drive during a command list write command. These words correspond to addresses 0 – 49 in the communication module.

Command list N9:0 – N9:149

This list contains the first fifteen available commands for the module.

Read data from slave N10:50 – N10:99

For this application, the data read from the drives starts at address 50 and goes to 100. This corresponds to Block ID #1. The first block is reserved for write message data.

Identify command list

The command list for the module lists all the reads and writes which the modbus master module will execute in sequence. In this case one way to make the command list is:

Command list

	0	1	2	3	4	5
N9:0	1	1	3	0	4	50
N9:10	2	2	3	0	4	60
N9:20	3	3	3	0	4	70
N9:30	4	4	3	0	4	80
N9:40	5	1	6	0	1	12
N9:50	6	2	6	1	1	12
N9:60	7	3	6	2	1	12
N9:70	8	4	6	3	1	12
N9:80	9	1	6	4	1	5101
N9:90	10	2	6	5	1	5101
N9:100	11	3	6	6	1	5101
N9:110	12	4	6	7	1	5101
N9:120	13	1	3	5100	1	55
N9:130	14	2	3	5100	1	65
N9:140	15	3	3	5100	1	75
N9:150	16	4	3	5100	1	85

Example for a read:

- Row N9:20 – N9:25 defines one read
- Message number 3 (N20:0)
- Read from drive 3 (N9:21)
- Using modbus command 3 (N9:22, read registers)
- Starting from address 40001 (N9:23 = 40001 - 40000 - 1)
- Four parameters (N9:24)
- Place the read parameters into registers starting from 70 (N9:25).

Example for a write:

- Row N9:110 – N9:115 defines one write
- Message number 12
- Write to drive 4
- Use modbus command 6, write one register
- Get the data from register 7 in the communication master
- Write one parameter 1
- Write to 5101 (= 45102 - 40000 - 1)

Module Configuration

The module configuration has the major communication setups. This is in table N7:0 – N7:10.

Module Configuration

	0, 5	1, 6	2, 7	3, 8	4, 9
N7:0	2	1	5	0	8
N7:5	5	4	0	4096	0
N7:10	0	–	–	–	–

This table has in the example the following information:

- N7:0 Parity: 2 (Even)
- N7:1 Stop bits: 1 (One stop bit)
- N7:2 Baud rate: 5 (9600 Baud)
- N7:3 RTS to TxD delay: 0
- N7:4 RTS Off-delay: 8
- N7:5 Data block count: 5
- N7:6 BT Count per poll 4
- N7:7 Poll list delay: 0
- N7:8 Message response Time-out: 4096 (~ 250 ms)
- N7:9 Poll list delay: 0
- N7:10 RTU/ASCII Mode: 0 (RTU)

A more detailed description of these options is in the user's manual. On the ACS 500 drive, RTU mode must be selected. Also, the value of RTS Off-delay must be set correctly corresponding to the selected Baud rate.

Data in PLC

The data in the PLC is in the example in two Blocks. Block 0 is for the data to be written to the drives, and Block 1 is used for the data being read from the drives.

In this example, the data is in the following registers:

Write Frequency reference to drives

	0	1	2	3
N10:0	**Ref1**	**Ref2**	**Ref3**	**Ref4**

Write Start / Stop command to drives

	4	5	6	7
N10:4	**SS1**	**SS2**	**SS3**	**SS4**

The references are in registers N10:0 – N10:3. The values are integers, where a value of 100 corresponds to 1.00 Hz. To send a reference of 12.34 Hz to drive number 2, place a value of 1234 to N10:1.

The start and stop controls are in registers N10:4 – N10:7. Each drive has in this example its own individual start and stop control. To start the drive, place a value 16 to the corresponding register. A value of 0 will stop the drive.

The drive actual values are in the following registers:

Drive actual values

	0	1	2	3
N10:50	**Frq**	**Spd**	**Curr**	**Torq**
N10:60	**Frq**	**Spd**	**Curr**	**Torq**
N10:70	**Frq**	**Spd**	**Curr**	**Torq**
N10:80	**Frq**	**Spd**	**Curr**	**Torq**

Drive Status

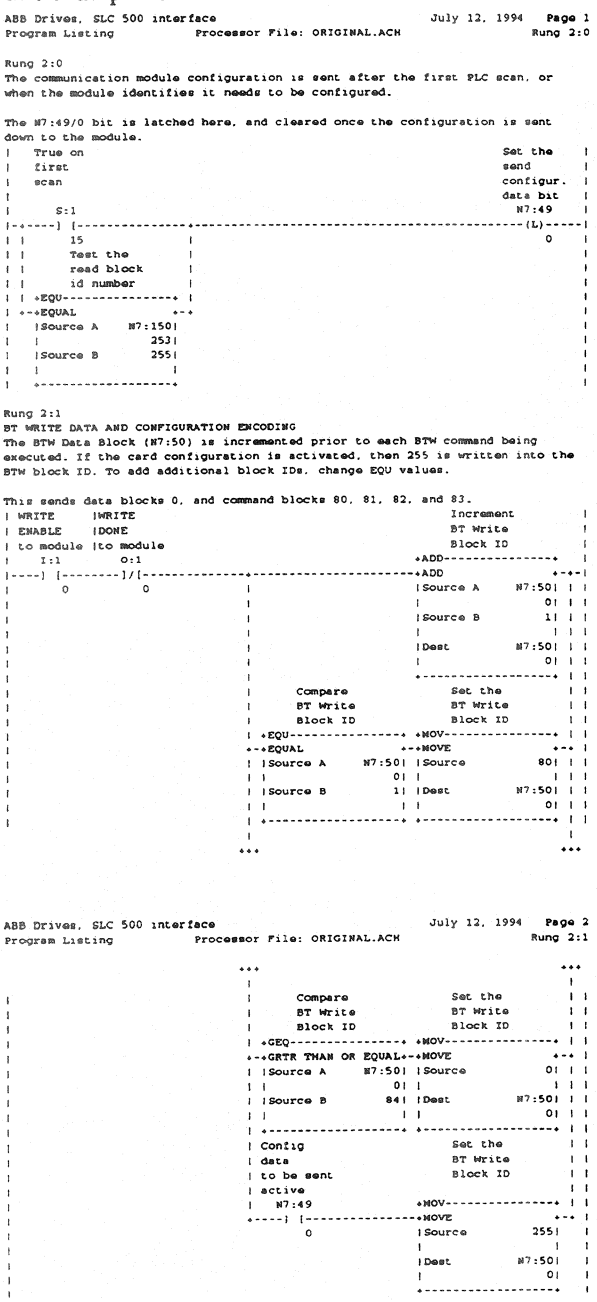
	N10:55	N10:65	N10:75	N10:85
	Drv1	**Drv2**	**Drv3**	**Drv4**

The actual values for drive 2 are in N10:60 – N10:63. N10:60 is the actual frequency, N10:61 is the drive speed, N10:62 is the motor current, and N10:63 is the torque.

The drive statuses are also stored in the N10 table. The status for drive 2 is in N10:65.

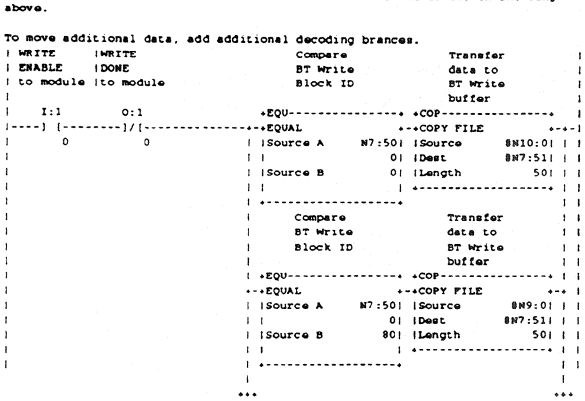
Ladder Logic

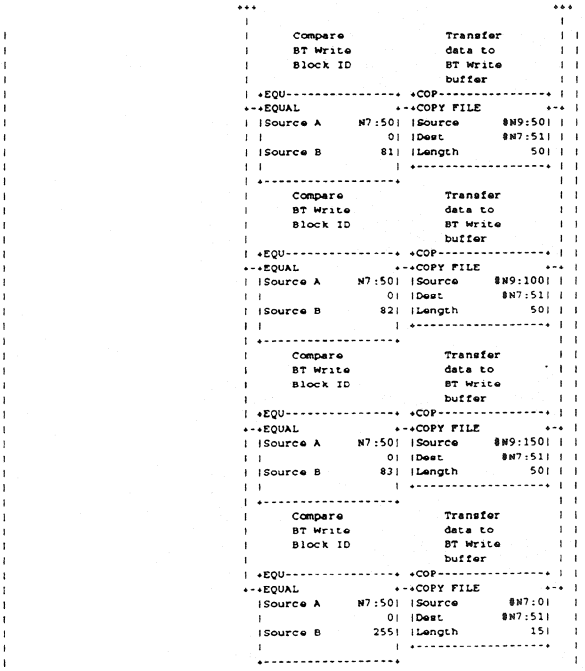
The PLC must be programmed and configured to include the communication in between the PLC and the communication module. An example ladder code for this example is:



Rung 2:2
WRITE DATA AND CONFIGURATION DATA TO MODULE
Based on the value in the BTW Block ID, move the data into the BT Write buffer area starting at N7:51. N7:50 is the Block ID number, and is set on the rung above.

To move additional data, add additional decoding brances.





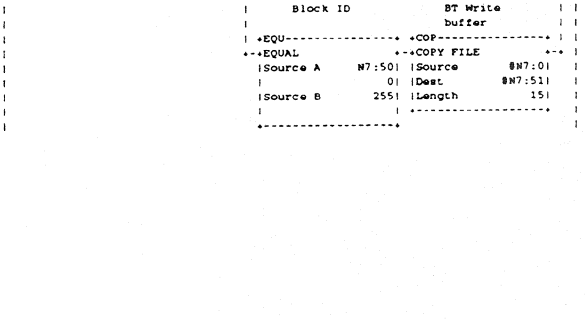
Rung 2:3
BT WRITE
"Block Transfer" Write to the module.

The Block transfer is taking the data from a fixed buffer starting at N7:50, and having a fixed length of 51. This is why the actual data was transferred into this buffer area in the two rungs above.

Rung 2:4
BT READ
"Block Transfer" Read from the module.

The read data is placed into a fixed buffer starting at N7:150. The N7:150 is the Block ID number for the data being read, and the following words contain the actual data received.

Rung 2:5
MOVE DATA AND ERROR STATUS FROM THE MODULE
After the Block Transfer read is complete, place the returned data into the correct areas in the PLC. In this example, one Block ID is read (N7:150 = 1), and the data is placed to buffer at N10:50 - N10:99. The BT Read Error status is returned, and stored to buffer N7:110.
To add more blocks, add a EQU and COP branch.



Rung 2:6

Program the ACS 500

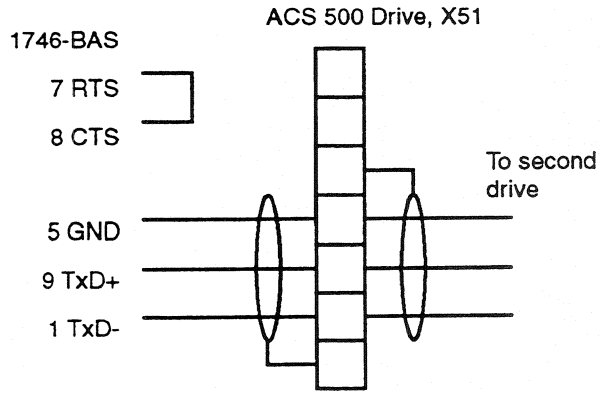
The ACS 500 drives need to be programmed for this application. In this case we wanted to use serial communication to both start and stop the drive, and to control the drive frequency.

To do this, the following steps will setup the drive properly:

- Load the factory macro
- Set the drive to EXTERNAL control. (Operational data parameter 9)
- Set 10.1.1 EXT 1 STRT/STP/DIR to STD COMM. This enables the start and stop control using serial communication.
- Set 10.2.2 EXTERNAL REF1 SEL to STD COMM. This enables the frequency reference to be sent using serial communication.
- Set 10.8.1 DRIVE ID-NUMBER to a number 1, 2, 3, or 4. Each drive on this example has to have a unique modbus station address.
- Set 10.8.2 PROTOCOL to MODBUS. If the previous value was GS-BUS, power the drive down and up to make the change.
- Set 10.8.3 BIT RATE SELECT to 9600.
- Set 10.8.4 PARITY to EVEN
- Power the drive down, and back up again. The changes in the group 10.8 take effect only on drive power up sequence.

Wiring

The communication master module needs to be connected to the drives using a serial cable. The cable makeup necessary to interface with the drives is:



In the modbus communication manual for the ACS 500 there are detailed description for wiring, grounding, and for proper termination of communication cable.

Contact

The 1150-MBM firmware is a ProSoft Technologies product, and needs to be ordered from them. Also ProSoft has a Technical Support line for questions on using the 1150-MBM firmware on the 1746-BAS module.

Order Placement, 1150-MBM

ProSoft Technologies
Kim Tatman, Office Manager
(805) 327 7066 / Phone
(805) 327 7322 / Fax

Technical Support, 1150-MBM

ProSoft Technologies
Wilbur Wong, Applications Engineer
(800) 326 7066 / Phone

Sales Support, South East US

Janice Hungerford, Account Manager
(713) 999 7565 / Phone
(713) 999 0823 / Fax

Sales Support, All other locations

Kim Tatman, Office Manager
(805) 327 7066 / Phone
(805) 327 7322 / Fax

Technical Support, ACS 500 Drives

ABB Drives
Standard Drives Tech Support
(800) 243 4384

Sales, ACS 500 Drives

ABB Standard and Process Drives
Cathy Holms
(414) 785 3416

ST-271

ABB Industrial Systems Inc.
Standard Drives Division
16250 W. Glendale Drive
New Berlin, WI 53151-2840
Tel: (414) 785 3416
(800) 752 0696
Fax: (414) 785 0397