Application Note

2000R
AN-79A-00

Substation Automation and Protection Division

Line Sectionalizing Using A, PLC And ABB Protective Relays

ABSTRACT: With the advent of utility deregulation, uptime and minimal line restoration times are demanded. With the economics of off-the-shelf equipment usage for substation control and decision making, PLC, and ABB relay use is widely accepted as a restoration solution. This advanced application note explains a method to inexpensively implement advanced Line Sectionalizing techniques using a TPU2000 R, DPU 2000R, PCD 2000, and a programmable logic controller. THE DISCUSSION AND LADDER LOGIC USED HEREIN IS TO BE USED AS A GUIDE TO PLC AND PROTECTIVE RELAY INTEGRATION.

Typical Installation

Figure 1 illustrates a typical installation in which protective relays are installed within a substation providing for restoration schemes. Both the network architecture and sample modified one line is shown for clarity.

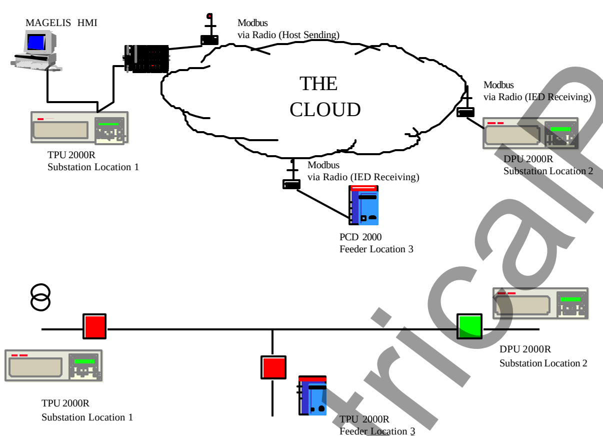


Figure 1 -Typical PLC/Protective Relay Report by Exception Application

The scenario illustrated is indicative of a line sectionalizing (load shedding) installation. Using intelligent off the shelf IEDs such as protective relays and PC based HMI interfaces and programmable logic controllers to analyze and perform intelligent switching decisions is more commonplace given the advantages afforded by the economics and increased functionality of such a system. Using an ABB protective relay and a PLC is a logical decision because:

- All items are commercially available Off The Shelf units
- The installations use inexpensive radio modems.
- The software to perform the tasks is programmed in ladder logic, which can be easily written by either a system house or a utility engineer.
- The DPU 2000R, TPU 2000R, or PCD may be field retrofitted by the user, if the Modbus Plus or Modbus protocols are not presently installed in the units.

- No proprietary protocols or equipment is utilized in this installation.
- An inexpensive operation interface allowing for visualization of local and remote status/operation is available.
- Real Time Switching is based on instantaneous decisions made by the microprocessor based IEDs.

As illustrated in Figure 1, the feeder and substation 2 are located a great distance away from the main PLC. Inexpensive radio scatter modems are used to allow communication between the PLC which has the RTU based functionality and logic imbedded within it and the remote IEDs.

Line Sectionalizing Explained

This application note is intended to illustrate the method of obtaining information from the IED's through the Modbus and Modbus Plus interfaces. Each one of the nodes, PLC, IED and HMI operate in concert as follows:

1. The PLC reads/writes/calculates information obtained from the TPU 2000R via the Modbus Plus network. Data gathered from the TPU 2000R is:
   - Breaker Status (52a, 52b) is read
   - Cumulative Watts, Vars, 3 Phase Power, Amps, Volts, Watts, Var values are read.
   - Calculation of the combined loading is performed when the Substation Location 1 is feeding the lines at Location 3 (PCD 2000).

2. The PLC reads/writes information between itself and the DPU 2000R. Data gathered by the PLC is:
   - Breaker Status (52a, 52b) is read.
   - Cumulative Watts (per phase and 3 phase), Vars (per phase and 3 phase), Amps, Volts, frequency values are read.
   - Calculation of the combined loading is performed when the Substation Location 2 is feeding the lines at Location 3 (PCD 2000).
   - Control Operation capabilities such as breaker trip and breaker close can be completed automatically (via logical decisions made by the PLC) or manually (via an operator at the HMI station).

3. The PLC reads/ writes information between itself and the PCD 2000. Data gathered by the PLC is:
   - Breaker Status (52 a, 52 b) is read.
   - Cumulative Watts (per phase and 3 phase), Vars (per phase and 3 phase), Amps, Volts, frequency values are read.
   - Control Operation capabilities such as breaker trip and breaker close can be completed automatically (via logical decisions made by the PLC) or manually (via an operator at the HMI station).

4. The MAGELIS HMI displays the data values held in the PLC. In this installation, the PLC acts as a data concentrator and arbitrator for the logic operations. The MAGELIS HMI can disable the automatic restoration functions preprogrammed in the PLC. With the PLC placed in manual control mode, the operator viewing the data on the MAGELIS screen can perform manual restoration since visualization of each breaker and all feeder metering values are displayed on the screen.

There are two modes of operation, MANUAL and AUTOMATIC restoration. With the system in AUTOMATIC mode, line restoration is performed by the PLC without any operator intervention. AUTOMATIC MODE operation occurs as such:

1. The breakers at Substation Location 1 and Feeder Location 3 are closed. The breaker at Substation Location 2 is open. In this example, the feed from the TPU 2000R is providing supply to the feeder at location 3. The PLC is constantly reading metering values and calculating the average load required by Feeder Location 3.
2. On the event of a fault, the TPU 2000R trips the breaker. The PLC recognizes the trip immediately (under 10 mS in this case) and immediately determines if sufficient reserve is available at Substation Location 2 to supply Feeder Location 3.
3. The PLC immediately opens the breaker at Feeder Location 3. The PLC verifies that reading the status of the breaker opens the breaker. The MAGELIS system immediately displays the metering values and breaker status on its screen and may also generate alarms to alert the operator or attached SCADA system.
4. If Substation Location 2 has sufficient reserve to supply the feeder at location 3, the PLC wait 3 seconds to ensure that the TPU at Substation Location 1's breaker is open, AND that the Breaker at Substation Location is also open.
5. The PLC shall Close the breaker at Substation Location 2 and read the metering values and breaker status reported by the DPU 2000R.
6. The PLC still calculates the load on the line and determines if there is still sufficient reserve to add additional feeder lines, the PLC shall wait another 1 second and send a close command to the breaker controlled by the PCD 2000.
7. The PLC shall read the metering values and breaker statuses at each of the 3 IED locations and report them to the MAGELIS system via the internal network at the substation.
8. At each of the above steps, a message is generated as to the step being performed by the PLC. The status of each step (along with any error) messages is displayed and archived by MAGELIS operator interface. The PLC will place the restoration scheme in MANUAL mode, so that an operator may place the system in the same state that it was in prior to the TPU 2000R trip.

IF any one of the steps fails to execute, the MAGELIS HMI will display an error message as to the cause of failure in the restoration process.

MANUAL MODE disables the PLC's capability to trip/close the breaker. The PLC still computes the loading values and alerts the operator as to alarms. The PLC also communicates with MAGELIS MMI and displays messages/ breaker status information/metering data informing the operator if adequate load is available to supply the feeder. This gives the operator additional information if a manual restoration is to be performed via the operator interface. Additionally, the operator commands may be sent directly to the PLC to perform manual trips and close commands.

Method Of Implementation

Two Ladder Logic instructions within the Modicon PLC allows line sectionalizing to occur:

- MSTR obtains the data from the TPU relay. Once the data is obtained, the PLC determines the field conditions and decides upon the control to be performed.
- XMIT instructions when executed by the PLC, initializes the COM port resident on the unit. The PLC can then act as an Remote Terminal Unit (RTU). The PLC then interrogates the DPU 2000R and PCD 2000R to determine if the units are available to be switched.

The remainder of this application note explains the programming process and ladder logic required to implement the application pictured in Figure 1.

Obtaining Relay Information Via MODBUS Plus And MODBUS

The PLC is programmed using four ladder logic segments. The logic within each segment is as follows (for this example).

SEGMENT 1: READ TPU 2000R VALUES VIA MODBUS PLUS. PROVIDE MANUAL TPU CONTROL OPERATIONS VIA MODBUS PLUS.

SEGMENT 2: READ DPU 2000R AND PCD 2000R VALUES VIA THE RADIO MODEMS USING MDOBUS PROTOCOL. PROVIDE MANUAL AND AUTOMATED CONTROL OPERATIONS VIA MODBUS.

SEGMENT 3: LOGIC REQUIRED FOR EACH STEP IN THE RESTORATION SEQUENCE FOR CALCULATION AND CONTROL. HMI LOGIC FOR TRIGGERNG MESSAGES AND ALARMS IN THE MAGELIS SYSTEM.

SEGMENT 4: ANCILLARY SUBROUTINES providing 32 bit number conversion since the Compact 984 does not easily allow for mathematics on a double register number or single register number containing a value of 9999 or greater. The ABB products allow numbers to be reported in a single register interpreted as 0 to 65535 (Unipolar) or –32767 to 32768 (Bipolar). If a number is a 32 bit representation, PLC logic must be added to the specific vendor's PLC allow computations to occur.

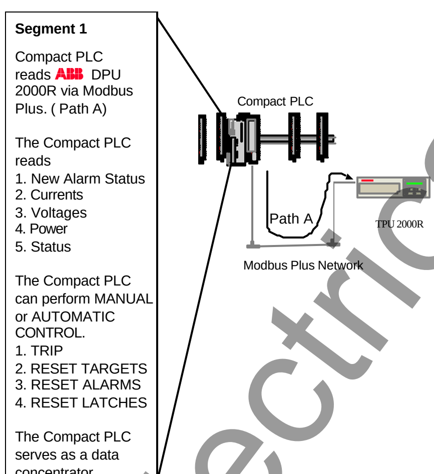


Figure 2 - Modbus Plus Ladder Logic Implementation Strategy.

The Ladder Logic is Segment 1 is very straight forward. Figure 3 illustrates the method of obtaining the data from the TPU 2000R via Modbus Plus.

The data requested includes logical element status which includes breaker trip information for phases A,B, and C. The phase information is latched, and its status must be reset by the operator to annunciate the alarm which is decoded by the PLC.

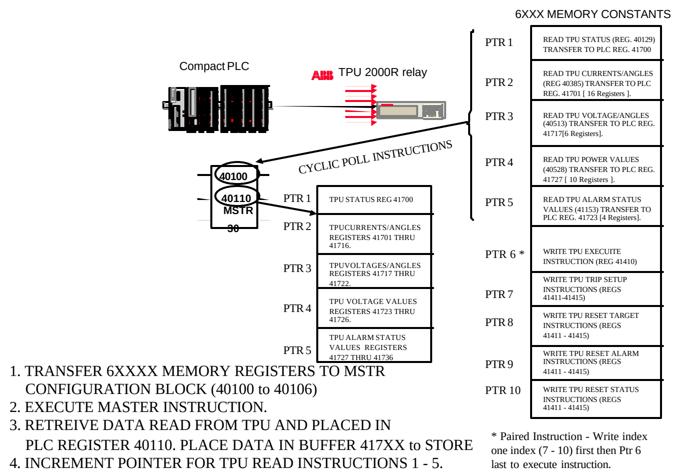


Figure 3 - Data Map Request from the Compact PLC and TPU 2000R via MODBUS Plus

The MSTR block logic follows.

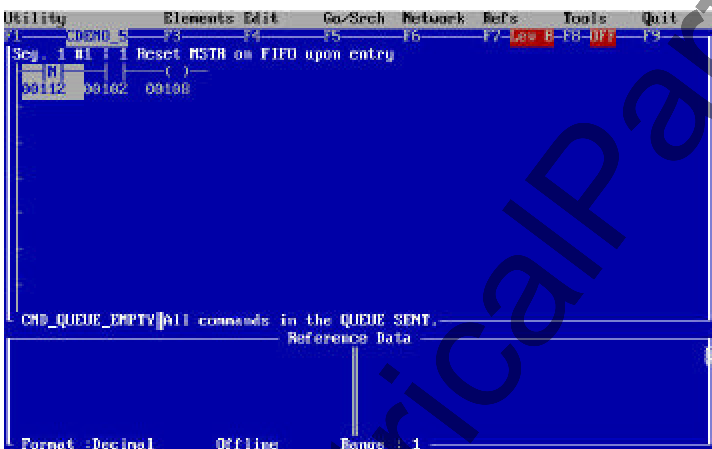


Figure 4 - Network 1 Ladder Logic (MSTR Modbus Plus Logic)

NETWORK 1 – FIGURE 4:

The logic is written with cyclic polling occurring to gather the data and place it in a 4X memory space as illustrated in Figure 3. If a command is to be initiated via the ladder logic program (AUTOMATIC RESTORATION) or via the operator interface (MAGELIS), a specific request pointer is placed into a FIFO buffer for immediate execution once the present command is executed by the MSTR block.

Network 1 senses that a pending command is to be executed in the FIFO block and the current command is terminated in the cyclic polling sequence for the MSTR block.

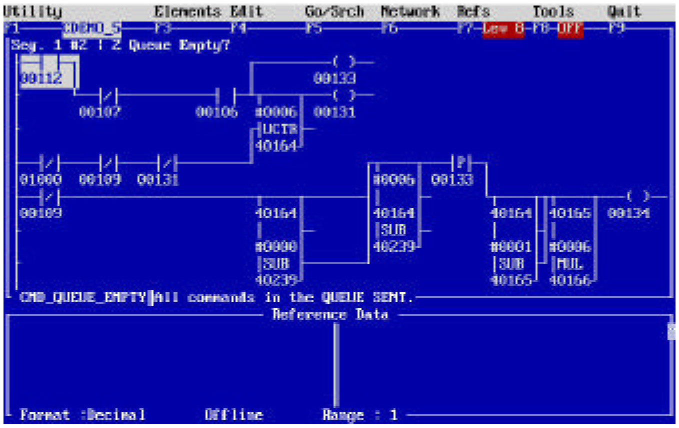


Figure 5 – Network 2 Segment 1 – Cyclic Poll Logic Block

NETWORK 2 – FIGURE 5:

There are 6 cyclic poll read instructions (illustrated by FIGURE 3 ABOVE). This network actuates the time to send each of the six instructions to the MSTR block to read the information from the TPU. If an instruction must be executed (such as a TPU TRIP, RESET TARGET LED Instruction) which is not part of the cyclic instruction sequence, this drum sequence is halted until all the commands in the buffer are executed by the MSTR instruction. When the FIFO is emptied, then cyclic polling resumes and this logic construct is energized.

NETWORK 3 – FIGURE 6:

As illustrated in figure 3, all instruction parameterization registers for the MSTR instruction is stored in the compact 6X memory registers. This network instruction (upon a change of the CTR instruction's cyclic poll) reads the block in 6X memory and places it in 4X memory. The contents are then moved into the MSTR block. Register 40100 = a 2 (read instruction) or a 1 (write instruction) based upon the data being read or written to the TPU. Registers 40102 through 40107 contain the MSTR parameters for the routing address (in this case the TPU is Address 1 Path 1) and such information as the number of registers to be transferred/read and the address in the TPU (to be read or written).

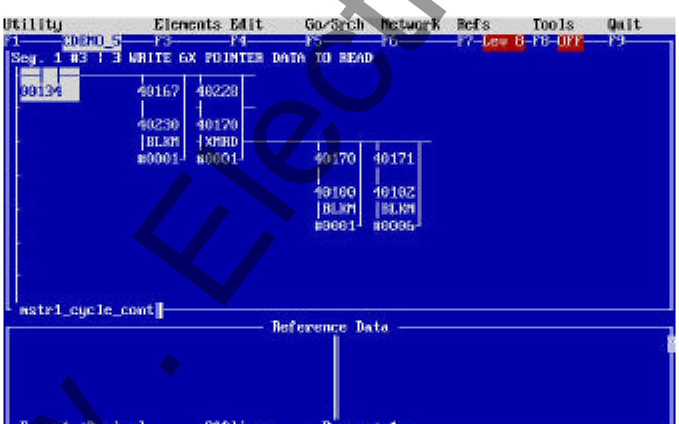


Figure 6 - Network 3 Figure 6 –6x Pointer Data

NETWORK 4 – FIGURE 7:

This is the MSTR send instruction which is parameterized for read and write commands. The upcounter in the logic counts the number of good network transactions (00104 energizes on a GOOD communication and 00103 energizes on a bad communication, 00102 is the instruction active indication). It should be noted that although 125 registers may be read/written at any one time, the program has been limited to Modbus Plus data accesses of 30 registers (to conserve PLC memory). The data is stored in 4X memory 40110 through 40149.

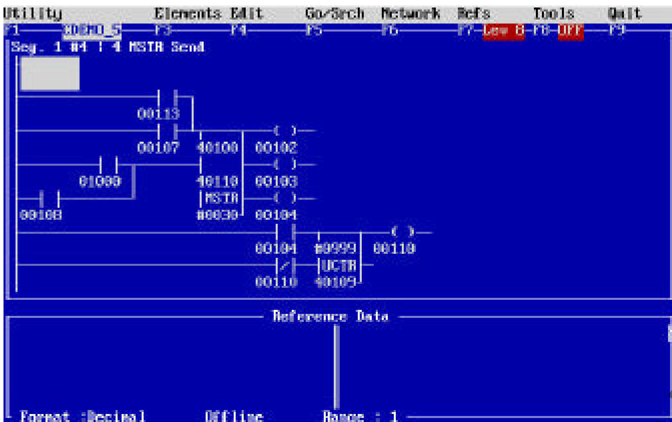


Figure 7 - Segment 1 Network 4 – Master Polling Block

NETWORK 5 – FIGURE 8:

The UCTR in this network counts the BAD Modbus Plus network transfers (an excellent indicator for network troubleshooting and program troubleshooting). This network also determines when an network access has finished executing. The TMR in this network construct places a dwell time of 200 mS between each Modbus Plus network transaction.

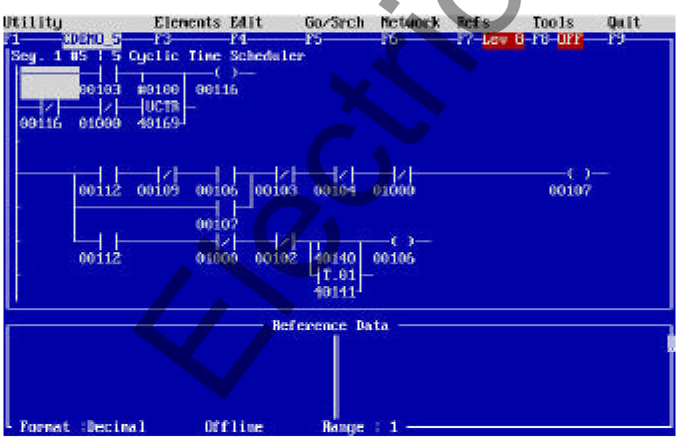


Figure 8 - Ladder Logic Dwell Timer and Bad Transaction Network

NETWORK 6 – FIGURE 9:

This network determines if the FIFO has an entry. If the FIFO is empty Coil 00112 is energized. If the

FIFO is full Coil 00111 is energized. The FIFO can contain 19 pending commands for execution in the TPU. As illustrated in Figure 3, the commands are numbered 1 through 10. 40142 and 40143 respectively are the data pointed to in the queue and the FIFO queue pointer.

Coil 00109 is the indicator for the program that the FIFO has an entry and the master should be halted.

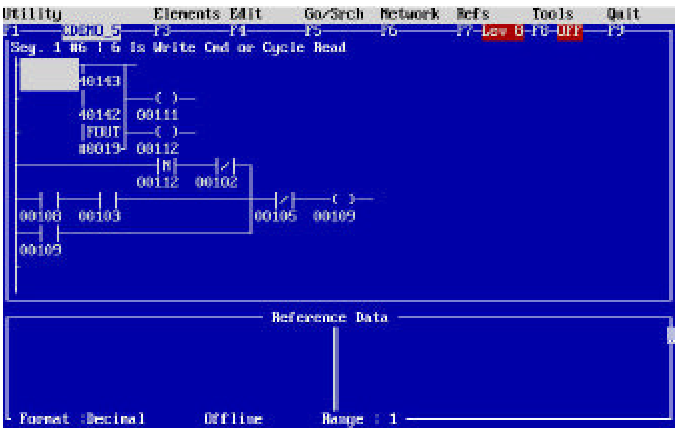


Figure 9 – FIFO MSTR Halt Logic

NETWORK 7- FIGURE 10:

This network delays the initiation of the MSTR block send instructions when a FIFO command has been sensed in the buffer. The pointer to this command is in the 40163 pointer register. The delay is 100 mS. As can be seen in the ladder logic, the FOUT (FIFO out stack) is popped and the parameters corresponding to the FIFO pointer are transferred to the logic to transfer the 6X MSTR pointers from the 6X registers pointed (in FILE 1 6X memory) to the MSTR 40100 through 40109 register block.

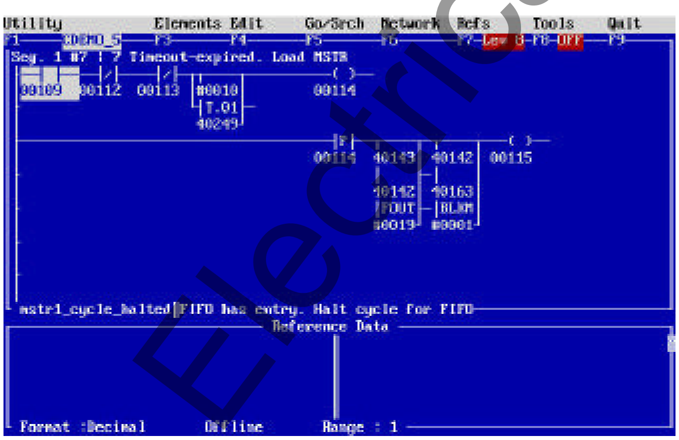


Figure 10 – Network 7 Segment 1 – FIFO Scheduling Halt Logic

NETWORK 8 – FIGURE 11:

This is the 6X transfer instructions as illustrated in FIGURE 11

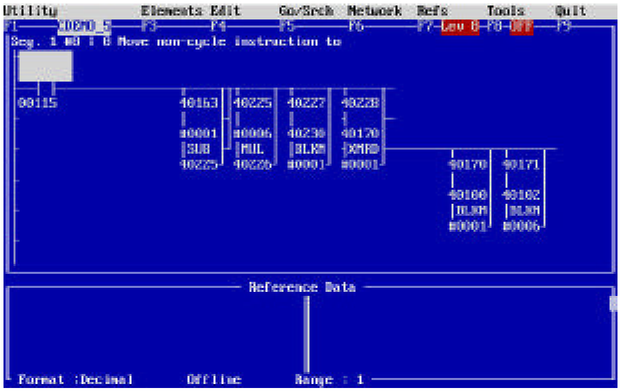


Figure 11 – Ladder Logic 6x Transfer Instructions.

The method to transfer the logic and the file layout in 6X memory is as such:

File 1 – Registers 60000 to 604999 contain the MSTR 1 instruction command parameterization. The commands are in a block of 6 register formats.

60XXXX+0 = MSTR Command 1 = read 2 = write
60XXXX+1 = Number of registers read/written
60XXXX+2 = Address in TPU to be read/written
60XXXX+3 = Node Route 1 Address
60XXXX+4 = Node Route 2 Address
60XXXX+5 = Node Route 3 Address

Routing address Paths 5 and 4 are a value of 0.

FILE 2 6X registers are the data written to the TPU. Each command is in a block of 10. If the command in the corresponding block is 1 (read) then the file 2 block of registers is a don't care. If the command in the corresponding block is 2, then the data in FILE 2 with the corresponding configuration data in FILE 1 is sent to the MSTR configuration registers. FILE 2 data is transferred to 40110 (MSTR data field). The grouping of the data in file 2 is always:

61XXXX + 0 to 61XXX + 9 = Block data.

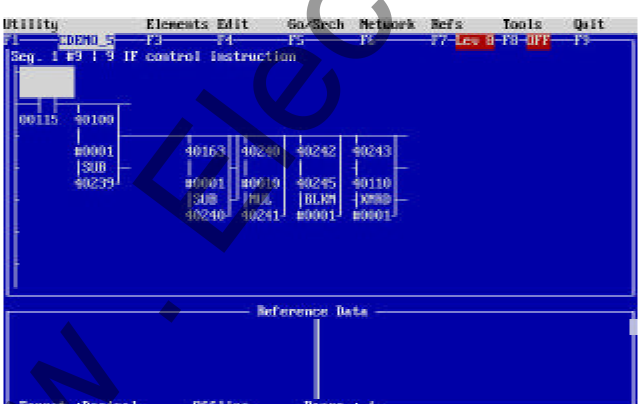


Figure 12 – Transfer Logic For Control Instruction Parameterization

NETWORK 9- FIGURE 12:

As explained above, if the pointer in the FIFO vectors to a control instruction, then the FILE 2 data corresponding to that pointer must be transferred to 40110 in the MSTR instruction. This network does just that. The 6X address is calculated by the MULT instruction and then passed as an argument to the XMRD instruction. Upon execution of the XMRD instruction, the data is transferred into the MSTR block. If the FIFO instruction in the buffer is a WRITE instruction, then the pointer placed in the buffer is multiplied by 10 to get the block of data to transfer to the MSTR data register buffer 40110. The multiplier for instructions are:

PTR = 1 Block 0 Add 0 to pointer for FILE 2 60000
PTR = 2 Block 1 Add 10 o pointer for FILE 2 60010
PTR = 3 Block 2 Add 20 pointer for FILE 2 60020
PTR = 4 Block 3 Add 30 to pointer for FILE 2 60030
PTR = 5 Block 4 Add 40 to pointer for FILE 2 60040
PTR = 6 Block 5 Add 50 to pointer for FILE 2 60050
PTR = 7 Block 6 Add 60 to pointer for FILE 2 60060
PTR = 8 Block 7 Add 70 to pointer for FILE 2 60070

… and so on, and so on ….

NOTE – The data is transferred if the function code in 40100 = 2 (write), else the data is meaningless.

NETWORK 10 – FIGURE 13:

This network is only a latch in which once the buffer FIFO contains an entry.

Figure 13 – Segment 1 Network 10 – Start Polling Sequence

NETWORK 11 to 19 – FIGURES 14 to 22:

The commands 1 through 5 are the cyclic ladder logic commands which read

- REG 40129 TPU STATUS
- REG 40385 – 40401 CURRENT ANGLE DATA
- REG 40513 – 40518 VOLTAGE ANGLE DATA
- REG 40528 – 40533 POWER VALUES

- REG 41153 –41156 ALARM STATUS VALUES

The Command 6 is the trigger command for all the WRITE commands which include

- TRIP COMMAND – Write 41411 – 41415 and WRITE 41410
- RESET TARGETS – Write 41411 – 41415 and WRITE 41410
- RESET ALARMS – Write 41411 – 41415 and WRITE 41410
- RESET LATCHED DATA – Write 41411 – 41415 and WRITE 41410

The commands as illustrated in FIGURE 3 of this document.

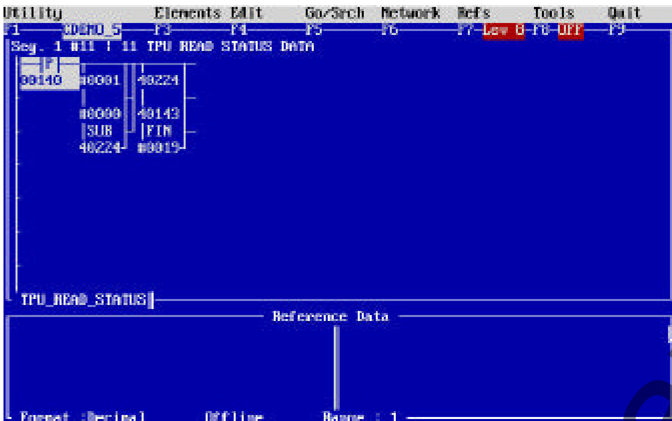


Figure 14 – TPU Read Status FIFO Pointer Load Logic

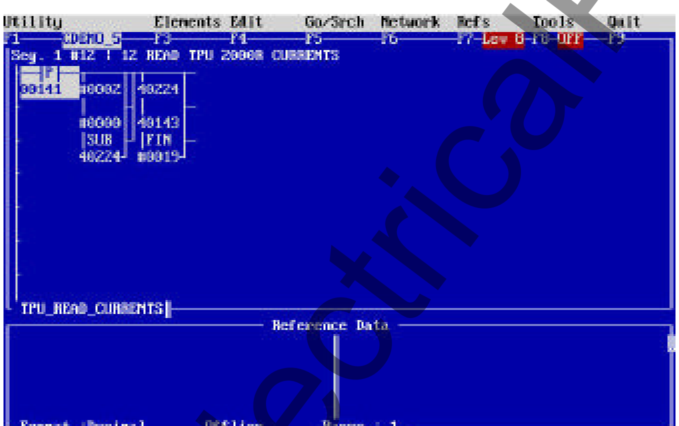


Figure 15 – TPU Read Currents FIFO Pointer Load Logic

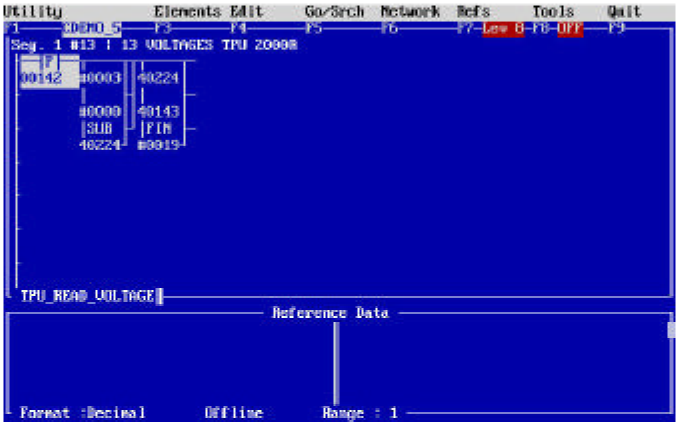


Figure 16 – TPU Read Voltage FIFO Pointer Load Logic

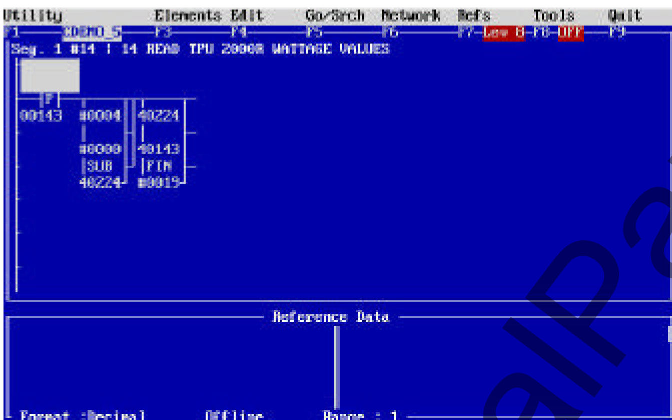


Figure 17 – TPU Read Wattage FIFO Pointer Load Logic

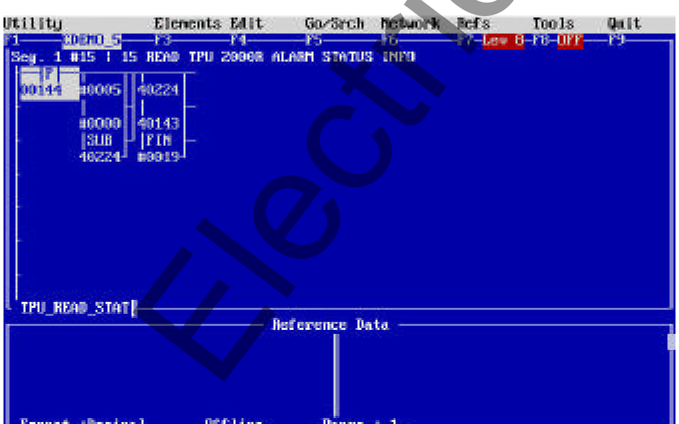


Figure 18 – TPU Read Alarm Status FIFO Pointer Load Logic

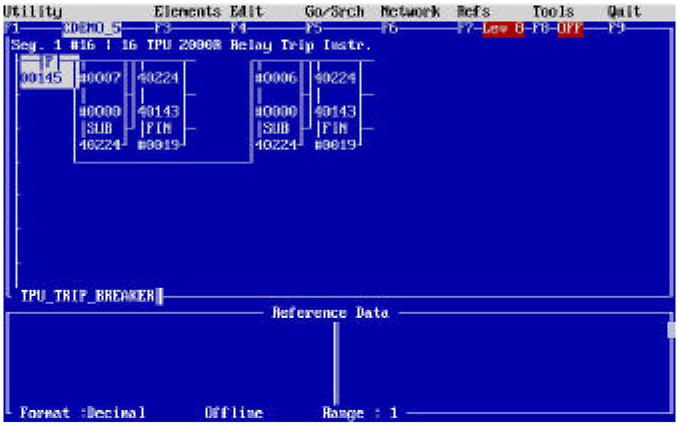


Figure 19 – TPU Trip Breaker FIFO Pointers Load Logic

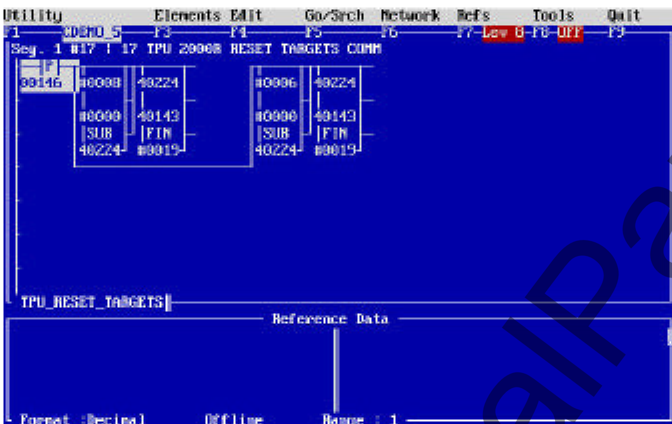


Figure 20 – Reset Targets FIFO Pointers Load Logic

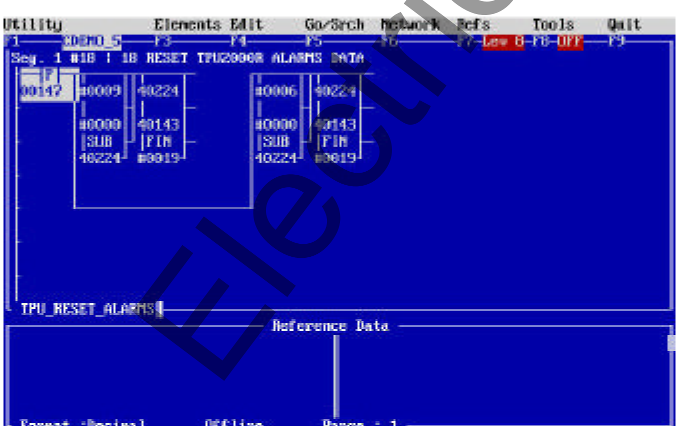


Figure 21 – Reset Alarms FIFO Pointers Load Logic

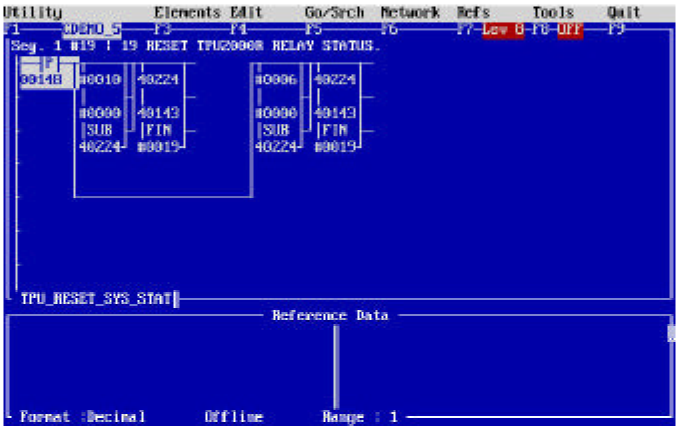


Figure 22 – Reset Latched Points FIFO Pointers Load Logic

NETWORKS 20 to 24 – FIGURES 23 to 27

Once the data is read from the relay using the cyclic reads (pointers 1 through 5) or via the FIFO commands, the data read must be transferred from the MSTR read buffer to a general buffer for retrieval from the PLC. The PLC then serves as a data concentrator. The TPU data registers are contained in 41700 through 41726. The ladder logic networks in this construct are triggered when the MSTR instruction has obtained the information from the relay. The instruction then transfers the appropriate quantity to the appropriate registers (as illustrated by FIGURE 3 above).

The SUB instruction determines the MSTR command executed and the BLKM command instruction block moves the information from the MSTR data buffer (40110) to the appropriate register location from 41700 through 41726 resident in the PLC.

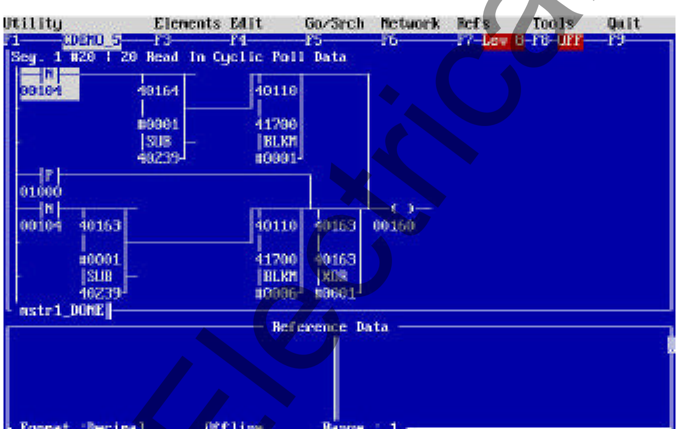


Figure 23 – Read In Reply To Status Data Request And Store In PLC Registers

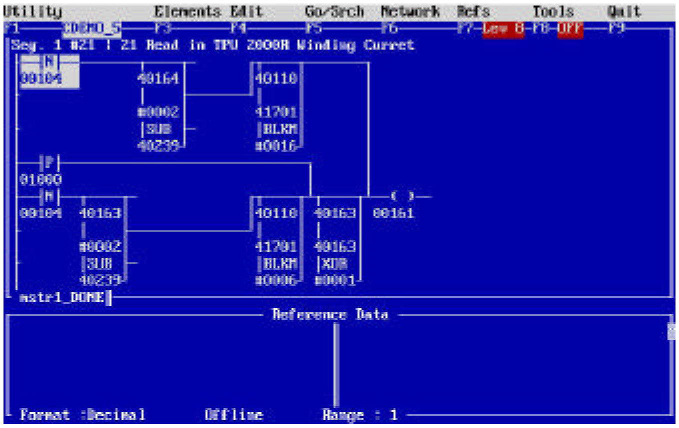


Figure 24 – Read In Reply To Phase Current Data Request And Store In PLC Registers

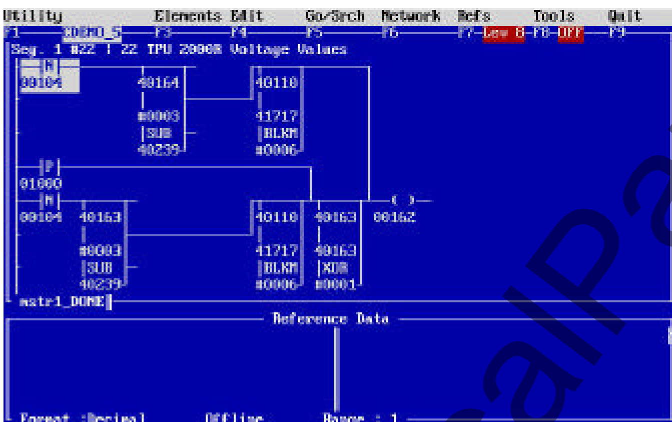


Figure 25 – Read In Reply To Phase Voltage Data Request And Store In PLC Registers

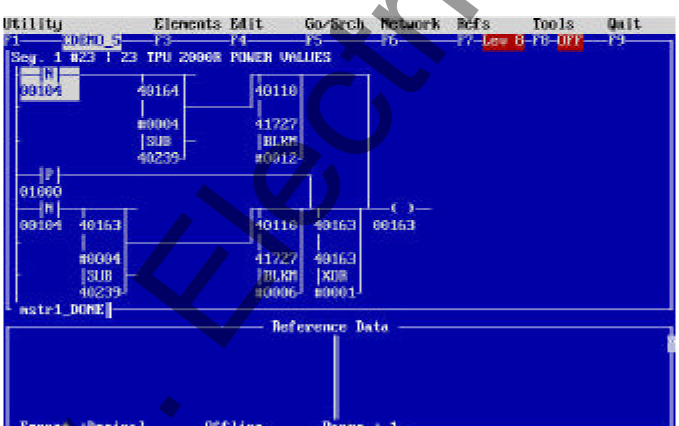


Figure 26 – Read In Reply To Wattage Data Request And Store In PLC Registers

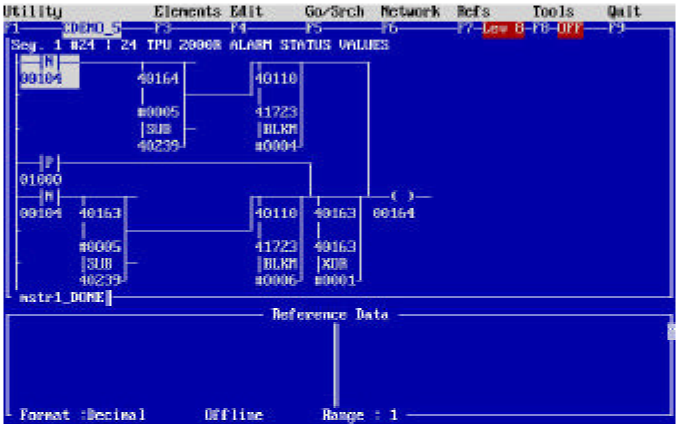


Figure 27 – Read In Reply To Alarm Data Request And Store In PLC Registers

Segment 2 – Data Gathering From The PCD 2000 And DPU 2000R.

The DPU 2000R and the PCD 2000 are both located some distance from the PLC. Attachment to these relays is accomplished using simple SCATTER RADIO MODEMS. The radio modems are able to communicate over a distance of 15 miles and retrieve information from them via a 10 bit protocol. The SCATTER RADIO MODEMS have the advantage that no special licensing is required for connection to the devices.

The ladder logic required for data retrieval is located in the ladder logic segment 2 and is based upon the XMIT instruction which turns the PLC's RS 232 port into a MODBUS master. This enables the PLC to query attached devices and poll for data. In this way the PLC is able to determine the loading of the remote feeder and breaker status of the feeders being monitored and protected by the DPU 2000R and the PCD 2000.

The PLC queries both the PCD and DPU via the command parameterization of the XMIT instruction. The topology of the installation is illustrated in FIGURE 28 as follows in this discussion.

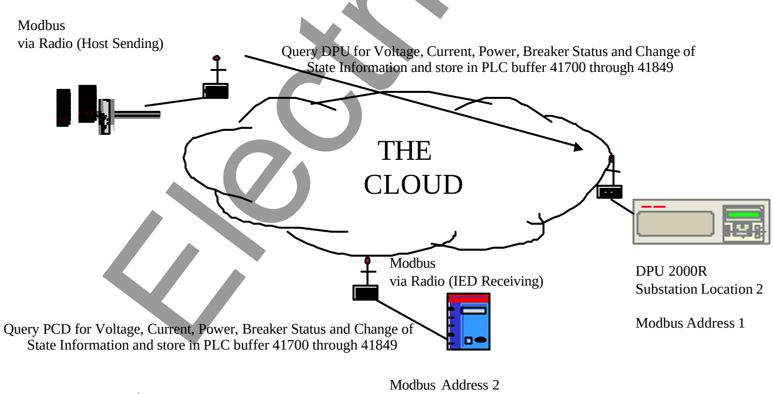


Figure 28 – Network Topology For Nodes Remote To The PLC And Polled Via MODBUS Command Responses

The ladder logic shall be reviewed for the method to complete the data exchange between the PLC and the IED's

SEGMENT 3 NETWORK 1:

The FIFO used to gather the information from the IED's is reset whenever data is transferred into the FIFO for polled queue. This is a standard instruction construct which is similar to that used for the MSTR instruction.

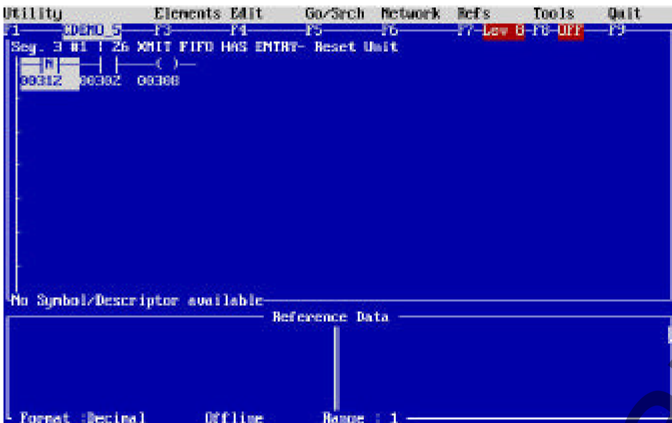


Figure 29 – Segment 3 Network 1 – FIFO Entry For XMIT Instruction Notification

SEGMENT 3 NETWORK 2:

The network is a cyclical poll which pages the 10 instructions to the XMIT to gather data from the two IED's (PCD 2000 and the DPU 2000R). The UCTR instruction increments between 1 and 10. The Pointer is then multiplied by 10 (stored in 40333) which serves as the pointer to the 6X command buffer which parameterizes the XMIT instruction to retrieve the IED's data.

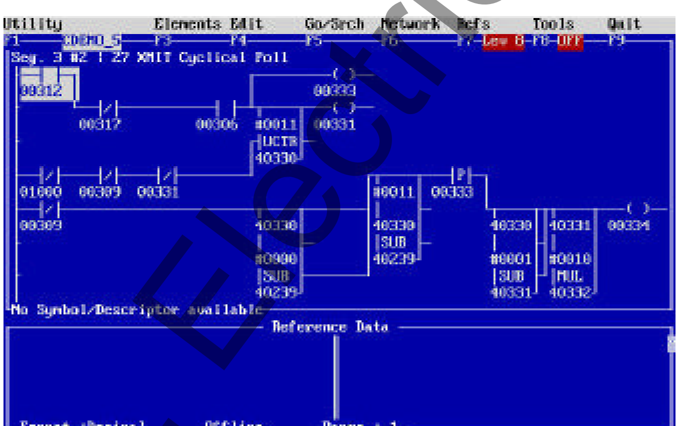


Figure 30 – Segment 3 Network 2 –Cyclic Poll Pointer Setup Logic

SEGMENT 3 NETWORK 3:

This network seems to be rather complex, but it really is not. The XMIT block needs two types of parameterization 1). parameterization of the block for delay parameters, timeout parameters and definition of pointers for the MODBUS data gathering which is in registers 40308 through 40315. 2). the parameterization of the Modbus commands which are pointed to by registers 40309 and 40310 (which is the address of the 5 registers for command parameterizations) and the length of the parameter block (which is 5 and its structure depends upon the instruction, please reference the MODICON XMIT BLOCK documentation for a more complete discussion of the data).

The XMIT instructions are stored in FILE 2 and FILE 1 of 6X memory beginning at addresses 600500 and 610500 respectively. File 1 parameterizes the XMIT block parameters and FILE 2 parameterizes the particular MODBUS request. The mathematics of this block calculates the data in the following way:

FILE 1 POINTER (Reg 40334) = Pointer (Reg 40331 * 10) + 500 offset.
FILE 2 POINTER (REG 30337) = Pointer (Reg 40331 * 30) + 500 offset.

The File 1 pointer is passed to the XMRD block which transfers 7 registers to the XMIT parameterization space 40308 to 40315. The File 2 Pointer is passed to the XMRD block which sends 30 registers from 6X memory to 40355 to 40384. FIGURE XX illustrates the parameterization which must occur in order to allow the XMIT block to gather the data successfully. As is illustrated in FIGURE 31 the PLC stores the data in PLC register 41700 through 41849. The PLC program has been optimized for data storage and grouped register usage. This program may be expanded to perform other functions.

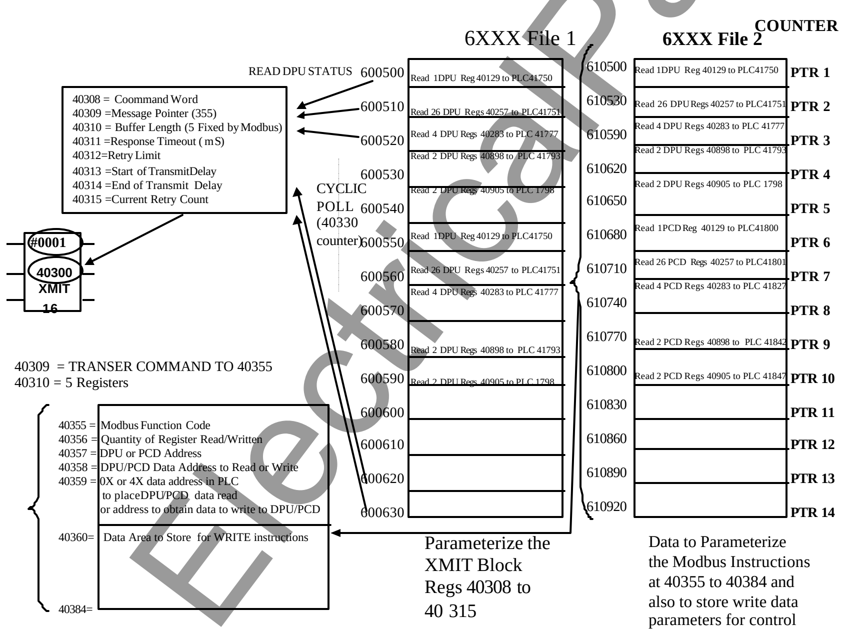


Figure 31 – XMIT Parameterization Philosophy For Data Control.

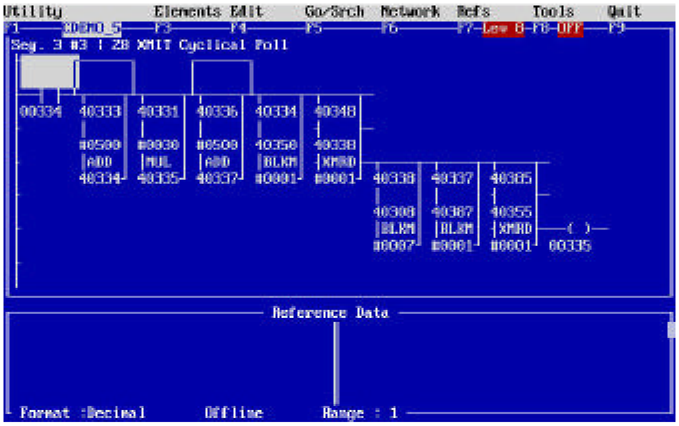


Figure 32 – Segment 3 Network 3 – 6x Pointer Computation Logic For Loading XMIT Instruction

SEGMENT 3 NETWORK 4:

Network 4 is the base XMIT instruction. As illustrated above, the data table is filled when the drum timer of Segment 3 network 2 counts between 1 and 10 (which are the cyclic read instructions. An optional UPCTR counts the good transmissions (which is good for keeping track of communication percentage failures) over the radio network. This is used with the next nework in the segment to keep track of the type of failures occurring during troubleshooting of the program. The Ladder Logic follows and is illustrated in FIGURE 33.

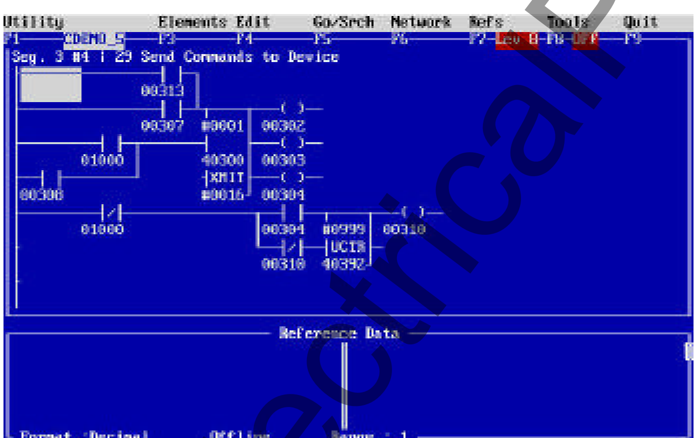


Figure 33 – Segment 3 Network 4– XMIT Instruction And Good Transaction Count

SEGMENT 3 NETWORK 5:

This network latches the last XMIT error in register 40300. It is reset by pulsing coil 00888. This is additional logic added for the ease of troubleshooting the program. The logic is illustrated in FIGURE 34.

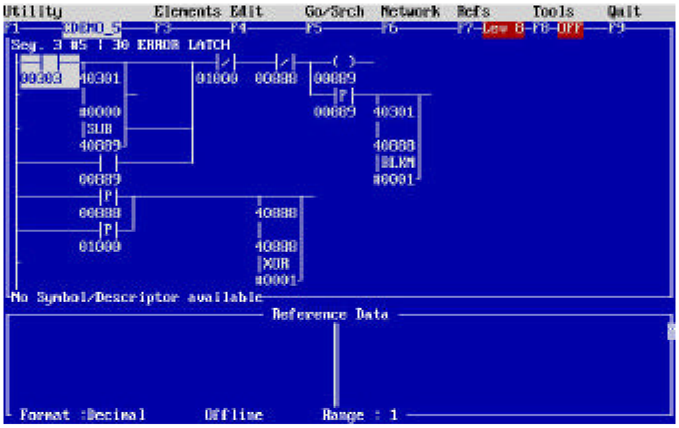


Figure 34 – Segment 3 Network 5 – XMIT Error Bad Transaction Counter And XMIT Error Latch Clear Logic

SEGMENT 3 NETWORK 6:

The output of the XMIT block signals when an error occurs on a transmission. The UCTR instruction in this logic construct counts the number of transmission errors experienced by the XMIT BLOCK. This is instructional in determining the amount of errors occurring on the network.

The second logic construct (with the TMR) places a dwell time of 100 mS between MODBUS transmissions. The coil 00307 pulses the XMIT instruction when the FIFO buffer is empty and the XMIT instruction is able to transmit and instruction as part of its cyclical poll structure. The ladder logic is illustrated in FIGURE 35.

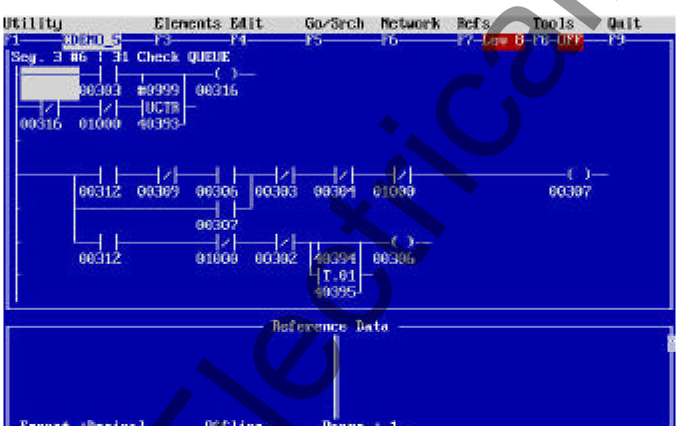


Figure 35 – Segment 3 Network 6- XMIT Dwell Timer And Bad Transmission Counter Logic.

SEGMENT 3 NETWORK 7:

As with the Modbus Plus MSTR Logic, another FIFO has been constructed in which a manual control (or automated control instruction initiated by the ladder logic may be performed). Note the FIFO may contain up to 19 control instructions which may be queue'd for processing. Additionally, note the trigger construct for pending FIFO commands to be sent to the XMIT block (coil 00309) interrupting the cyclical poll. Using this philosophy ensures that control commands and operator initiated commands are immediately scheduled for operation by the XMIT block.

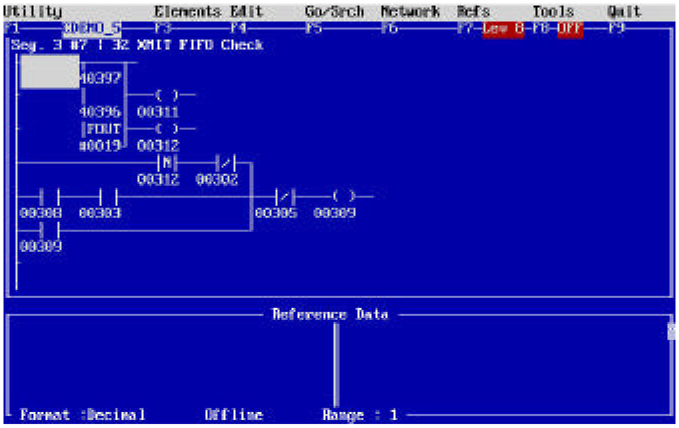


Figure 36 – Segment 3 Network 7 – FIFO Empty/Full Notification Logic

SEGMENT 3 NETWORK 8:

If the FIFO has data, this logic construct interrupts the polling of the XMIT instructions and immediately parameterizes the block with the parameters pointed to within the FIFO queue. A dwell time of 100 mS creates a pause for the XMIT instruction to terminate. The logic in SEGMENT 3 NETWORK 9 creates the pointers for obtaining the 6X register data for passing to the appropriate 4X XMIT parameterization registers. FIGURE 37 illustrates the logic to accomplish this task.

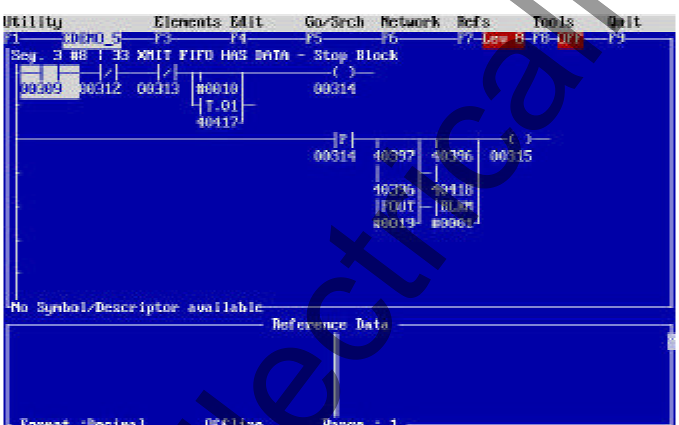


Figure 37 – Segment 3 Network 8 – Cyclic Poll Cessation Logic

SEGMENT 3 NETWORK 9:

As illustrated, this is essentially a copy of Segment 3 Network 3 logic. It is copied here in order to keep the same philosophy for instruction parameter loading whether it is from a cyclic poll request or a FIFO task interruption. The logic is shown in FIGURE 38.

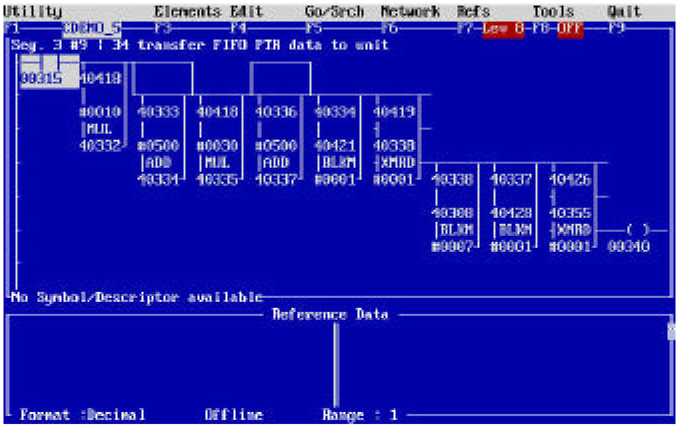


Figure 38 – Segment 3 Network 9 – XMIT Pointer Computation Logic

SEGMENT 3 NETWORK10:

This network determines that the FIFO is empty and cyclic polling using the logic in Segment 3 Network 2 resumes. The Logic is illustrated in FIGURE 39.

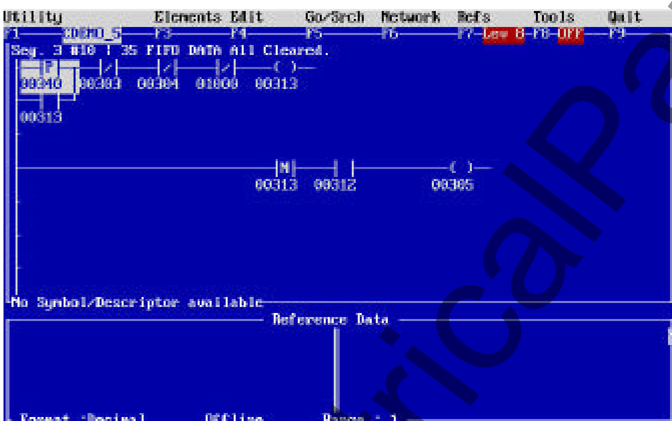


Figure 39 – Segment 3 Network 10 – Resume Cyclic XMIT Poll Logic

SEGMENT 3 NETWORKS 11 THROUGH 20.

As illustrated in FIGURE 31 – The first 10 instructions in the buffer are to read data from the Modbus registers in the DPU or PCD and stack them in the PLC for later data processing to perform the load shedding/load restoration (line sectionalizing). Figure 31 illustrates in the FILE 2 explanation, that whenever the POINTER from number 1 to 10 is placed in the FIFO (if an interrupt command is required) or if the UCTR instruction in Segment 3 Network2 changed within it's sequence, the data is transferred as illustrated. Networks 11 through 20 perform the following:

- If the SUB and FIN combination logic construct is energized by the preceding contact, the FIFO is filled with the queue'd command for processing by the XMIT block. This stops the cyclic polling (if occurring) and schedules the command for execution in the next 100 mS/
- When the XMIT block executes and receives the successful transmission, the data received is transferred from the XMIT block buffer and transmitted to the PLC data storage buffer

41750 through 41850 . The data may be viewed by an operator interface such as MAGELIS, stored for further processing (decisions on whether to perform the line sectionalizing), or stored in other memory areas for data buffering/concentration for transfer to a host device at a later date.

The contact to the left of the rail (1st logic construct) are those which schedule the XMIT transmission.

FIGURES 40 through 49 illustrate the data FIFO instruction queue initiation and the data storage mechanism.

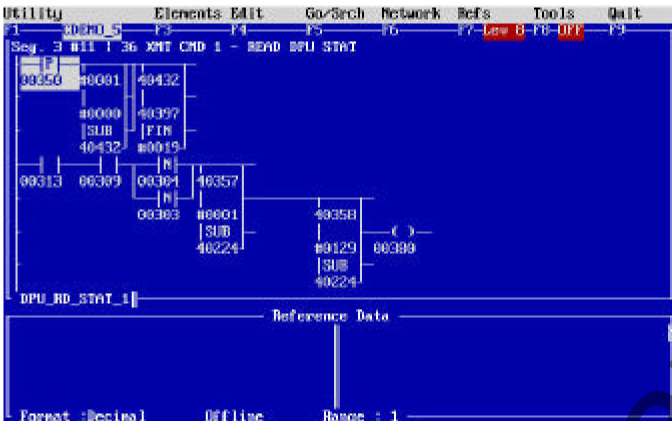


Figure 40 – Segment 3 Network 11 – Read DPU Status Information Or FIFO Instruction Load.

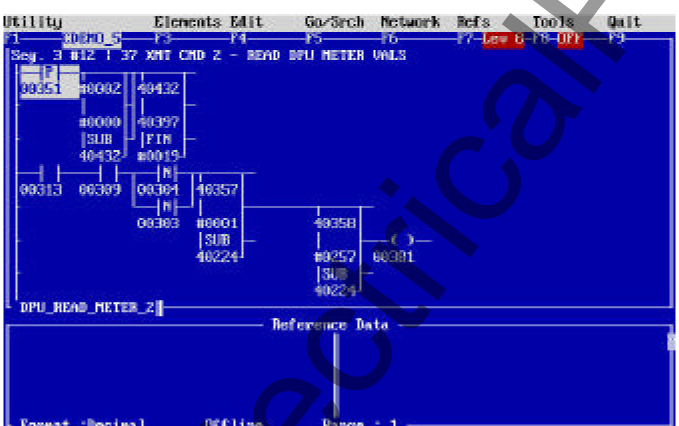


Figure 41 – Segment 3 Network 12 – Read DPU Metering Information Or FIFO Instruction Load.

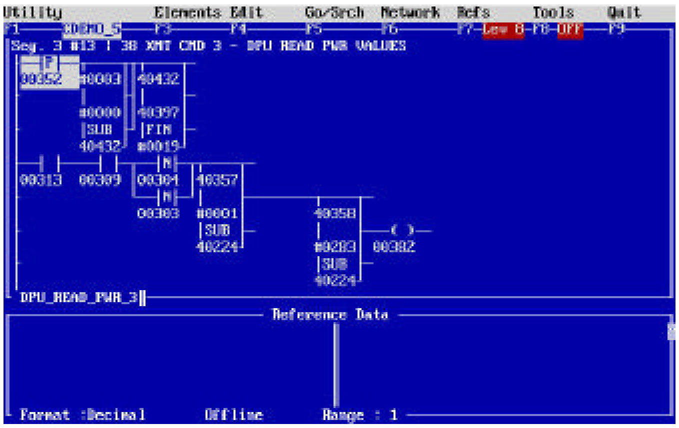


Figure 42 – Segment 3 Network 13 – Read DPU Power Information Or FIFO Instruction Load.

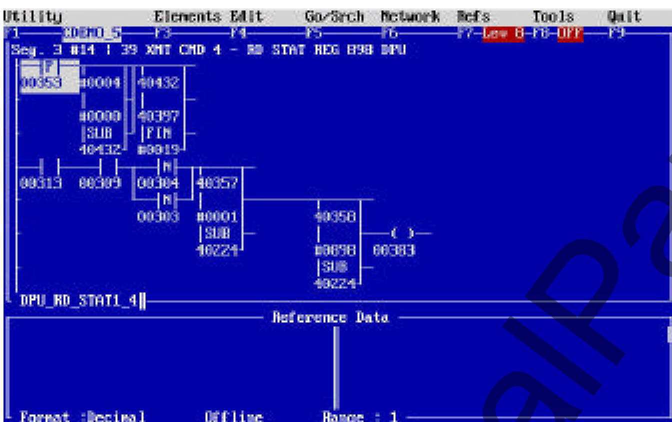


Figure 43 – Segment 3 Network 14 – Read DPU Breaker Status Information Or FIFO Instruction Load.

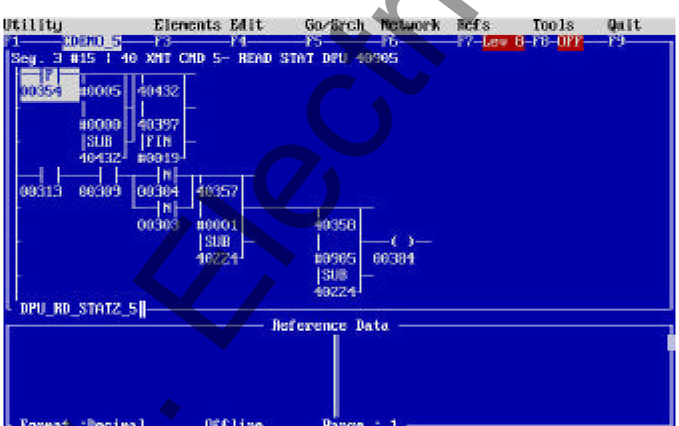


Figure 44 – Segment 3 Network 15 – Read DPU Status Information Or FIFO Instruction Load.

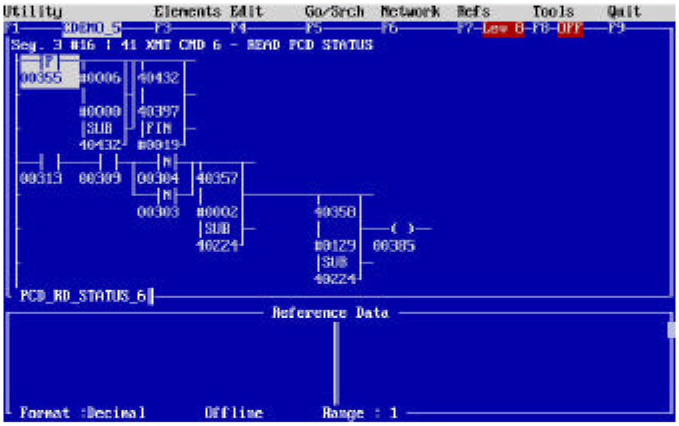


Figure 45 – Segment 3 Network 16 – Read PCD Status Information Or FIFO Instruction Load.

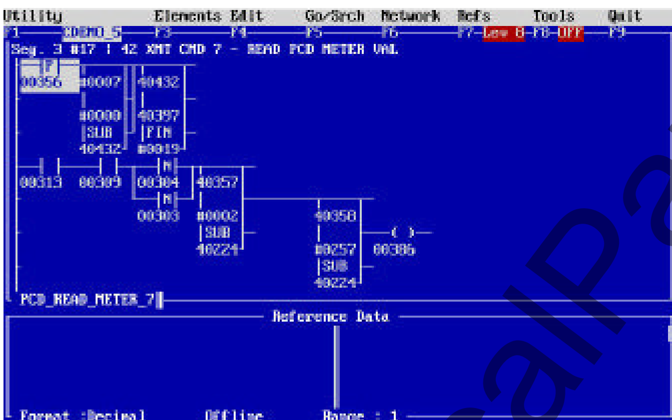


Figure 46 – Segment 3 Network 11 – Read PCD Metering Information Or FIFO Instruction Load.

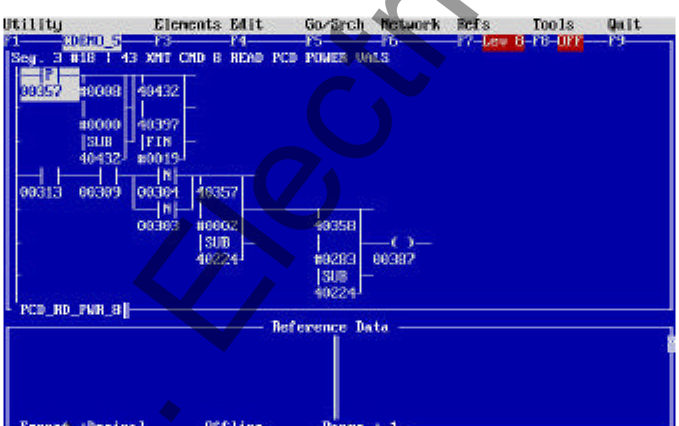


Figure 47 – Segment 3 Network 18 – Read PCD Power Information Or FIFO Instruction Load.

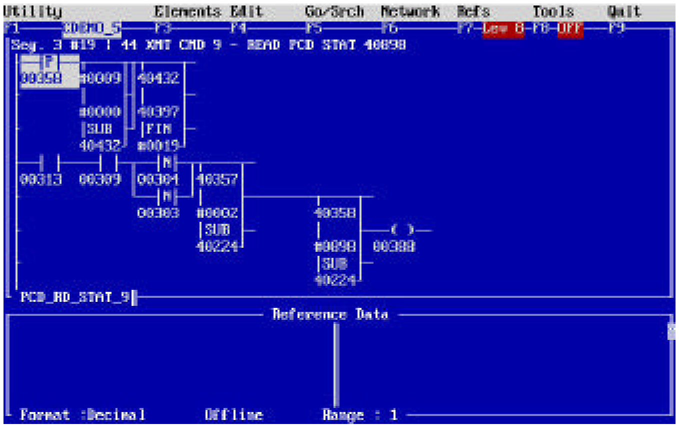


Figure 48 – Segment 3 Network 19 – Read PCD Breaker Status Information Or FIFO Instruction Load.

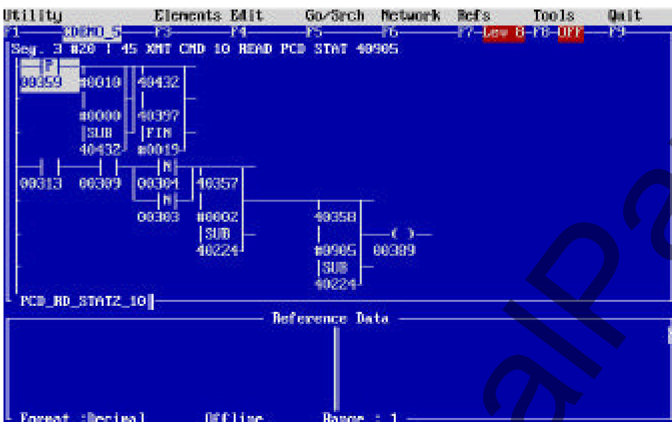


Figure 49 – Segment 3 Network 20 – Read PCD Status Information Or FIFO Instruction Load.

SEGMENT 3 NETWORKS 21 THROUGH 30:

As the previous networks 11 through 20 only data access instructions were programmed in the device. Networks 21 through 30 perform control instructions. As illustrated in the DPU 2000R Automation Manual and the PCD 2000 Modbus Protocol documents, the procedure for performing control is to write a group of registers parameterizing the control (usually writing 5 consecutive registers) and then writing one single register with the execute command (usually 1) and send it to the relay within 100 seconds of the parameterization commands. As illustrated in FIGURES 51 through 60, the procedure to do this is shown in FIGURE 50.

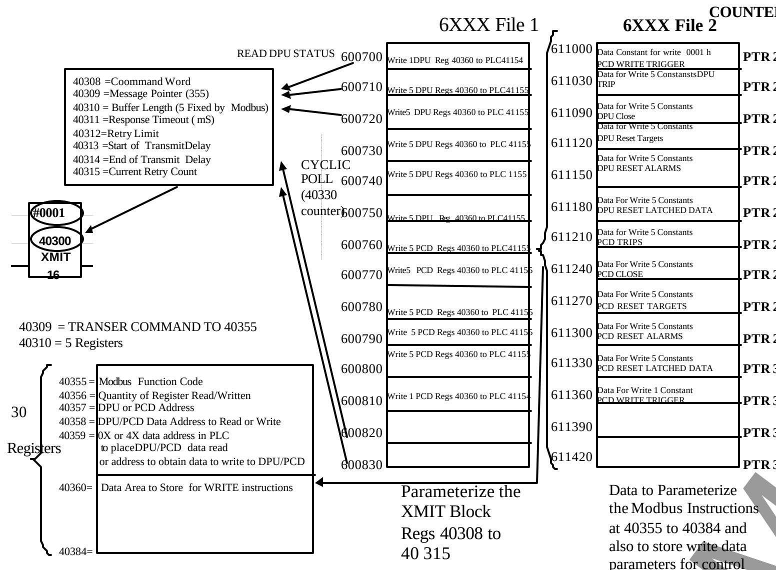


Figure 50 – Write Ladder Logic Methodology.

In order to do a write instruction for the DPU, the FIFO must be preloaded with an instruction between 21 through 25 and then the FIFO must be loaded with the trigger instruction (a Write of 1 to register 41154) which is pointer 20. In order to do a write instruction for the DPU, the FIFO must be preloaded with an instruction between 26 through 30 and then the FIFO must be loaded with the trigger instruction (a Write of 1 to register 41154) which is pointer 31. FIGURES 51 – 60 illustrates the ladder logic to perform the base relay control and read data structures by which this entire program is predicated upon.

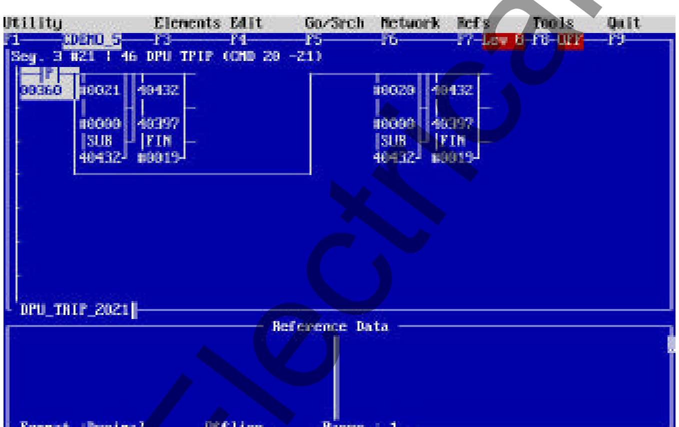


Figure 51 – Segment 3 Network 21- Place DPU Trip Command In FIFO

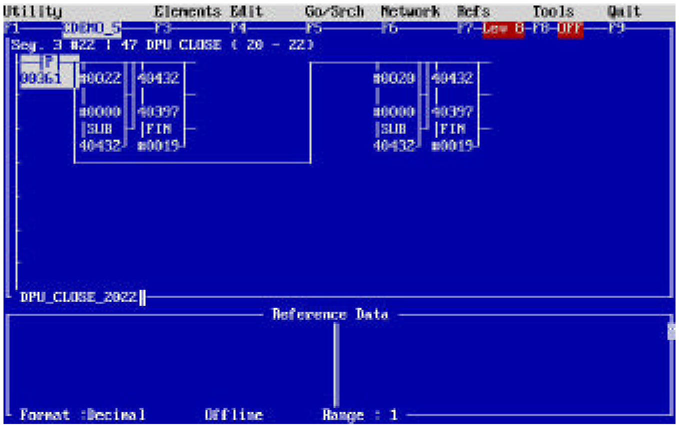


Figure 52 – Segment 3 Network 22 - Place DPU Close Command In FIFO

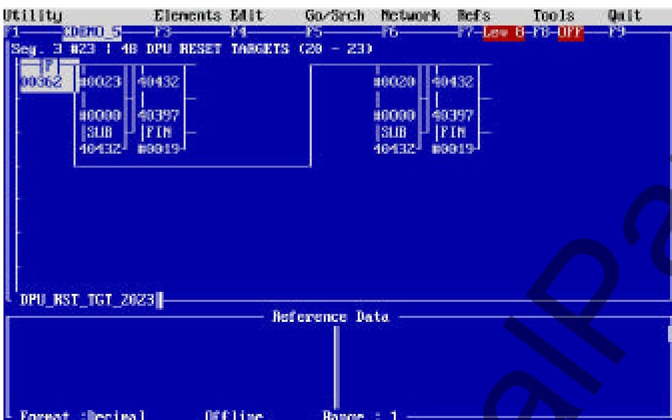


Figure 53 – Segment 3 Network 23 - Place DPU Reset Targets Command In FIFO

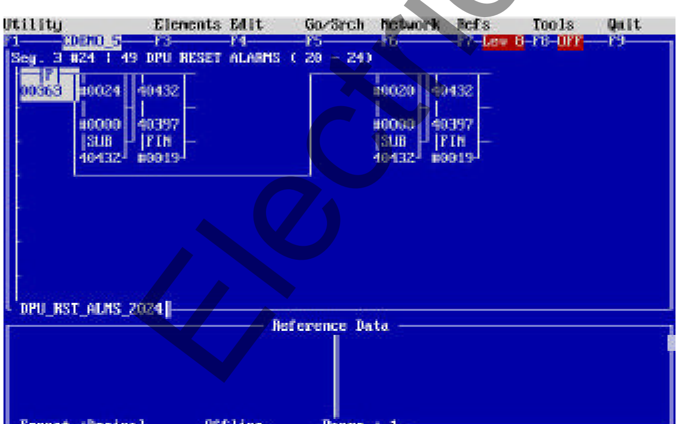


Figure 54 – Segment 3 Network 24 - Place DPU Reset Alarms Command In FIFO

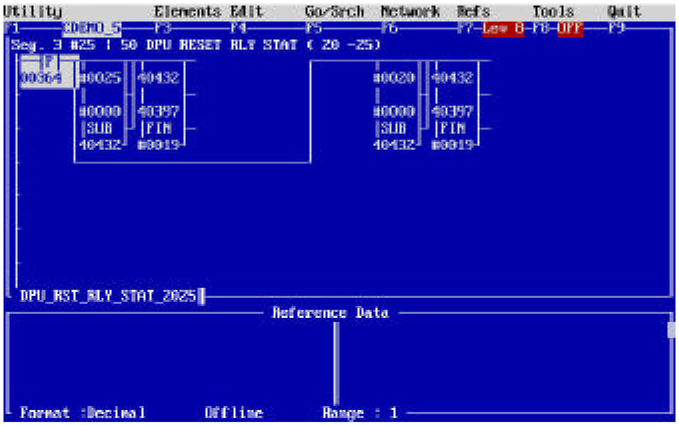


Figure 55 – Segment 3 Network 25 - Place DPU Reset Status Command In FIFO

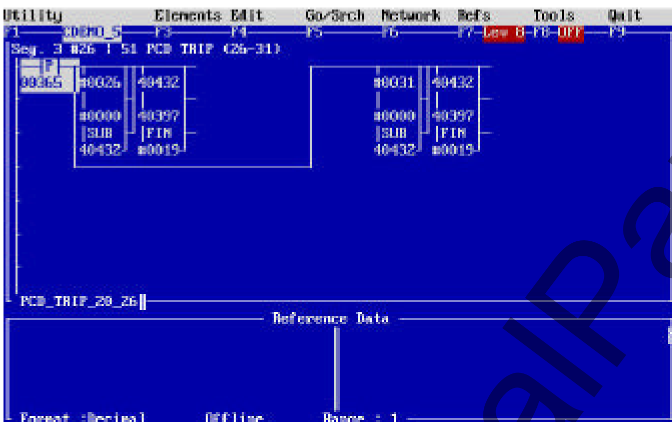


Figure 56 – Segment 3 Network 26 - Place PCD Breaker Trip Command In FIFO

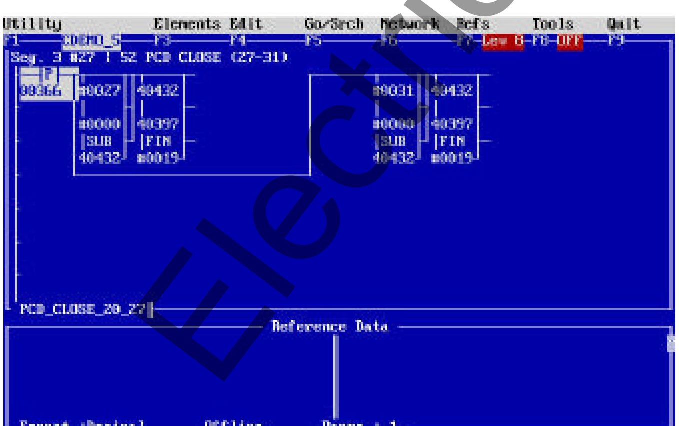


Figure 57 – Segment 3 Network 27- Place PCD Close Command In FIFO

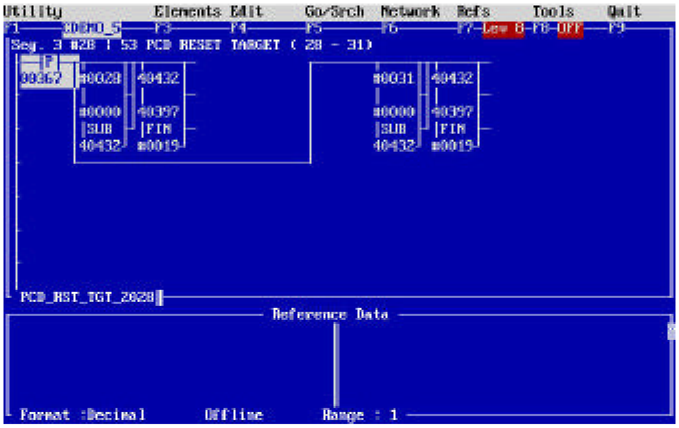


Figure 58 – Segment 3 Network 28 - Place PCD Reset Targets Command In FIFO

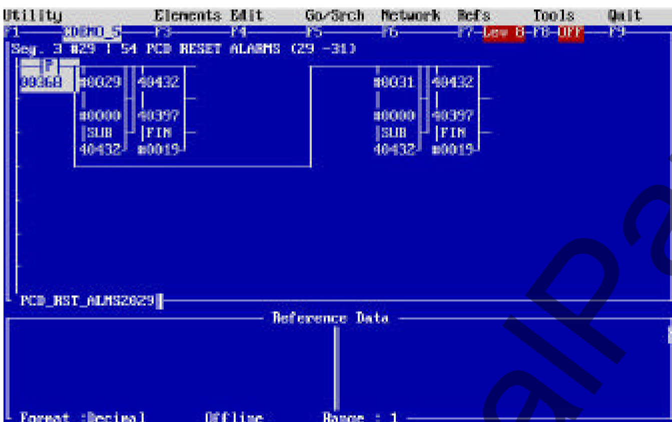


Figure 59 – Segment 3 Network 29 - Place PCD Reset Alarms Command In FIFO

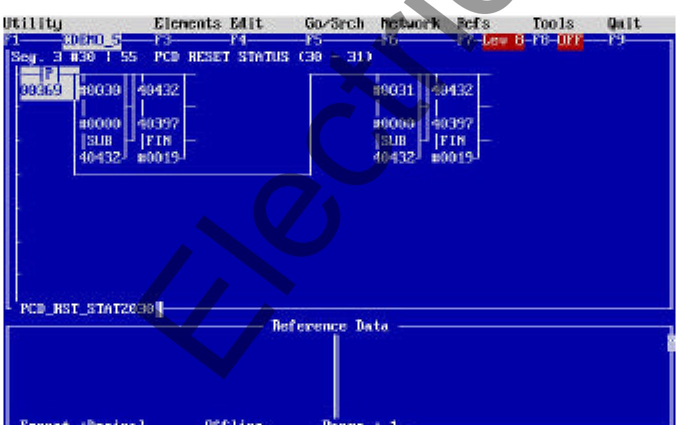


Figure 60 – Segment 3 Network 30 - Place PCD Reset Status Command In FIFO

Segment 4 Network 1: Operator Interface Control Screens

In this example, the PLC program exists as a central data concentrator. In this case, the program was developed to have a register be set in order to trigger the control instructions via an operator interface. The MMI control screens are described herein on a network to network basis.

Register 40051 is the input bit control register (the MAGELIS sets the bit momentarily) and the ladder logic fills the FIFO with the appropriate command for toggling the graphic. Bit 16 or 14 in the word is set to indicate that automatic or manual control for restoration is followed. Automatic restoration allows the logic for restoration to be enacted. If the Manual control is selected, the operator via the operator screen controls restoration.

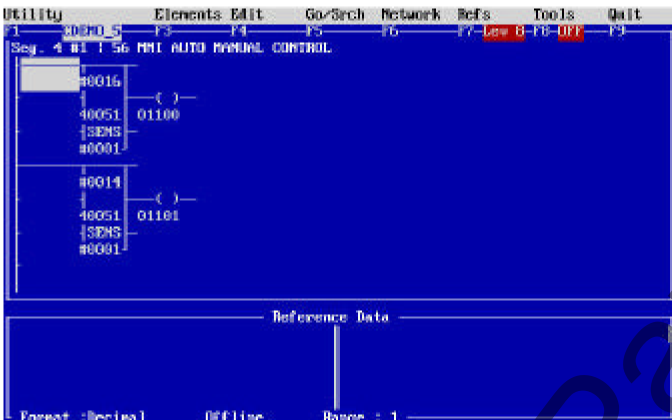


Figure 61 – Segment 4 Network 1- MAGELIS F1 And F2 Function Key Auto Manual Control Logic

SEGMENT 4 NETWORK 2:

This is more MMI control logic required for the MAGELIS operator interface. NOTE the pushbuttons for trip, close and reset operations only operate when the system is in AUTOMATIC mode.

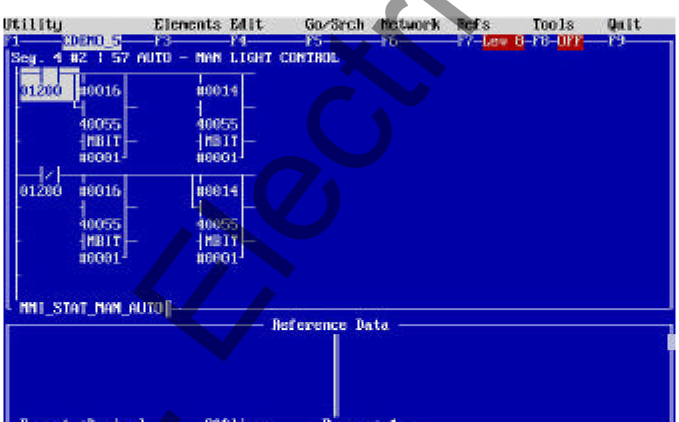


Figure 62 – Segment 4 Network 2 - MAGELIS F1 And F2 Function Key Led Control Auto Manual Control Logic

SEGMENT 4 NETWORK 3:

This network upon the MMI control screen being issues a system reset, the pending control operations, buffers and latched commands are reset to an initial state.

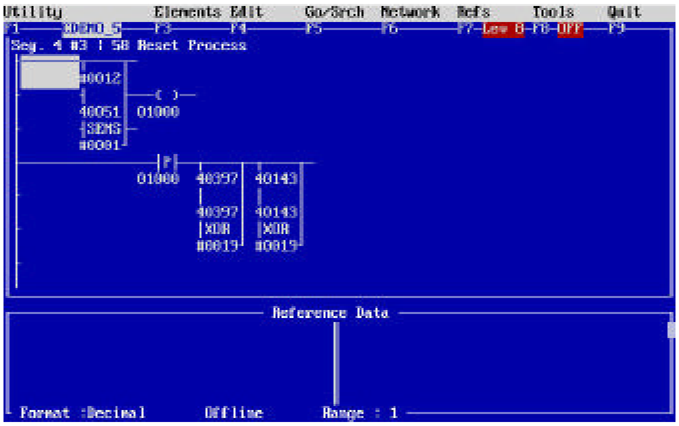


Figure 63 – segment 4 network 3 - system reset logic

SEGMENT 4 NETWORK 4:

If the control key for a MANUAL TRIP of the DPU is depressed on the MMI, this logic construct loads the FIFO with the XMIT pointer commands 21,20 to perform a breaker trip operation on the DPU 2000R. Bit 15 of register 40051 is set by the MMI to trigger this instruction (SENS). The ladder logic is illustrated in FIGURE 64.

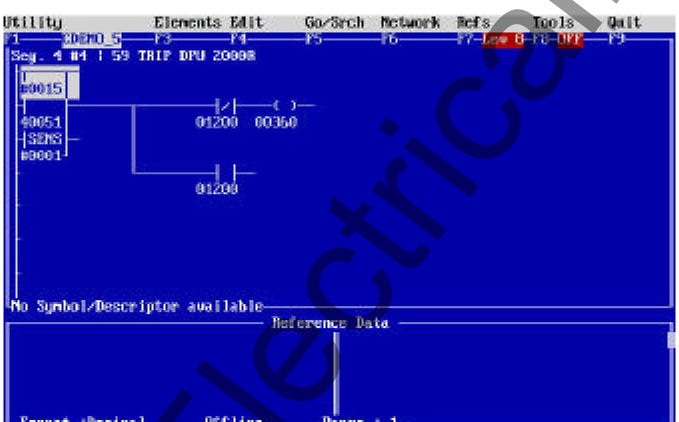


Figure 64 – Segment 4 Network 4 - MAGELIS Pushbutton Manual Trip Logic

SEGMENT 4 NETWORK 5:

If the control key for a MANUAL CLOSE of the DPU is depressed on the MMI, this logic construct loads the FIFO with the XMIT pointer commands 22,20 to perform a breaker trip operation on the DPU 2000R. Bit 13 of register 40051 is set by the MMI to trigger this instruction (SENS). The ladder Logic is illustrated in FIGURE 65.

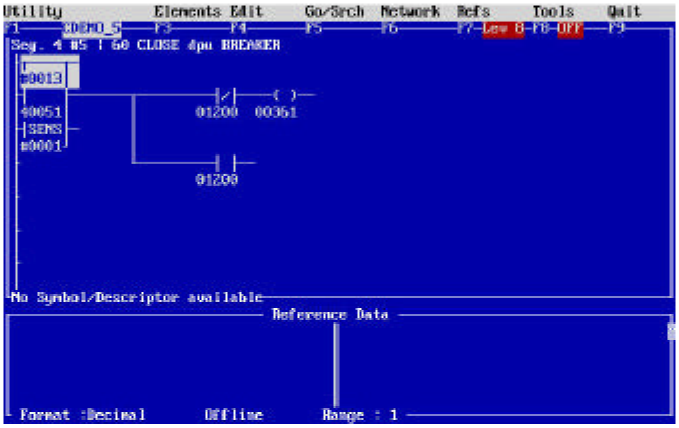


Figure 65 – Segment 4 Network 5 - MAGELIS Manual Close Pushbutton Logic

SEGMENT 4 NETWORK 6:

If the control key for a MANUAL TRIP of the PCD is depressed on the MMI, this logic construct loads the FIFO with the XMIT pointer commands 26,31 to perform a breaker trip operation on the DPU 2000R. Bit 11 of register 40051 is set by the MMI to trigger this instruction (SENS).

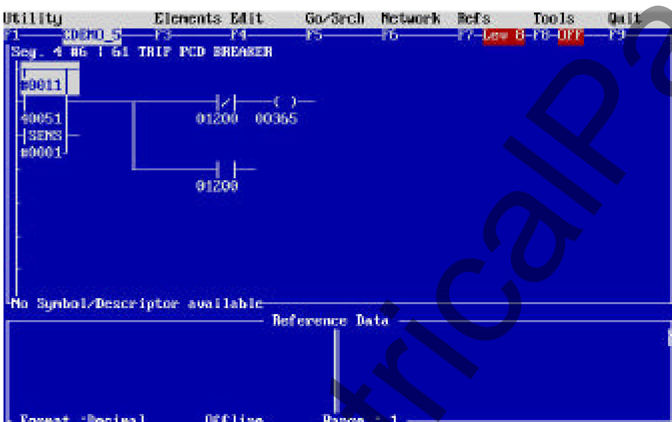


Figure 66 – Segment 4 Network 6 - MAGELIS PCD Manual Trip Pushbutton Logic

SEGMENT 4 NETWORK 7:

If the control key for a MANUAL CLOSE of the PCD is depressed on the MMI, this logic construct loads the FIFO with the XMIT pointer commands 27,31 to perform a breaker trip operation on the DPU 2000R. Bit 5 of register 40051 is set by the MMI to trigger this instruction (SENS).

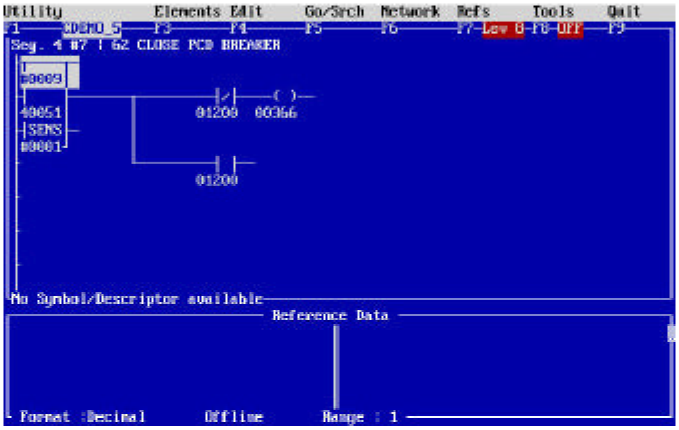


Figure 67 – Segment 4 Network 7- MAGELIS PCD Manual Close Pushbutton Logic

SEGMENT 4 NETWORK 8:

If the control key for a MANUAL TRIP of the TPU is depressed on the MMI, this logic construct loads the FIFO with the MSTR pointer commands 7,6 to perform a breaker trip operation on the DPU 2000R. Bit 11 of register 40051 is set by the MMI to trigger this instruction (SENS). Note this operation sends a Modbus Plus comand to the TPU.

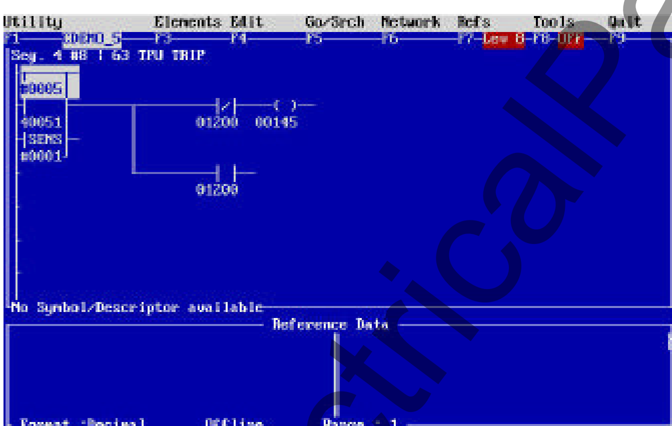


Figure 68 – Segment 4 Network 8 - MAGELIS TPU Manual Trip Pushbutton Logic

SEGMENT 4 NETWORK 9:

The MAGELIS operator interface does not display Floating Point Values. All the mathematics in this program is performed using floating point math. In order to display the information on the MMI display (MAGELIS), it must be converted from floating point to integer for display, Kwatts for phases A, B, C, and loading of the DPU prior to the TPU trip. These values are calculated using floating point math instructions.

Figure 69 – Segment 4 Network 9 - MAGELIS Watt Hour Display In Integer Units

SEGMENT 4 NETWORK 10,11,12:

The operator is also able to reset the target information via the MAGELIS MMI. If the unit is in manual mode, the operator may depress the function key to reset the targets on the TPU (BIT 8 REGISTER 40051) DPU (BIT 10 REGISTER 40051) AND PCD (BIT 6 REGISTER 40051) . The ladder logic for these constructs are listed in FIGURES 70 through 72.

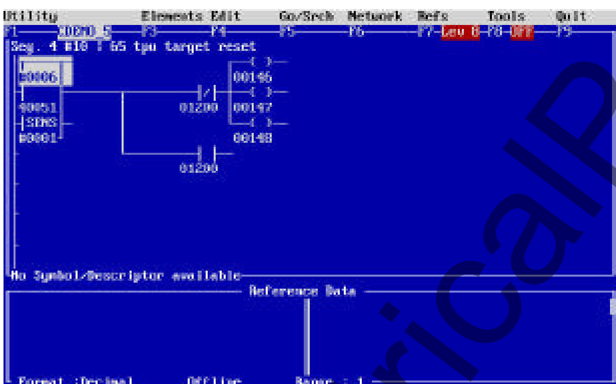


Figure 70 – Segment 4 Network 10 - MAGELIS TPU Manual Trip Pushbutton Logic

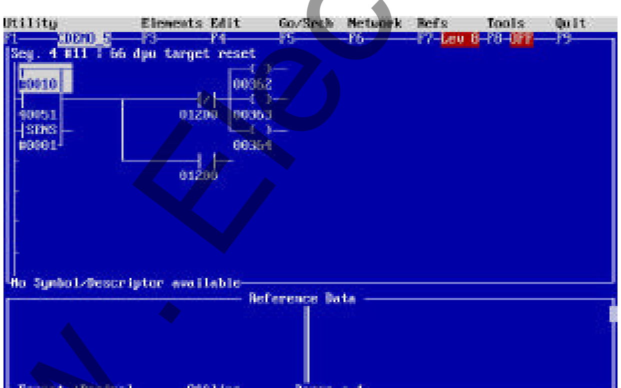


Figure 71 – Segment 4 Network 11 - MEGELIS TPU Target Reset Pushbutton Logic

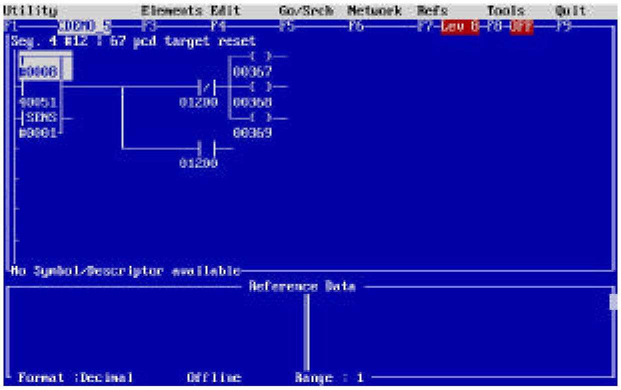


Figure 72 – Segment 4 Network 12 - MAGELIS PCD Manual Target Reset Pushbutton Logic

SEGMENT 4 NETWORKS 13 AND 14:

The MMI displays data via bit data which toggles the graphics. The logic constructs in Networks 13 and 14 illustrate the logic to indicate on the display the breaker status of the TPU as well as the AUTO/MANUAL PLC program control. The logic is illustrated in FIGURES 73 and 74.

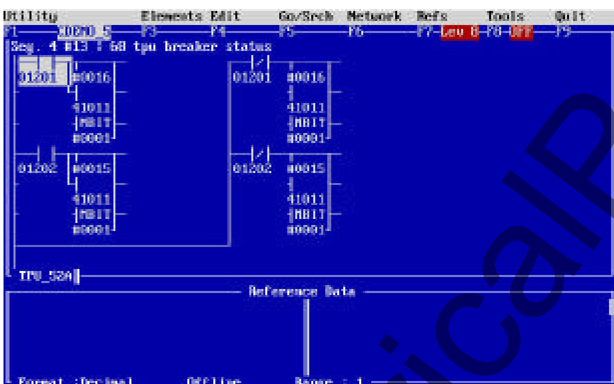


Figure 73 – Segment 4 Network 13 - MAGELIS PCD Breaker Status Logic To Manipulate The Screen Graphics.

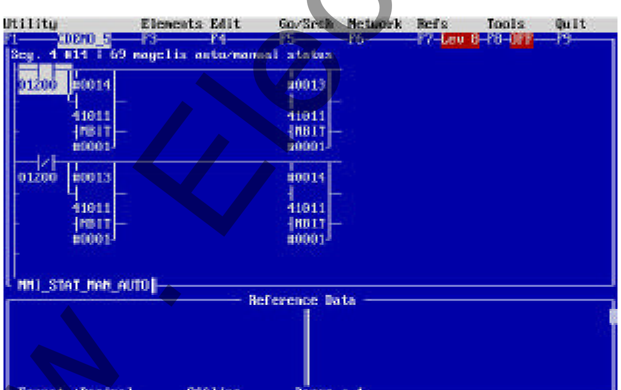


Figure 74 – Segment 4 Network 12 - MAGELIS Auto/Manual Screen Status Icon Logic

Segments 5 and 6 performs the logic which initiates the procedure upon a TPU monitored/protected feeder trip. The explanation of the logic follows.

SEGMENT 5 NETWORK 1:

Since 52a and 52b are not direct points within the TPU, the PLC program reads WINDING currents for phase A (41702) , B (41704) , and C (41706) and if the currents are less than 2 amperes, the TPU denotes the relay as tripped and generated 52a (01201) and 52b (01202) internal status coils. Note that labels have been affixed to each of the registers contacts and coils. The logic is illustrated in FIGURE 75.

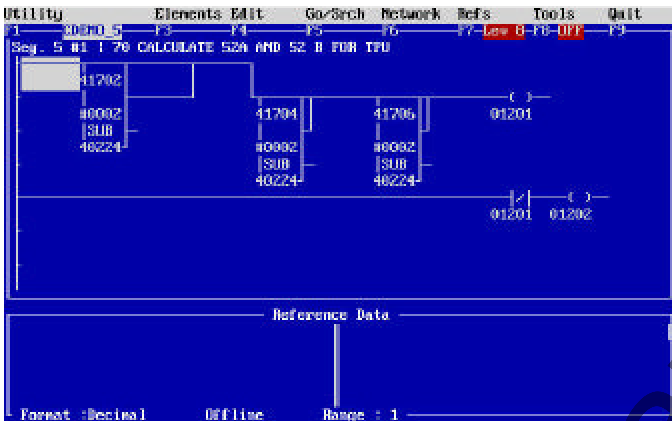


Figure 75 – Segment 5 Network 1 - Calculate 52A And 52B On TPU Since Contacts Are Not Mapped.

SEGMENT 5 NETWORKS 2, 3, 4, And 5:

Segments 2, 3, 4, and 5 (for the sake of this program since it is a demonstration and illustration of the power of the relay and PLC's capabilties), convert the KW of Phase A, B, and C, which was read from the TPU and supplied to the feeder (controlled by the PCD). The Compact 984 PLC only performs integer math on numbers from 0000 to 9999. The PLC calls a subroutine to convert the number from a 32 bit number integer (which is obtained via the MSTR block and stored in registers 41728 and 41729 [Phase A integer Units], 41730 and 41731 [Phase B Integer Units] and, 41732 and 41733 [Phase C Integer Units]) and converted into floating point numbers which enable easy mathematical conversion feeder load control. The floating point converted numbers are calculated in the subroutine segment (segment 7) and are labeled as JSR 2 and JSR 1. The floating point numbers are located in registers 41739 and 41740 [Phase A Floating Point quantity]], 41741 and 41742 [Phase B Floating Point quantitiy], and 41743 and 41744 [Phase C Floating Point quantity]. Since this program was tested on a simulator, the values were made to be positive quantities for the sake of illustration in a demonstration environment.

Network 5 adds each of the quantities and stores it for comparison to a predefined feeder supply value which is compared when the line sectionalizing occurs.

Figure 76 – Segment 5 Network 2 - Calculate KW For Phase A

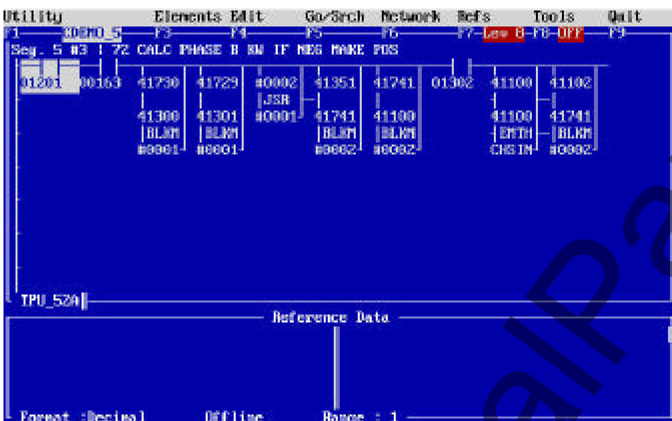


Figure 77 – Segment 5 Network 3 - Calculate KW For Phase B

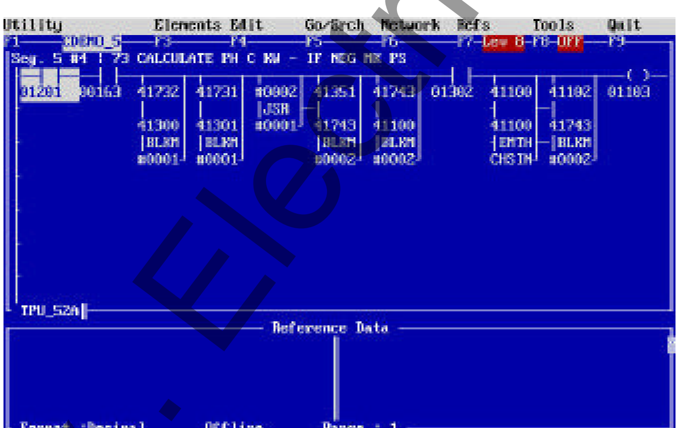


Figure 78 – Segment 5 Network 4 - Calculate KW For Phase C

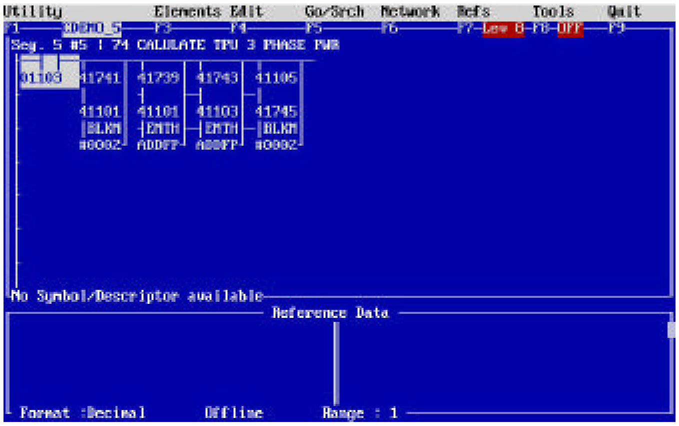


Figure 79 – Segment 5 Network 5 - Calculate KW For All Three Phases

SEGMENT 5 NETWORK 6:

Since the TPU does not have 52a and 52 b reported for a trip condition (since it is not wired into the simulator in this example), if the current of each of the phases is a value less than 2 amps, the TPU is determined to be tripped. This instruction construct sends a trip command (via the commands 6 and 7) via the FIFO for the MSTR block. This trips the TPU to ensure the state of the unit. The network logic is illustrated as per FIGURE 3 of this note. This network performs the action in MANUAL mode.

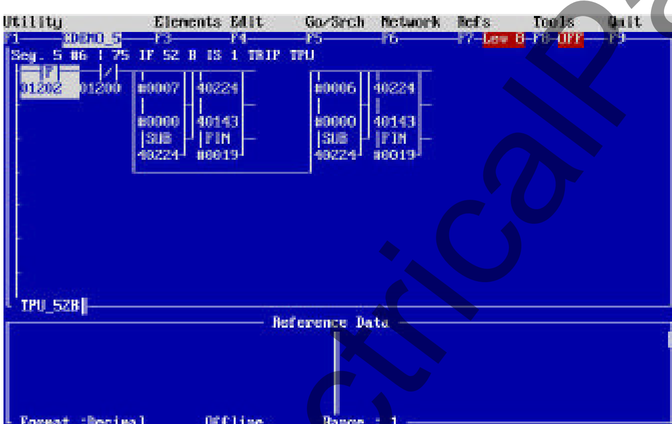


Figure 80 – Segment 5 Network 6 - Trip TPU If Readings Are Less Than 1 A Per Phase (Since 52A And B Contacts Are Not Wired Into Demo Case)

SEGMENT 6 NETWORK 1:

This network (although out of place in the scheme of things), takes a pushbutton input from the Magelis MMI and places the PLC program in the MANUAL or AUTOMATIC restoration status. If coil 01200 is a energized, the program is in AUTO mode. If the coil 01200 is de-energized, the program is in manual mode. The logic is illustrated in FIGURE 81.

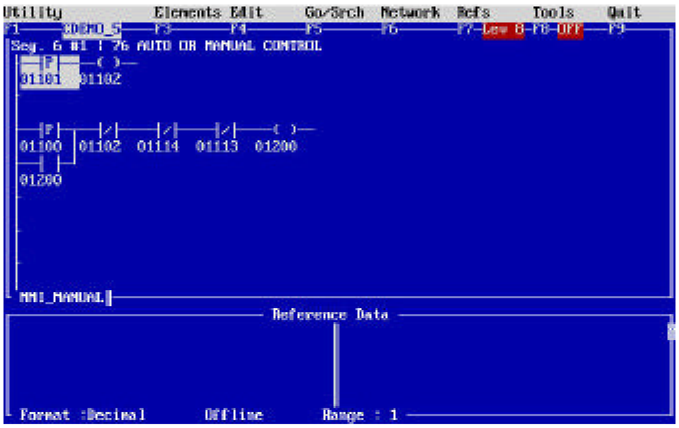


Figure 81 – Segment 6 Network 1- Upon MMI MAGELIS Action, Place The Program In Manual Or Automatic Restoration Mode.

SEGMENT 6 NETWORK 2:

This network checks the TPU TARGET status which was stored in register 41725. If a target is on the front panel interface, an indication is given by coil 001150 which is used in this program. The logic is illustrated in FIGURE 82.

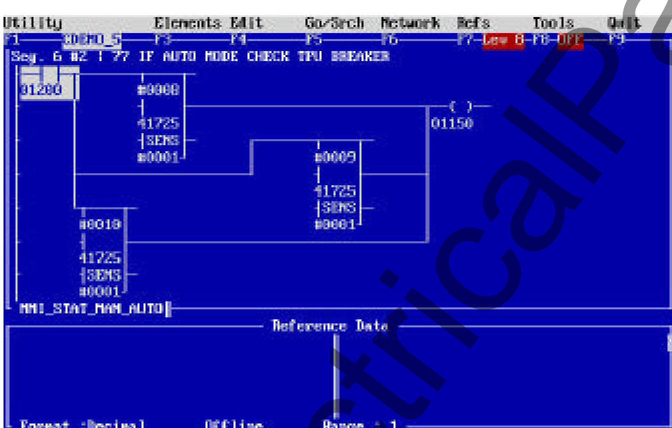


Figure 82 – Segment 5 Network 6 - Trip TPU If Readings Are Less Than 1 A Per Phase (Since 52A And B Contacts Are Not Wired Into Demo Case)

SEGMENT 6 NETWORK 3:

Since the TPU does not have 52a and 52 b reported for a trip condition (since it is not wired into the simulator in this example), if the current of each of the phases is a value less than 2 amps, the TPU is determined to be tripped. This instruction construct sends a trip command (via the commands 6 and 7) via the FIFO for the MSTR block. This trips the TPU to ensure the state of the unit. The network logic is illustrated in section 82. This network performs the action in AUTOMATIC mode. The coil 01151 carries this action to the next instruction network.

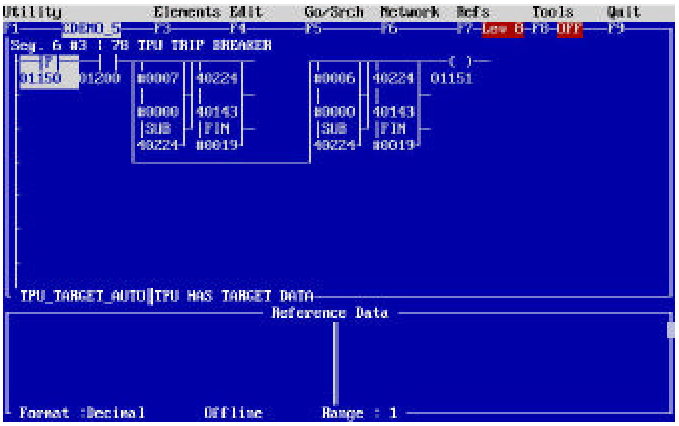


Figure 83 – Segment 5 Network 6 - Trip TPU If Readings Are Less Than 1 A Per Phase (Since 52A And B Contacts Are Not Wired Into Demo Case)

SEGMENT 6 NETWORK 4:

If there is no fault, calculate the loading prior to the trip of the TPU. This figure is used to determine if the DPU at the other end of the feeder has the capability to drive the load of the PCD 2000 . The ladder logic is illustrated in FIGURE 84 which follows.

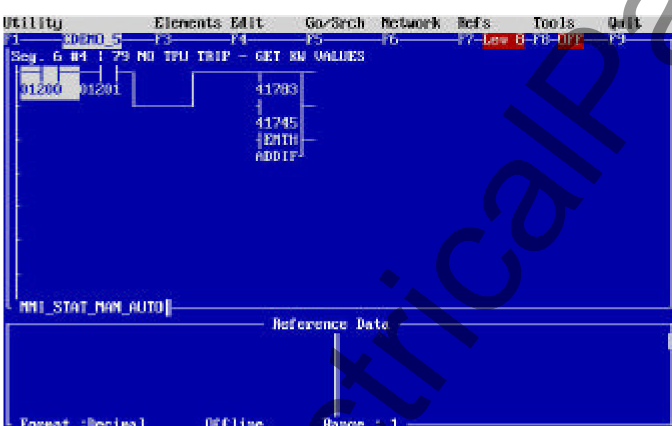


Figure 84 – If No TPU Trip Calculate KW Values Prior To Trip For Later Loading Calculations For Bus Line Sectionalizing Calculations

SEGMENT 6 NETWORKS 5, 6, 7, 8, 9, AND 10:

If the loading is appropriate for the DPU to supply the PCD circuit in lieu of the TPU which tripped, the following sequence occurs.

The DPU status is checked and the breaker is closed as long as the TPU 2000R breaker is tripped.

The program is delayed by 3 seconds and the PCD 2000 is then closed and the close is verified by the program.

The operator interface echo's the state of the restoration sequence on the operator interface as each of the steps is being performed.

The ladder logic networks are illustrated below as figures 85 through 87.

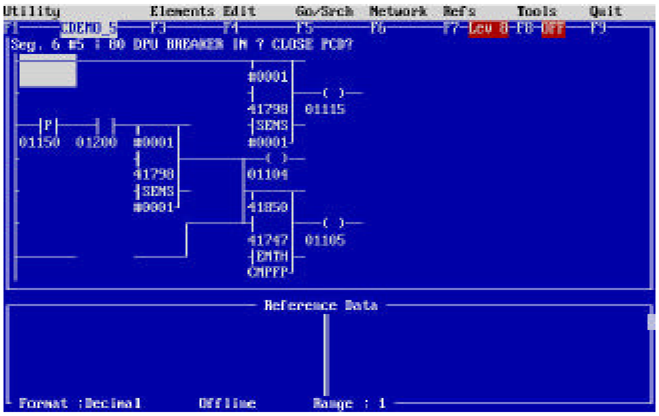


Figure 85 – Segment 6 Network 5 – Check DPU Breaker Status

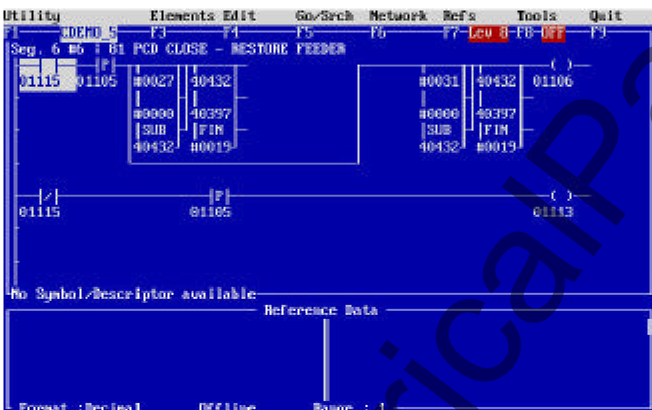


Figure 86 - Segment 6 Network 6 – Close PCD In Anticipation Of Feeder Restoration Upon TPU Trip

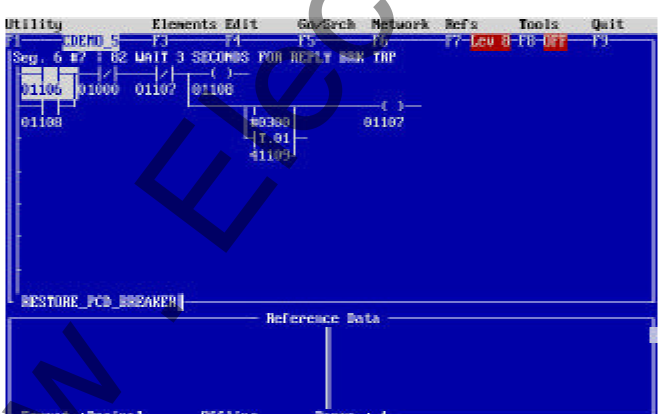


Figure 87 - Segment 6 Network 7 - Wait 3 Seconds For Breaker Action

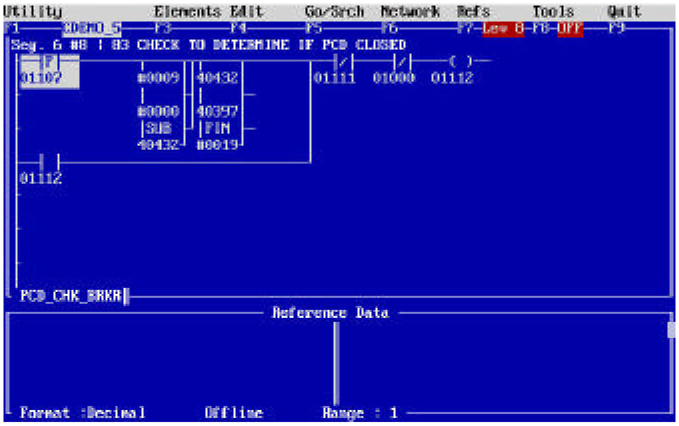


Figure 88 - Segment 6 Network 8 - Check To Determine If PCD Is Closed

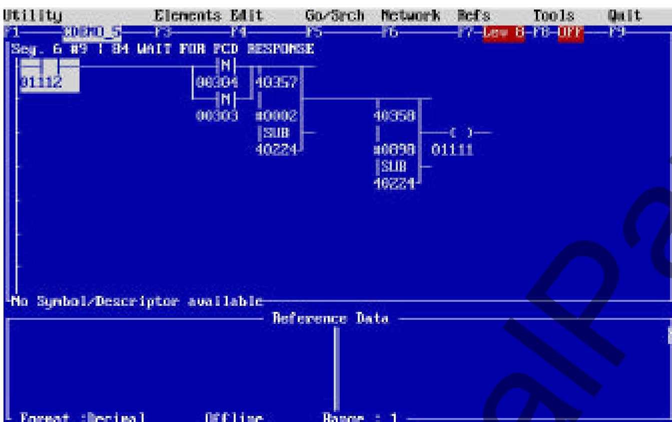


Figure 89 - Segment 6 Network 9 - Wait For PCD Response To Breaker Action

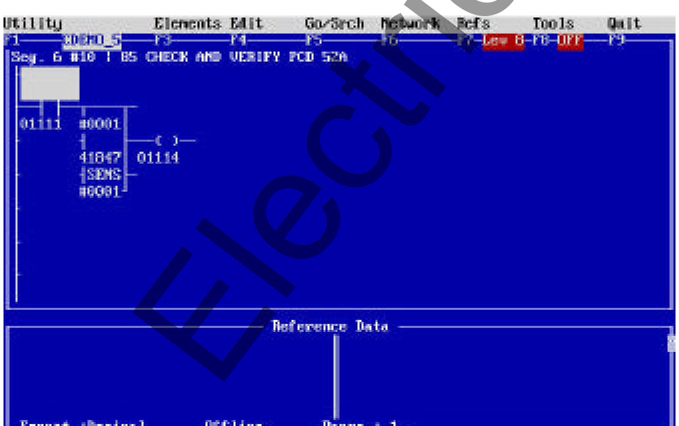


Figure 90 - Segment 6 Network 10 - Verify That Breaker Is Closed.

NOTE this program was developed for a demonstration of line restoration/sectionalizing applications. The TPU simulator used did not have breaker status, thus the need for the additional logic to calculate and maintain the correct status of the TPU breaker action (even if a manual trip command from the front panel was performed, the status of 52A and52B derived from this program is valid).

Subroutines

Two subroutines are included in this program. The subroutines are called from within the main program located in Segments 1 through 6 (JSR 2) and from within the subroutine (JSR 1). The subroutines are:

SUBROUTINE 1 – Convert an UNSIGNED 32 bit double register integer into a floating point number.

INPUT INTEGER to be converted: 41300 and 41301

FLOATING POINT RESULT located in 41351 and 41352.

SUBROUTINE 2 – Convert a SIGNED 32 bit double register integer into an absolute value floating point number.

INPUT INTEGER 41300 and 41301

NORMALIZED FLOATING POINT NUMBER 41330 and 40331.

These subroutines require constants to be placed in specific registers as illustrated in the constant screen windows listed at the end of this document. The constant values are used in allowing the subroutine to calculate the numbers correctly. NOTE: these subroutines are required for three reasons:

- The TPU,DPU and PCD use true integer numbers and the PLC only calculates numbers using integer math for a range of 0000 to 9999 (Compact 984 limitation).
- The COMPACT 984 PLC can perform mathematics calculations in IEEE Floating POINT, thus a calculation must be made from the PLC numbers (0000 to 9999 or 00000000 to 99999999 [double precision integer]) to floating point numbers.
- The MAGELIS MMI cannot display IEEE floating point numbers, so the results of the floating point number must be changed to the integer format required by the MMI.

SEGMENT 7, the last segment in the program, is not set up in the ladder logic segment scheduler (as is necessary for ladder logic subroutines). As illustrated in the ladder logic segments, 1 through 9 (FIGURES 91 through 96), the subroutine starts with a LAB instruction and ends at the RET command. The ladder logic segments are listed with the constants required for operation.

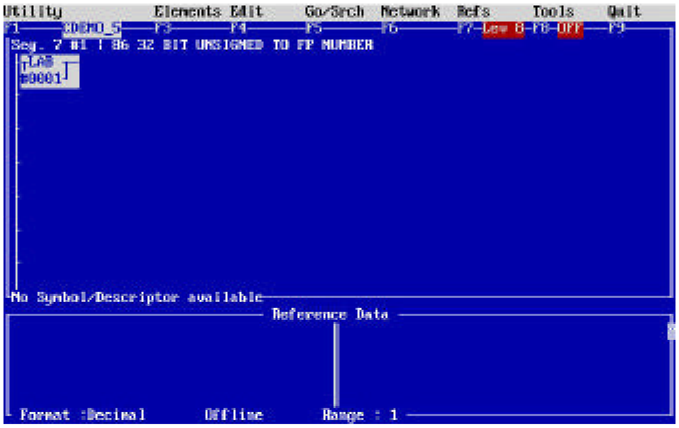


Figure 91 - Segment 7 Network 1 - 32 Bit Integer To Floating Point Number . Subroutine 1

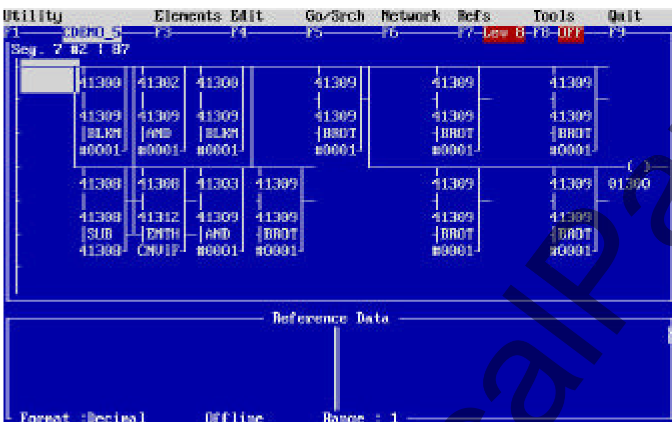


Figure 92 - Segment 7 Network 2 - 32 Bit Integer To Floating Point Number . Subroutine 1

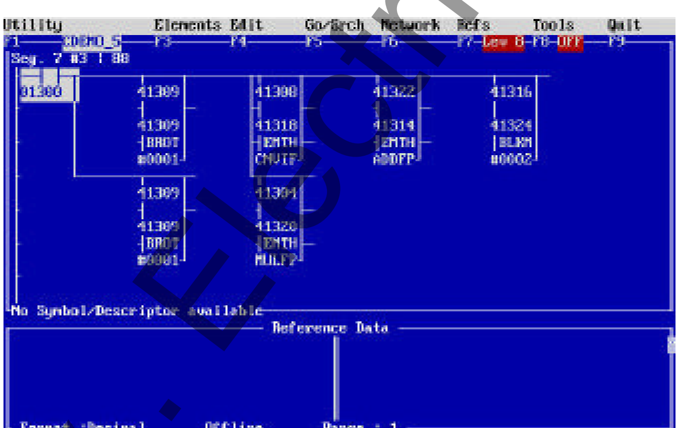


Figure 93 - Segment 7 Network 3 -32 Bit Integer To Floating Point Number . Subroutine 1

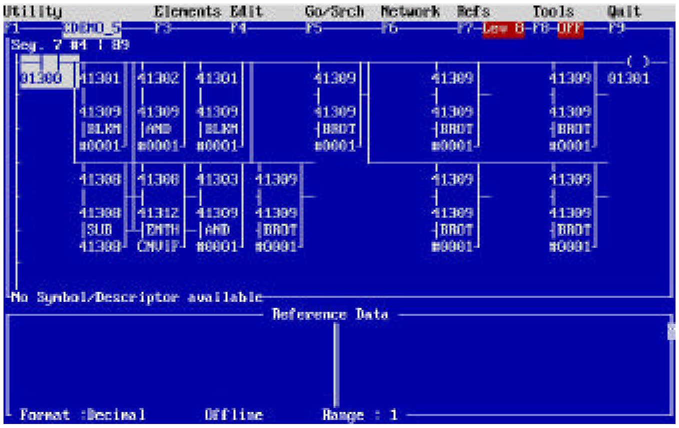


Figure 94 - Segment 7 Network 4 - 32 Bit Integer To Floating Point Number. Subroutine 1

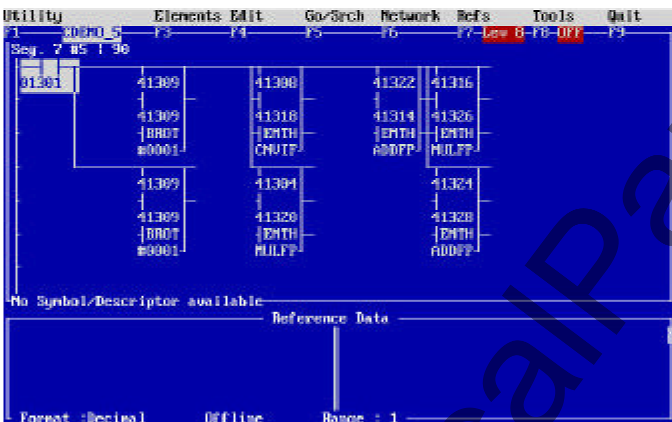


Figure 95 - Segment 7 Network 5 - 32 Bit Integer To Floating Point Number. Subroutine 1

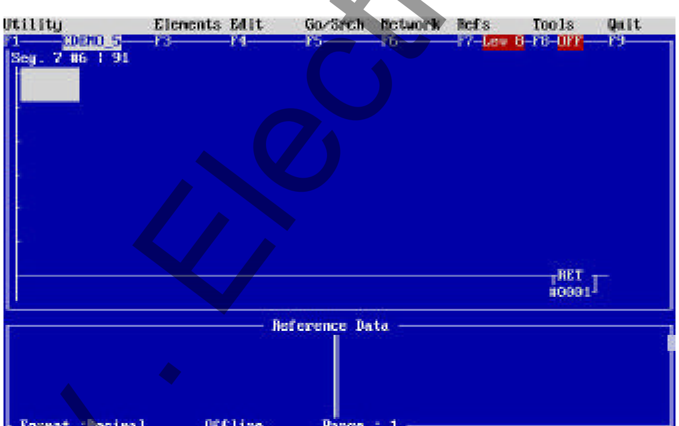


Figure 96 - Segment 7 Network 6 - 32 Bit Integer To Floating Point Number. Subroutine 1

Subroutine 2 uses subroutine 1 and it takes a negative number and converts it to a positive number (used for the sake of this demo to vary the KW readings using those from the simulator). This is used because the simulators use a single phase source and makes KW readings appear negative on some of the phases.

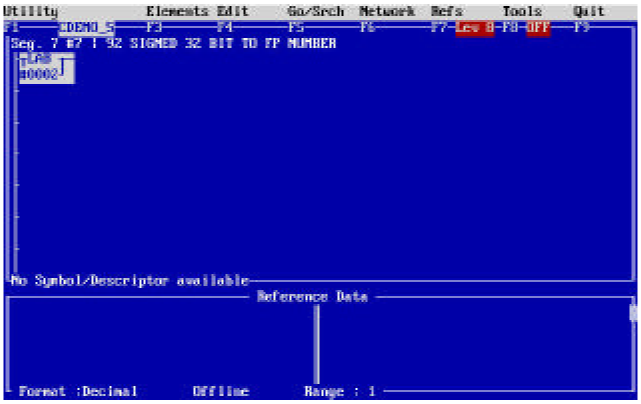


Figure 97 - Segment 7 Network 7 - 32 Bit Signed Integer To Floating Point Number. Subroutine 2

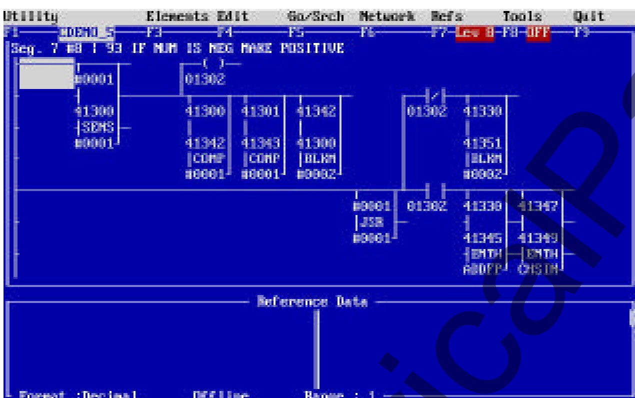


Figure 98 - Segment 7 Network 8 - 32 Bit Signed Integer To Floating Point Number. Subroutine 2

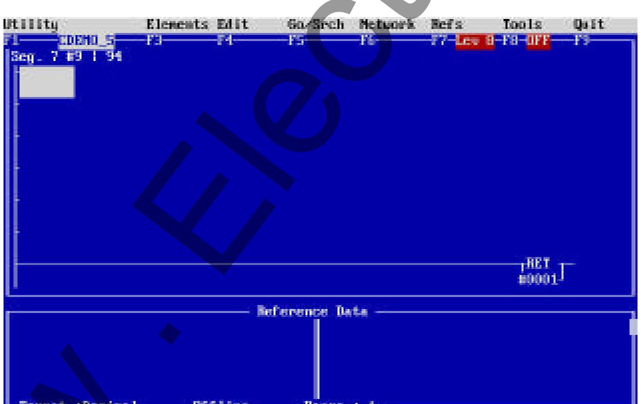
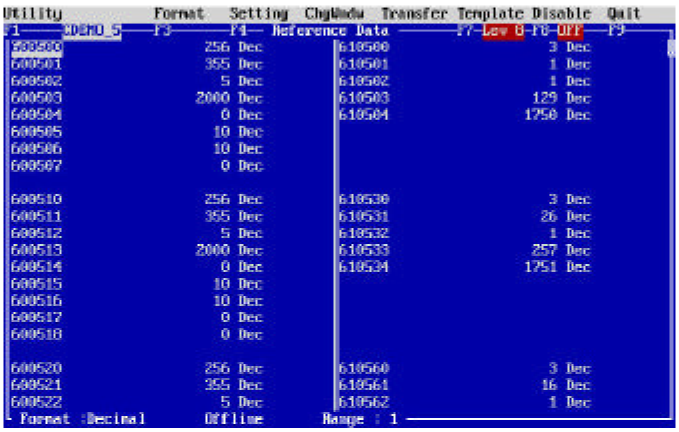


Figure 99 - Segment 7 Network 9 - 32 Bit Signed Integer To Floating Point Number. Subroutine 2

PLC Program Constants

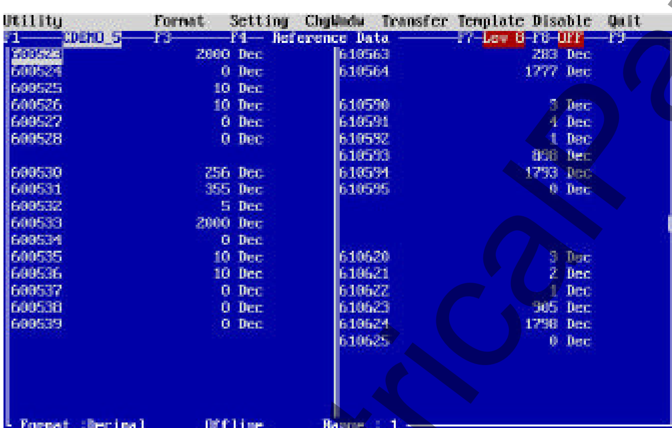
As illustrated previously, there are certain constants in 4X memory and 6x memory which must be preloaded into PLC memory for this program to function properly. The screens which follow illustrate the contents of each of the registers which are needed for this program's proper operation.



Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 Off F9

600500	256 Dec	610500	3 Dec
600501	355 Dec	610501	1 Dec
600502	5 Dec	610502	1 Dec
600503	2000 Dec	610503	129 Dec
600504	0 Dec	610504	1750 Dec
600505	10 Dec		
600506	10 Dec		
600507	0 Dec		
600510	256 Dec	610530	3 Dec
600511	355 Dec	610531	26 Dec
600512	5 Dec	610532	1 Dec
600513	2000 Dec	610533	257 Dec
600514	0 Dec	610534	1751 Dec
600515	10 Dec		
600516	10 Dec		
600517	0 Dec		
600518	0 Dec		
600520	256 Dec	610560	3 Dec
600521	355 Dec	610561	16 Dec
600522	5 Dec	610562	1 Dec

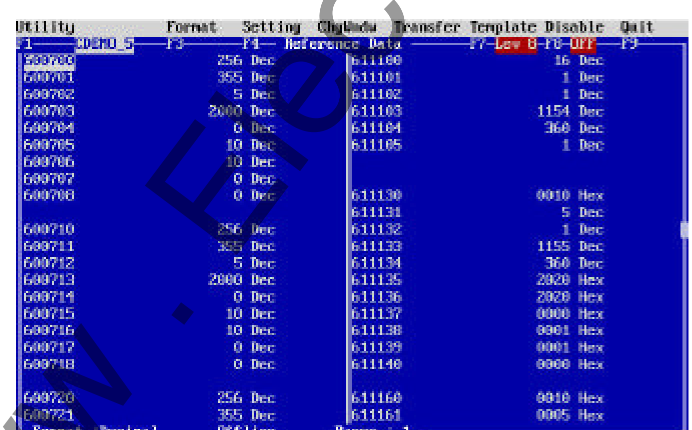
Format :Decimal Offline Range : 1



Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 Off F9

600523	2000 Dec	610563	283 Dec
600524	0 Dec	610564	1777 Dec
600525	10 Dec		
600526	10 Dec	610590	3 Dec
600527	0 Dec	610591	4 Dec
600528	0 Dec	610592	1 Dec
		610593	898 Dec
600530	256 Dec	610594	1793 Dec
600531	355 Dec	610595	0 Dec
600532	5 Dec		
600533	2000 Dec		
600534	0 Dec		
600535	10 Dec	610620	3 Dec
600536	10 Dec	610621	2 Dec
600537	0 Dec	610622	1 Dec
600538	0 Dec	610623	905 Dec
600539	0 Dec	610624	1798 Dec
		610625	0 Dec

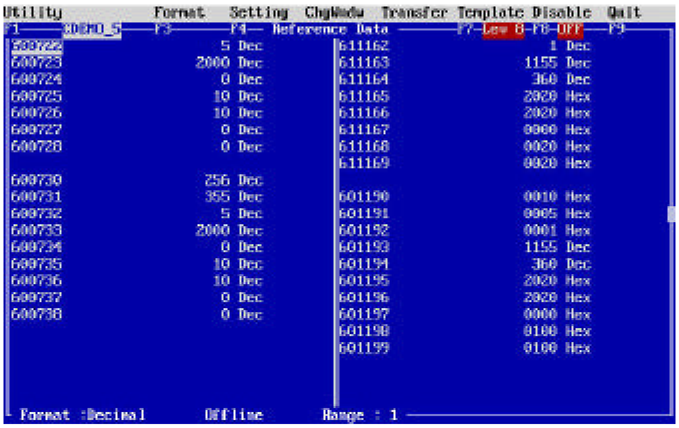
Format :Decimal Offline Range : 1



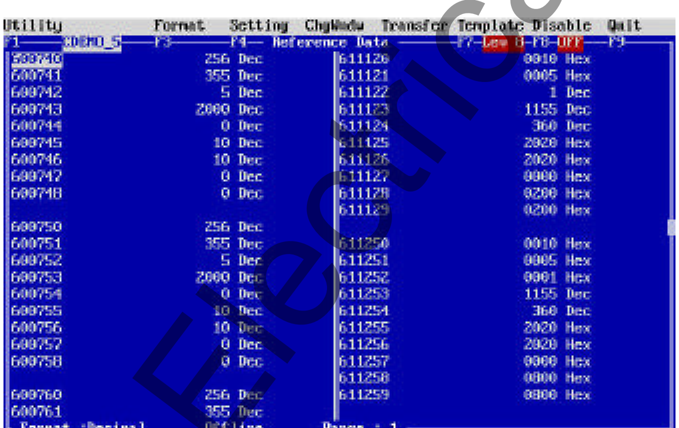
Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 Off F9

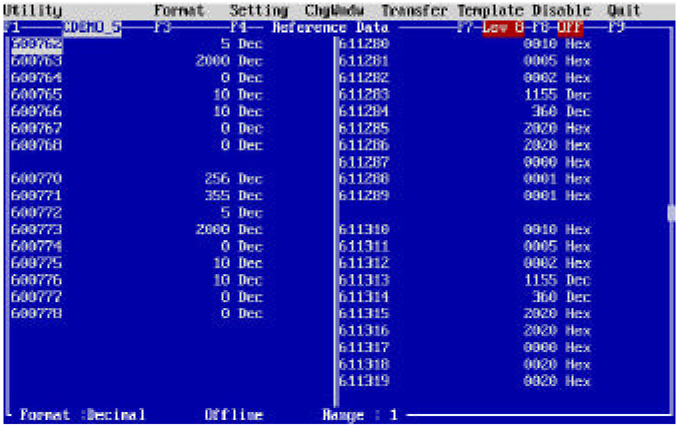
600700	256 Dec	611100	16 Dec
600701	355 Dec	611101	1 Dec
600702	5 Dec	611102	1 Dec
600703	2000 Dec	611103	1154 Dec
600704	0 Dec	611104	360 Dec
600705	10 Dec	611105	1 Dec
600706	10 Dec		
600707	0 Dec		
600708	0 Dec	611130	0010 Hex
		611131	5 Dec
600710	256 Dec	611132	1 Dec
600711	355 Dec	611133	1155 Dec
600712	5 Dec	611134	360 Dec
600713	2000 Dec	611135	2020 Hex
600714	0 Dec	611136	2020 Hex
600715	10 Dec	611137	0000 Hex
600716	10 Dec	611138	0001 Hex
600717	0 Dec	611139	0001 Hex
600718	0 Dec	611140	0000 Hex
600720	256 Dec	611160	0010 Hex
600721	355 Dec	611161	0005 Hex

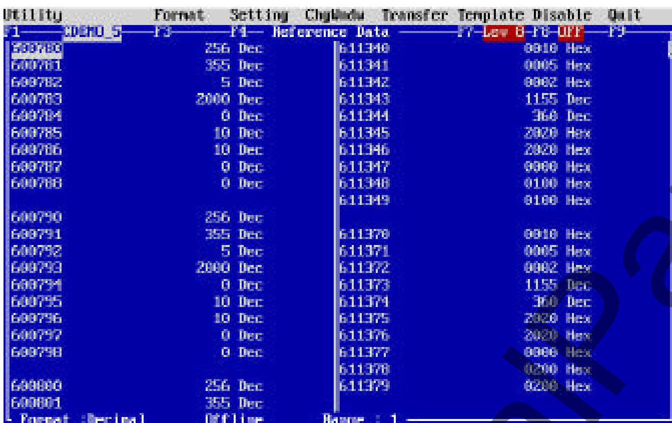
Format :Decimal Offline Range : 1

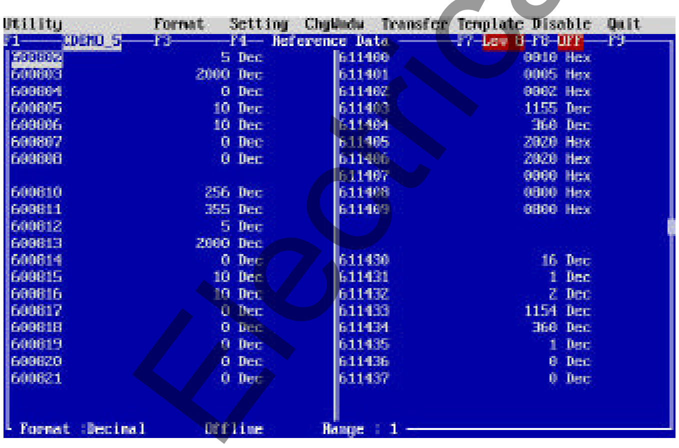

Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Low 8 F8 OFF F9
600722 5 Dec | 611162 1 Dec
600723 2000 Dec | 611163 1155 Dec
600724 0 Dec | 611164 360 Dec
600725 10 Dec | 611165 2020 Hex
600726 10 Dec | 611166 2020 Hex
600727 0 Dec | 611167 0000 Hex
600728 0 Dec | 611168 0020 Hex
611169 0020 Hex
600730 256 Dec
600731 355 Dec | 601190 0010 Hex
600732 5 Dec | 601191 0005 Hex
600733 2000 Dec | 601192 0001 Hex
600734 0 Dec | 601193 1155 Dec
600735 10 Dec | 601194 360 Dec
600736 10 Dec | 601195 2020 Hex
600737 0 Dec | 601196 2020 Hex
600738 0 Dec | 601197 0000 Hex
601198 0100 Hex
601199 0100 Hex
Format :Decimal Offline Range : 1

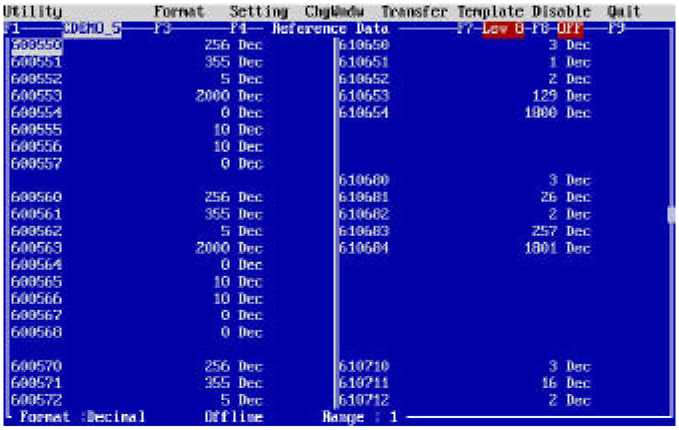

Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Low 8 F8 OFF F9
41300 CONV_VAL-> 0 Dec | 41322 0 Dec
41301 CONV_VAL-> 0 Dec | 41323 0 Dec
41302 CONST_00FF 255 Dec | 41324 0 Dec
41303 CONST_FF00 65280 Dec | 41325 0 Dec
41304 CONST_25-> 0 Dec | 41326 CONST_65-> 65280 Dec
41305 17280 Dec | 41327 18303 Dec
41306 0 Dec | 41328 0 Dec
41307 0 Dec | 41329 0 Dec
41308 0 Dec | 41330 FP_RES_3-> 0 Dec
41309 0 Dec | 41331 0 Dec
41310 0 Dec | 41332 0 Dec
41311 0 Dec | 41333 0 Dec
41312 0 Dec | 41334 0 Dec
41313 0 Dec | 41335 0 Dec
41314 0 Dec | 41336 0 Dec
41315 0 Dec | 41337 0 Dec
41316 0 Dec | 41338 0 Dec
41317 0 Dec | 41339 0 Dec
41318 0 Dec | 41340 constant-> 255 Dec
41319 0 Dec | 41341 constant-> 65280 Dec
41320 0 Dec | 41342 831 Dec
41321 0 Dec | 41343 0 Dec
Format :Decimal Offline Range : 1

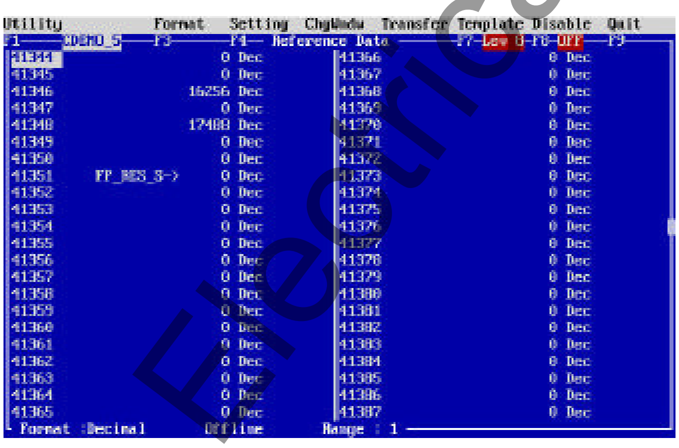

Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Low 8 F8 OFF F9
600740 256 Dec | 611120 0010 Hex
600741 355 Dec | 611121 0005 Hex
600742 5 Dec | 611122 1 Dec
600743 2000 Dec | 611123 1155 Dec
600744 0 Dec | 611124 360 Dec
600745 10 Dec | 611125 2020 Hex
600746 10 Dec | 611126 2020 Hex
600747 0 Dec | 611127 0000 Hex
600748 0 Dec | 611128 0200 Hex
611129 0200 Hex
600750 256 Dec
600751 355 Dec | 611250 0010 Hex
600752 5 Dec | 611251 0005 Hex
600753 2000 Dec | 611252 0001 Hex
600754 0 Dec | 611253 1155 Dec
600755 10 Dec | 611254 360 Dec
600756 10 Dec | 611255 2020 Hex
600757 0 Dec | 611256 2020 Hex
600758 0 Dec | 611257 0000 Hex
611258 0800 Hex
600760 256 Dec | 611259 0800 Hex
600761 355 Dec
Format :Decimal Offline Range : 1


Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 OFF F9
600762 5 Dec 611280 0010 Hex
600763 2000 Dec 611281 0005 Hex
600764 0 Dec 611282 0002 Hex
600765 10 Dec 611283 1155 Dec
600766 10 Dec 611284 360 Dec
600767 0 Dec 611285 2020 Hex
600768 0 Dec 611286 2020 Hex
611287 0000 Hex
600770 256 Dec 611288 0001 Hex
600771 355 Dec 611289 0001 Hex
600772 5 Dec
600773 2000 Dec 611310 0010 Hex
600774 0 Dec 611311 0005 Hex
600775 10 Dec 611312 0002 Hex
600776 10 Dec 611313 1155 Dec
600777 0 Dec 611314 360 Dec
600778 0 Dec 611315 2020 Hex
611316 2020 Hex
611317 0000 Hex
611318 0020 Hex
611319 0020 Hex
Format :Decimal Offline Range : 1


Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 OFF F9
600780 256 Dec 611340 0010 Hex
600781 355 Dec 611341 0005 Hex
600782 5 Dec 611342 0002 Hex
600783 2000 Dec 611343 1155 Dec
600784 0 Dec 611344 360 Dec
600785 10 Dec 611345 2020 Hex
600786 10 Dec 611346 2020 Hex
600787 0 Dec 611347 0000 Hex
600788 0 Dec 611348 0100 Hex
611349 0100 Hex
600790 256 Dec
600791 355 Dec 611370 0010 Hex
600792 5 Dec 611371 0005 Hex
600793 2000 Dec 611372 0002 Hex
600794 0 Dec 611373 1155 Dec
600795 10 Dec 611374 360 Dec
600796 10 Dec 611375 2020 Hex
600797 0 Dec 611376 2020 Hex
600798 0 Dec 611377 0000 Hex
611378 0200 Hex
600800 256 Dec 611379 0200 Hex
600801 355 Dec
Format :Decimal Offline Range : 1


Utility Format Setting ChgWndw Transfer Template Disable Quit
F1 DEMO_5 F3 F4 Reference Data F7 Lev 0 F8 OFF F9
600802 5 Dec 611400 0010 Hex
600803 2000 Dec 611401 0005 Hex
600804 0 Dec 611402 0002 Hex
600805 10 Dec 611403 1155 Dec
600806 10 Dec 611404 360 Dec
600807 0 Dec 611405 2020 Hex
600808 0 Dec 611406 2020 Hex
611407 0000 Hex
600810 256 Dec 611408 0800 Hex
600811 355 Dec 611409 0800 Hex
600812 5 Dec
600813 2000 Dec
600814 0 Dec 611430 16 Dec
600815 10 Dec 611431 1 Dec
600816 10 Dec 611432 2 Dec
600817 0 Dec 611433 1154 Dec
600818 0 Dec 611434 360 Dec
600819 0 Dec 611435 1 Dec
600820 0 Dec 611436 0 Dec
600821 0 Dec 611437 0 Dec
Format :Decimal Offline Range : 1





Conclusion

As illustrated, the ladder logic is segmented according to the tasks required by the PLC. The tasks are:

- MSTR Modbus Plus Control
- XMIT Modbus Control of the Radio Modem Polling

- Operator Interface (MAGELLIS MMI) Control and Function Key Processing
- Calculation of Feeder Loading
- Completion of Line Sectionalizing Routines
- Subroutines to allow the PLC to easily calculate mathematics in floating point mathematics.

The ABB protective relay becomes a versatile device with the inclusion of common off the shelf equipment such as PLC's Operator interfaces such as MAGELIS and inexpensive radio modems. Building systems based upon solid communication protocols such as Modbus Plus (giving fast response to equipment communicating inside a substation) and Modbus (allowing efficient communication between devices at remote locations) allows complex systems to be added and engineered incrementally as a budget permits. Events occurring within the relay can easily be accessed. The easy to configure programming language within an Modicon PLC allows for additional automation capability to be added within a substation at minimal cost and minimal programming capability. It is easy to see why the use of PLC's and microprocessor relays is more prevalent in today's substation designs.

This program has been used with standard ABB product simulators. It was first presented in a joint Groupe Schneider and ABB seminar in 1998. Copies of this program may be obtained from ABB at no charge. It is intended for this program to serve as a guide for using PLC's and ABB IED's in automation systems. There is no expressed or implied warranty or any implication as to the accuracy of the logic and the content within.

Contributed by:
John Popiak
12/00

ABB, Inc.
7036 Snowdrift Road
Allentown, PA 18106
800-634-6005 Fax 610-395-1055
Email: powerful.ideas@us.abb.com
Web: www.abb.com/substationautomation