MODBUS / DF1 (PLC5/SLC) TRANSLATOR
OPERATOR'S MANUAL

Calta Computer Systems Ltd.
Feb. 18, 2003

GENERAL

The standard “PLC09 Converter” hardware is equipped with specialized software that provides a bi-directional conversion between Modbus and Allen Bradley DF1 protocols (full duplex only) The PLC5 protocol is used in the translator.

The PLC09 Converter continuously polls the DF1 slave device and stores the data in its internal database. The PLC09 Converter responds as a slave to Modbus requests received on the Port 1. A status register is defined which reflects the current DF1 communications status back to the Modbus host.

The PLC09 Converter is capable of being mapped on a point by point basis (coils in groups of 16) by downloading a configuration file to Port 2. The configuration file is a standard ASCII text file that can be maintained on the PC with a text editor. (“Notepad” for example). The configuration file is stored in nonvolatile RAM in the PLC09, therefore once configured the PC link can be removed until a configuration change is required. By maintaining a separate configuration file for each site the standard PLC09 can be quickly reconfigured for different locations.

The Port 4 accepts messages from a Host using DF1 protocol. Access to the PLC on Port 3 is shared with the Modbus to DF1 task.

OPERATION

On power up, the “Converter” checks EPROM, RAM, and the checksums for the data tables. If there is a problem, the LED on the front of the unit slowly blinks once for an EPROM problem, twice for a RAM problem, and three times when it is in need of a Downline load. When data is output from the “Converter” to the PLC on Port 3, the Front LED blinks rapidly. If a fault condition was detected upon power up however, it takes precedence over this feature. The “Converter” uses the mapping tables currently retained in battery backed up ram memory to poll the slave device using DF1 format, remap and store the data in the “Converter”. When requested by the Modbus host, the remapped data is returned in Modbus format. Note that the DF1 status register should always be polled as part of the data request to determine that the returned data is current. Note that specification of the status register is part of the download configuration. Pending commands from the host are serviced between each data request. Servicing consists of remapping the command, converting it to DF1 format and sending it to the slave, then returning the response to the host. Standard Converter operation is identified by a bright LED on the front of the unit that flickers each time data is sent to the slave.

Commands received when a DF1 poll is active are queued in the Converter until the current data poll is complete. The queued command is then sent to the slave device. An acknowledge is sent back to the host upon receipt of acknowledgement from the slave. Note that the host command timeout must be long enough to allow this entire sequence to complete. Integers may be sent one at a time (function code 6) or more than one up to the Modbus protocol limit using function code 16. For multiple writes, only the first register in a sequence has to be configured. It uses this information to do the mapping. <u>This means that the Allen-Bradley elements to be mapped must be sequential with no gaps, the same as the Modbus registers</u>. Floating Point values are always sent using function code 16 and the same conditions apply as multiple integer writes.

The Converter is configured to communicate with the PLC on Port 3 using 8 data bits, 1 start, 1 stop and even or odd or none parity using CRC or BCC error detection. As well, the baud rate is set from the configuration file.

CONFIGURATION

"Converter" mapping tables may be modified by downloading an ASCII configuration file from a PC. The configuration file can be maintained with any ASCII text editor such as Windows Notepad. The file is transferred to the PLC09 Converter using "Terminal" under Windows 3.1 or "Hyperterminal" under Windows 95/NT. The PC port used for downloading must be configured for 9600 baud, 8 data, no parity and 1 stop. The file must be transferred as a text file using Hyperterminal. This is done by selecting the "Transfer" item and then "Send Text File" from the dropdown list. Select the directory and file and select "Open" to commence sending the file. There is a 500-byte configuration buffer used to processes the configuration file as it is downloaded. If the file is downloaded too quickly the download process will not complete. If this happens place a small delay after each line. This is a configurable parameter in most terminal emulation programs such as Hyper Terminal. (See Appendix B for full download instructions using HyperTerminal).

Once the "Converter" receives the string "BEGIN" normal operation suspends and the "Converter" enters the configuration mode. This mode is identified by a message DOWNLOAD HAS BEGUN. Data following the start string will consist of data packets separated by carriage return/line feed. Commas separate individual items in each packet. Each data packet must contain 4 data items. Special type packets contain a fifth specifying name. The last data packet is followed by "ENDX". When this string is received the downloaded file is checked for validity. If OK, normal operation resumes as by the message ""DOWNLOAD HAS ENDED SUCCESSFULLY". If any of the following inconsistencies are found an error message is output. If the error can be traced to a configuration line, the line the error has occurred on is displayed. The error can then be corrected and the "Converter" rearmed for download by pressing the Esc key and resending the file.

Possible errors are as follow:

1. duplicate registers defined
2. invalid mapping
3. invalid format
4. point count for type exceeded
5. invalid register number
6. poll list size exceeded
7. input buffer queue overrun
8. baud rate problem
9. rts on problem
10. timeout problem
11. miscellaneous problem
12. plc additional problem

There are also additional error messages such as the poll list size has exceeded memory capability that pinpoint the area in which the problem occurred.

Once verified, the new mapping table replaces the existing one in battery backed up ram memory and the PLC09 is restarted.

DOWNLOAD CONFIGURATION FILE

The Converter is configured from an ASCII file maintained on the PC.

To ensure correct synchronization with the PLC09 the first line of the configuration file must contain a single carriage return.

The file is structured in a line format with each new entry requiring a new line. Comments may be freely added anywhere in the file. Comment lines are identified by a * in the first column. All text in a line starting with a * is ignored. The file data must begin with the string “BEGIN” and end with the string “ENDX”. Between the two strings each line consists of a data packet. This packet may be either a standard mapping packet or a special packet which is used to set specific parameters. Each type is discussed below:

Standard Packet Configuration

Each packet consists of four comma separated values followed by a carriage return, line feed packet terminator. For mapping tables the values are interpreted as follows:

value 1 - destination (host register/group number)
value 2 - file type in slave device
value 3 - file number in slave device
value 4 - address in slave device

where

the destination refers to the modbus register number for holding and input register and the group number for coil and status types. The group number is defined as the coil (status) number divided by 16.

the file type is defined as follows:
0 – Binary (B)
1 – Integer (N)
2 – Floating Point (F)
3 – Binary Write (B)
4 – Integer Write (N)
5 – Floating Point Write (F)

Special Packet Configuration

A number of special configuration packets are provided to permit field modification of special parameters. These consist of a standard packet as described above preceded with a special ASCII identifier and a comma. Following is a description of the special packets

BAUD

value 1 - Port 1 Baud Rate (Modbus Host)
value 2 – Port 3 Baud Rate (PLC)
value 3 - Port 4 Baud Rate (DF1 Host)
value 4 - spare
The following baud rates are supported 19200, 9600, 4800, 2400, 1200 and 300.

RTS

value 1 - Port 1 RTS Delay (sec * 100). Max 2.55 seconds
value 2 - Port 3 (PLC) RTS Delay (sec * 100). Max 2.55 seconds
value 3 - spare
value 4 - spare

Note - The RTS delay time is the time between raising RTS and transmission of the first data byte.

TMO

value 1 - Port 1 Intercharacter Time-out (sec * 100). Max 2.55 seconds
value 2 - Port 3 (PLC) Receive Data Time-out (sec * 10). Max 25.5 seconds
value 3 - Port 3 (PLC) Poll Delay (sec * 10). Max 25.5 seconds
value 4 - spare

Note - The Port 1 Interchar Timeout is the time between characters received from the host beyond which the message is declared invalid.

The Port 3 (PLC) Receive Data Timeout is the amount of time following a PLC data request that the Converter will wait for an answer.

The PLC Poll Delay is the amount of time the Converter will wait before polling for the next data.

PLC

value 1 - Destination PLC address (decimal)
value 2 - PLC parity (0=none, 1=even, 2=odd)
value 3 - Data addressing (0=logical, 1=base)
value 4 – Destination file in PLC / “Src” address (decimal)

where:

The destination PLC address is the address used as the “DST” value in the DF1 message.

The PLC parity should be set to 0 for none, 1 for even and 2 for odd.

The data addressing is set to 0 for logical (PLC 5) type addressing or 1 for base type addressing.

The destination file is the file which the remote will use for the data exchange. It is the number to which the “SRC” value in the DF1 message is set to.

MISC

value 1 - Converter Modbus Address
value 2 - Error Register
value 3 – Error Mode
value 4 – PLC security

where:

The Converter address is the Modbus address the host uses to address the Converter

The error register is the register used to store the current DF1 communication status. A nonzero value in this register indicates a DF1 communication problem.

The error mode indicates the format of the error value. A 0 indicates an integer format and a 1 indicates a floating point format.

The PLC security is set to a 0 for CRC security checking and 1 for BCC.

Note -- the values identified, as spare above should be 0 filled in the file.

ADDRESSING NOTE

KF2 Addressing

When addressing a PLC through a KF2 card the “DST” address should be set to the PLC number the KF2 card is connected to. Be careful, as some PLC addresses are in octal whereas the numbers entered in the configuration file are decimal. The file number of interest is set into the data address of the KF2 card. In this configuration the “SRC” value in the configuration should be set to 0.

When addressing a PLC directly through a modem port the “DST” value should be set to the PLC address and the “SRC” value set to the number of the file which is to act as interface.

SPECIFICATIONS

- the following are maximum numbers for each data type

Data Type	Maximum Number
coils	1000
status	200
input	200
holding	2000

APPENDIX A

A configuration template file follows:

```
*       Converter Configuration File
*      NOTE always begin the file with a single carriage return / line
*      feed on the first line
*       Each line with a * as the first character is treated as a comment
*       NOTE -- DO NOT PUT COMMENTS ON DOWNLOAD LINES
*       History
*       Date               Author             Description
*       Apr 29/98         LRS/Calta          Original test
*       Nov 25/98         LRS/Calta          Upgraded documentation
*       Dec 5/98          LRS/Calta          Changed control definition
*
*       Identify the start of download
BEGIN
*
*       Set the baud rate (Port 1, Port 3 (PLC), Port 4, spare)
*       Note -- When changing Baud Rate change the Baud Rate, download
*               then restart the Converter for the new Baud Rate to
*               take effect
BAUD,1200,9600,9600,0
*
*       Set the RTS delays (Port 1, Port 3 (PLC), spare, spare)
*       Note that the RTS delay is the time between raising RTS and
*       sending the first data
*       Values are in hundredths of seconds Max value is 255 (2.55 sec) RTS,2,3,0,0
*
*       Set the delay and timeout values
*       (Port 1 interchar, PLC Rx data timeout, PLC poll delay, spare)
*       Interchar -- time between characters from the host before the
*                    message is declared invalid
*       RX Data -- time the Converter waits for a PLC message before
*                  declaring the message invalid
*       Poll Delay -- time the Converter waits between data polls
*
*       Interchar value is in hundredths of seconds Max value is 255 (2.55 sec)
*       Other values are in tenths of seconds Max value is 255 (25.5 sec)
TMO,15,80,40,0
*
*
*       Set the miscellaneous parameters
*       (Converter address, error register, error mode, PLC security)
*            Arbitrator address is the Modbus address of the Arbitrator
*            Error register -- register number for storage of communication
*                        status.  A bit set in the register identifies
*                        a communications failure.  The bit position
*                        identifies which poll is failing.  Note that
*                        this register must be either an input or
*                        holding register.
*       Error mode indicates the format of the error.
*         0 is integer, 1 is floating point
*       PLC Security indicates the type of security on the PLC message
*         0 is for CRC (2 bytes), 1 is for BCC (1 byte)
MISC,46,40001,0,1
*
*            Set the PLC miscellaneous additional parameters
*       (PLC address, parity, address type, source address)
*            PLC address used as "Dst" in PLC message
*            PLC Parity (0=none, 1=even, 2=odd)
*            Address type (0=logical, 1=base)
*            Source address used as "Src" in PLC message
PLC,8,0,0,0
*
*       Define the cross mapping tables
*       Each entry consists of 4 comma separated values
```

```
*       (Destination Reg #, Source File Type, Source File #, Source Address)
*
*       The source file type is decoded as follows:
*                   0 - B (binary)
*                   1 - N (integer)
*                   2 - F (floating point)
*                   3 - binary write
*                   4 - integer write
*                   5 - floating point write
*
*
*      The source file number is the Allen Bradley file number for the data
*
*      The source address is the Allen Bradley address for the data.
*
*       The number of the destination register determines its data type.
*      Coils start at 1
*      Status Inputs start at 10001
*      Input Registers start at 30001
*      Holding Registers start at 40001
*
*      The ending register for each type is determined from the maximum
*      of each type supported in an individual application
*
*       Note - floating point numbers use two contiguous registers
*              only the first is required in the list
*            - floating point numbers must be an odd register number
*            - coil and status point types are converted in groups of 16 with
*              the register number represented by the group number
*              For example group 1 contains coils 1 to 16.
*
*       Data Poll Requests
*
*      Coils
*             Read Group 3 from B7:21
3,0,7,21
*
*       Registers
* Note 40001 is defined in MISC above as the communications register
* therefore it should not be used to store any other data.  The host
* must poll for this register as it is the only way to check the
* validity of the returned data.
*
*             Read 16 status bits from B7:10 as holding register 2
40002,0,7,10
*
*             Read a 16 bit integer from N20:6 as input register 8
30008,1,20,6
*
*             Read a floating point number from F8:2 as holding register 7
*                    note that register number must be odd
40007,2,8,2
*
*       Write Requests
*             Note that the required write information is obtained
*             from the host Modbus message.
*      Coils
*             Coil writes also utilize the group concept where a group represents 16
*             coils.  Defining a write group enables the writing of any coil in that
*             group.
*
*             Associate group 6 with B3:35
6,3,3,35
*
*       Registers
*
*             Associate register 40017 with N6:2
40017,4,6,2
*
```

```
*             Associate register 40005 with F9:21
40005,5,9,21
*
*       Identify the end of download
ENDX
```

APPENDIX B

Downloading Configuration Files to the PLC09 Converter MOD_DF1 Model

The unit is supplied with a floppy containing two files – plc09.ht (configuration file for HyperTerminal for 95/98/NT) and testcon.txt (mod_df1 configuration file).

The following describes the procedure to send the example configuration contained in file testcon.txt to the Converter using HyperTerminal.

1. Put the floppy in the A drive of the PC.
2. Under Start/Run type a:\plc09.ht
3. HyperTerminal will start up with the configuration necessary to talk to the Converter.
4. HyperTerminal will be set up to communicate via Com port 1. If this is correct, skip to item 6.
5. To change the PC Com port, in the HyperTerminal Window, select File/Properties/Connect To. Connect using should show Direct To Com1. Select the Direct To port to be used. Then click on OK.
6. To send the configuration file testcon.txt to the Converter make sure a cable is connected from the correct PC Com port to Port 2 on the Converter. Then select the HyperTerminal Menu item Send File.
7. Then select Transfer.
8. Then select Send Text File.
9. Then select testcon.txt on A:
10. Then select Open.
11. The file should now begin to be transferred.
12. The message DOWNLOAD HAS BEGUN should appear.
13. If the Download is successful, the message DOWNLOAD HAS ENDED SUCCESSFULLY should appear after a significant time delay depending upon the size of the Configuration file. If all is well, skip to item 16.
14. If the Converter encounters an error, the line number and line containing the error will appear on the Monitor.
15. Hit Escape (ARMED FOR DOWNLOAD should appear), correct the Configuration file and repeat items 6 to 11.
16. When the file has been received correctly with no errors, the last message ARMED FOR DOWNLOAD should appear without having to hit Escape.