305 Programmable CPU

User's Manual

FACTS engineering

Order Number: F3-RTU-M

TRADEMARKS

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the property of their respective owners.

Texas Instruments is a registered trademark of Texas Instruments, Inc. **TI**, **TIWAY**, **Series 305**, **Series 405**, **TI305**, and **TI405** are trademarks of Texas Instruments, Inc. **Siemens** and **SIMATIC** are registered trademarks of Siemens, AG. **GE** is a registered trademark of General Electric Corporation. **Series One** is a registered trademark of GE Fanuc Automation North America, Inc. **Modbus** is a registered trademark of Gould, Inc. **IBM** is a registered trademark of International Business Machines. **MS-DOS** and **Microsoft** are registered trademarks of Microsoft Corporation. **Windows** is a trademark of Microsoft Corporation. **OPTOMUX** and **PAMUX** are trademarks of OPTO 22. **MCS** is a trademark of Intel Corporation. **Wordperfect** is a trademark of WordPerfect Corporation. **Wordstar** is a trademark of MicroPro International Corporation.

COPYRIGHT

WARNING

Thank you for purchasing automation equipment from FACTS Engineering. We want your new FACTS Engineering automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manuafacturers Association (NEMA). There may be local regulatory or government offices that can help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 1-800-783-3225.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware and software, nor to provide for every possible contingency in connection with installation, operation, and maintenance. Features may be described herein which are not present in all hardware and software systems. FACTS Engineering assumes no obligation of notice to holders of this document with respect to changes subsequently made. FACTS Engineering retains the right to make changes to hardware and software at any time, without notice. FACTS Engineering makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability of fitness for purpose shall apply.

Manual History

Refer to this history in all correspondence and/or discussion of this manual.

Title: 305 Programmable Remote Terminal Unit User's Manual

Part Number: F3-RTU-M

Issue	Date	Effective Pages	Description of Changes
First	1/94	Cover	First Edition
		Trademarks	
		Warning	
		Manual History	
		TOC	
		Chapter 1	
	6/94	Chapter 2	Added SYSTEM 6 and 7
		Chapter 3	
		Chapter 4	
		Chapter 5	
		Appendix A	
		Appendix B	
		Appendix C	
	2/94	Appendix D	Added TADV,PPR,PINP2,PRIMN ANSR, ASUM, LCSTAT
		Appendix E	
		Appendix F	

305 Programmable RTU User's Manual
Table of Contents

CHAPTER 1: INTRODUCTION

This manual describes details specific to the F3-RTU Programmable Remote Terminal Unit. This document should be used to supplement the FACTS Extended BASIC User's Reference when programming the FACTS Engineering F3-RTU module.

The F3-RTU module may be programmed in BASIC to read and write digital and analog I/O and communicate with external devices through the two RS232/422/485 serial communication ports.
The F3-RTU module communicates with up to 184 DL305 I/O points using the ACTIVATE, DEACTIVATE, and STATUSIO instructions and the analog data-base.

The F3-RTU module is installed in the CPU slot of a DL305 base. The F3-RTU module communicates with external devices using the the built in serial ports. The F3-RTU module supports the following communication protocols.

DYNAMIC PROTOCOL

The Dynamic protocol is built into the F3-RTU operating system. The Dynamic protocol can be used with radio or direct wired connections. The Dynamic protocol supports many RTU features including:

_ Short and concise message format
_ Program uploading and downloading over Network
_ Digital Inputs support status, transition, counting (accumulation), duration, and pulse rate (flow) with report by exception if values are out of user defined deadband
_ Analog inputs support status, minimum, maximum, average, and accumulation with report by exception if values are out of user defined deadband
_ Optional data time stamping (in seconds since midnight)
_ Report by exception may be solicited (polled) or unsolicited via dial up telephone or peer to peer packet radio network.

MODBUS RTU PROTOCOL

The Modbus RTU protocol is implemented as a user downloadable BASIC program. The Modbus protocol can be used with radio or direct wired communications. This Modbus implementation supports:

_ Modbus Function Codes 1,2,3,4,5,6,15,16
_ Register Data Types correspond to Analog Channels
_ FACTS Extensions allow reading logged data values

CUSTOM PROTOCOL

FACTS Extended BASIC simplifies the implementation of custom communication protocols through an extensive instruction set.

_ SETPORT for configuring serial port baud rate and framing
_ SETINPUT for configuring exactly how the INPUT statment works
_ INLEN to check the number of characters in an input buffer
_ INPLEN to check the number of characters last input
_ PRINT USING for formatting serial port output
_ ASC and CHR$ for manipulating binary strings
_ ERRCHK for calculating Checksum, LRC, or CRC-16 error checking codes

F3-RTU GENERAL SPECIFICATIONS

Mounting Requirement	- CPU Slot of a DL305 base
Power Consumption	- 416 mA @ 5 Vdc maximum (supplied by base)
Operating Enviroment	- 0 to 60 degrees C (32 to 140 degrees F) - 5 to 95% humidity (non-condensing)
Processor	- Intel 80C51FA-1
Clock Speed	- 16 Mhz
Memory	- 128K Total (64K Data, 64K Program)
Physical Connectors	- Two 9 Pin D type connectors - Port 1 (Top 9 Pin) - Port 2 (Bottom 9 Pin)
Indicator LEDs	- ON, R2, T2, ìP, OK, R1, T1 (See Appendix B)
Port 1	- RS232/422/485 Jumper Selectable - 57600 Baud Maximum (Jumper Configurable)
Port 2	- RS232/422/485 Jumper Selectable - 9600 Baud Maximum (SETPORT Configurable)
Additional Features	- Battery Backed Calendar/Clock - Programmable from Port 1 or Port 2

DYNAMIC PROTOCOL SUPPORT

The F3-RTU supports Texas Instruments DYNAMIC Protocol. Port 1 of the F3-RTU can be dedicated to the DYNAMIC Protocol, assigned to BASIC or shared by both. The DYNAMIC protocol supports the following features:

- Read and Write local I/O from a remote location
- Remote upload and download of the BASIC control program
- Report by Exception
- Unsolicited report by exception

A summary of the F3-RTU DYNAMIC implementation is in Appendix D. For a complete description of the protocol, please refer to the Texas Instruments "DYNAMIC PROTOCOL REFERENCE GUIDE, Designing SCADA Systems, December 1986".

I/O MODULES NOT SUPPORTED BY THE F3-RTU

Currently the 305 Bridge CPU supports over 36 differnet Direct Logic 305 I/O modules. The sales catalog contains a complete listing of all available I/O modules. Please call us if you do not find the I/O interface you need for your application.

The only Direct Logic modules which are incompatible with the Bridge CPU are listed below.

D3-HSC	High Speed Counter
D3-04AD	Analog Input, 8-Bit, 4 Points
D3-02DA	Analog Output, 8-Bit, 2 Points

CHAPTER 2: F3-RTU STATEMENTS

ACTIVATE

Function Turns ON a discrete output or a group of discrete outputs

Syntax ACTIVATE(*I/O register*) = *mask*

Shorthand A.

Usage ACTIVATE is used to turn specified outputs in a group of eight outputs ON without affecting the state of other outputs in the group. *I/O register* specifies the output group.

mask is used to specify the output or group of outputs to turn ON. Turning ON a bit in the *mask* byte turns ON the corresponding output. To turn ON multiple bits in an I/O register, add the *mask* values together.

Output	***mask***
0	1
1	2
2	4
3	8
4	16 (10H)
5	32 (20H)
6	64 (30H)
7	128 (80H)

The first 8 slots in a base have two *I/O register* groups for each slot. Either 8 or 16 point I/O modules may be installed. The next 8 slots in the CPU base and the expansion base have one *I/O register* group each. Only 8 point modules may be installed in slots 8-15 (first I/O module is in slot 0). See Appendix C for I/O register addressing of different base configurations.

Slot	*I/O Register* (Low Byte)	*I/O Register* (High Byte)
0	0	1
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15
8	16	N/A
9	17	
10	18	
11	19	
12	20	
13	21	
14	22	
15	23	

Example

```
1000    REM   Turn ON outputs 3 and 6 in slot 3 (low byte)
1010    REM   All other outputs are unchanged
1020    ACTIVATE(6)=72
1030    REM   Turn ON outputs 0, 2, and 7 in slot 3 (high byte)
1040    BITS=0 : BIT(0)=1 : BIT(2)=1 : BIT(7)=1
1050    ACTIVATE(7)=BITS
```

DEACTIVATE

Function Turns OFF a discrete output or a group of discrete outputs

Syntax DEACTIVATE(*I/O register*) = *mask*

Shorthand D.

Usage DEACTIVATE is used to turn specified outputs in a group of eight outputs OFF without affecting the state of other outputs in the group. *I/O register* specifies the output group.

mask is used to specify the output or group of outputs to turn OFF. Turning ON a bit in the *mask* byte turns OFF the corresponding output. To turn OFF multiple bits in an I/O register, add the *mask* values together.

Output	***mask***
0	1
1	2
2	4
3	8
4	16 (10H)
5	32 (20H)
6	64 (30H)
7	128 (80H)

The first 8 slots in a base have two *I/O register* groups for each slot. Either 8 or 16 point I/O modules may be installed. The next 8 slots in the CPU base and the expansion base have one *I/O register* group each. Only 8 point modules may be installed in slots 8-15 (first I/O module is in slot 0). See Appendix C for I/O register addressing of different base configurations.

Slot	*I/O Register* (Low Byte)	*I/O Register* (High Byte)
0	0	1
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15
8	16	N/A
9	17	
10	18	
11	19	
12	20	
13	21	
14	22	
15	23	

Example

```
1000   REM   Turn OFF outputs 3 and 6 in slot 3 (low byte)
1010   REM   All other outputs are unchanged
1020   DEACTIVATE(6)=72
1030   REM   Turn OFF outputs 0, 2, and 7 in slot 3 (high byte)
1040   BITS=0 : BIT(0)=1 : BIT(2)=1 : BIT(7)=1
1050   DEACTIVATE(7)=BITS
```

STATUSIO

Function STATUSIO sets the state of outputs and reads input status

Shorthand S.

Syntax *var* = STATUSIO (*I/O register*)
STATUSIO (*I/O register*) = *expr*

Usage STATUSIO is used to set the state of discrete outputs in a group of eight outputs to either ON or OFF. *I/O register* specifies the output group.

var contains the discrete input status for the specified *I/O register* group. An ON bit in the *var* byte has a corresponding ON input point.

expr is used to specify which outputs to turn OFF and which to turn ON. Turning ON a bit in the *expr* byte turns ON the corresponding output.

Output	***expr***
0	1
1	2
2	4
3	8
4	16 (10H)
5	32 (20H)
6	64 (30H)
7	128 (80H)

The first 8 slots in a base have two *I/O register* groups for each slot. Either 8 or 16 point I/O modules may be installed. The next 8 slots in the CPU base and the expansion base have one *I/O register* group each. Only 8 point modules may be installed in slots 8-15 (first I/O module is in slot 0). See Appendix C for I/O register addressing of different base configurations.

Slot	*I/O Register* (Low Byte)	*I/O Register* (High Byte)
0	0	1
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15
8	16	N/A
9	17	
10	18	
11	19	
12	20	
13	21	
14	22	
15	23	

Example

```
1000 REM   Turn ON outputs 3 and 6 in I/O register in slot 3
1010  REM   Turn all other outputs OFF
1020 STATUSIO(6)=72 : REM 33H
1030 REM   Display status of 8 inputs 1st I/O register in slot 1
1040 BITS=STATUSIO(2)
1050 FOR INP = 0 TO 7
1060 IF BIT(INP) THEN PRINT1 "ON   ", ELSE PRINT1 "OFF  ",
1070 NEXT INP
```

SYSTEM

Function	Read and Set system parameters that define CPU-RTU operation
Syntax	SYSTEM (*code*) = *expr* *var* = SYSTEM(*code*)
Usage	See page 40 for addition Dynamic specific system codes.

code	DESCRIPTION
0	Address of start of user PRM 0 .
1	Re-print last error message in direct mode, P. SYSTEM(1). In RUN mode, return line number of last error.
2	If true then add CRC-16 error check characters to PRINT statements and check CRC-16 on strings INPUT (COMERRn).
3	If true, output CRC-16 LSB first (has no effect on dynamic).
4	Timer milli-second value (was dby(71)).
5	ERRCHK result (CRC-16 word, LRC byte, sum check byte).
6	Address of first free memory location in bank 1. This is the location of the first byte after the end of the saved programs.
7	Selects port 1 for the programming port if expr. evaluates to 0. If expr. is 1 then port 2 becomes the programming port. This is the run time equivalent to COMMAND@n. If port 1 is selected then Dynamic is disabled at power up.
20	Start of slot 0 data base.
22	MSB channel 1 analog input or output offset from system(20).

CHAPTER 4: ANALOG DATA BASE

The analog I/O in a F3-RTU system is managed by an analog data base. Each analog slot has a block of memory dedicated to that slot. Analog input values for each channel are automatically updated in the data base for the slot the analog input is installed in. Analog outputs are updated by the CPU based on the values contained in the data base for that slot. The WORD instruction is used to read analog inputs and write analog outputs.

In addition to current analog values, the analog database contains deadband values used for RBE (Report by Exception) and URBE (Unsolicited RBE) message generation. Analog Minimum, Maximum, Average, and Accumulated values are also contained in the analog data base. Contact FACTS Engineering for applications information on these features.

The jumpers that specify the number of analog channels must be set correctly before analog inputs can be read and analog outputs can be written. See Chapter 5 (installation) for jumper settings.

WORD Instruction

Function WORD reads from or writes to a specific memory location two bytes

Syntax *var*=WORD(*address*)
WORD(*address*)=*expression*

Usage *address* is an expression from 0 to 131071(1FFFFH), representing a two byte memory location. WORD retrieves or assigns an integer value (0 to 65535). WORD can be used to store integer values in a region of memory protected from BASIC. WORD can also be used to retrieve integer values from any where in memory.

The following information is necessary for calculating the WORD *address* for specific slot and channel information:

SYSTEM(20) is the pointer to the start of slot 0 analog data base.

SYSTEM(22) is the offset to analog values from the start of each slot data base.

The data for each channel is separated by two memory locations. Each memory location is one byte.

The data base for each slot is separated by 100 Hex (256 Decimal) memory locations.

ANALOG INPUT EXAMPLE

The following example reads and displays 16 analog input channels from a 16 channel analog input (F3-16AD) module installed in slot 0 (Slot 0 is the first I/O slot after the CPU slot). If a lower density analog input module (8 channel or 4 channel) is used then the loop in line 70 and 120 should be modified to reflect the actual number of channels to read. The data base pointers are pre-calculated in this example to simplify the program. Note that the most significant 4 bits are masked off in line 130 (Analog data is 12 bits). Line 135 is required only on modules that use a sign bit such as the thermocouple input module F4-08THM.

```
10      REM -----------------------------------------------------
20      REM Read Analog Inputs Slot 0
30      REM
40      DIM AINPTR(16)
50      DIM AIN(16)
60     SLOT=0
65      REM Initialize Pointers to Analog Inputs for Slot 0
70      FOR CH=0 TO 15
80     AINPTR(CH+1)=SYSTEM(20)+SYSTEM(22)+(0100H*SLOT)+(CH*2)
90      NEXT CH
100     REM
110     PRINT2
120     FOR X=1 TO 16
125     REM Mask off Most Significant 4 Bits and assign
130    AIN(X)=WORD(AINPTR(X)).AND.0FFFH
135    IF (WORD(AINPTR(X)).AND.1000H)<>0 THEN AIN(X)=-AIN(I)
140     PRINT2 USING(##),"Channel ",X;
150     PRINT2 USING(####)," ->",AIN(X)
160     NEXT X
```

ANALOG OUTPUT EXAMPLE

The following example writes and displays 4 analog outputs to an analog output module (F3-04DA) installed in slot 1 (Slot 1 is the second I/O slot after the CPU slot). The data base pointers are pre-calculated in this example to simplify the program. Note that bit 12 is turned ON in line 160 with AOUT(X).OR.01000H so that the new value will be applied to the analog output.

```
10     REM -----------------------------------------------------
20     REM Write Analog Output Slot 1
30     REM
40     DIM AOUTPTR(16)
50     DIM AOUT(16)
60    SLOT=1
65     REM Initialize pointers to Analog Outputs for Slot 1
70     FOR CH=0 TO 15
80    AOUTPTR(CH+1)=SYSTEM(20)+SYSTEM(22)+(0100H*SLOT)+(CH*2)
90     NEXT CH
100    REM
110   AOUT(1)=2048 :  REM 50% Output on Channel 1
120   AOUT(2)=4095 :  REM 100% Output on Channel 2
130    REM
140    PRINT2
150    FOR X=1 TO 4
155    REM Turn on Bit 12 of the channel location so the new
156    REM value will be applied to the analog output channel
160   WORD(AOUTPTR(X))=AOUT(X).OR.01000H
170    PRINT2 USING(##),"Channel ",X;
180    PRINT2 USING(####)," ->",AOUT(X)
190    NEXT X
```

CHAPTER 5: INSTALLATION

F3-RTU JUMPER LOCATIONS

Many operation modes and I/O module configurations are supported by the F3-RTU. The configuration jumpers described in the following pages are:

```
Dynamic Remote Address (0-255) -> J2
Analog Slot Assignment         -> J4
Dynamic Baud Rate              -> J4
Dynamic Error Checking Method  -> J4
Clear Jumper                   -> CLR
RS232/422/485 Operation        -> RXD1 and RXD2
```

```
                       RXD2                RXD1
                      . . .               . . .


                       1 2
                     2 . .
                   1. .  . .                   CLR
                    . .  . .                   . .
                    . .  . .
                    . .  . .
                    . .  . .
                    . .  . .
                    . .  . .
                    . .  J2
                    J4
```

CLR

The CLR jumper is used to select the AUTOSTART mode that the module uses at reset. Placing the jumper on both posts selects AUTOSTART mode 0. Placing the jumper on a single post allows the module to use the last stored AUTOSTART parameters (this is the default factory setting).

For normal operation, place the CLR jumper on a single post. See Appendix B for troubleshooting information using the CLR jumper.

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@1, remove LOCKOUT, and clear stored AUTOSTART information.

CONFIGURATION JUMPERS

There are two sets of configuration jumpers on the smaller F3-RTU daughter board. The module is shipped from the factory with shunts on all the jumpers. This configures the module for 38400 baud, all analog I/O, CRC-16 error checking and DYNAMIC address 255. The jumpers are defined as shown in the following two diagrams.

NOTE: These jumpers have no effect when the built-in Dynamic protocol is not used.

```
                                            8pt Header "J2"

DYNAMIC Remote Address                   2  • • • • • • • •  J2
                                         1  • • • • • • • •


   + 1
   + 2
   + 4
   + 8
  + 16
  + 32
  + 64
 + 128
```

```
                                   8pt Header "J4"

                                2  • • • • • • • •  J4
                                1  • • • • • • • •

AIn Slots + 1
AIn Slots + 2
AIn Slots + 4
AIn Slots + 8
Protocol Select
Baud 0
Baud 1
Baud 2
```

ANALOG SLOT SELECT

All analog modules must be installed in contiguous slots beside the CPU. Install jumpers on AIn Slots + n in the combination that matches the number of analog modules that are installed. All 305 slots after slot 7 (slot next to CPU is slot 0) and all expansion rack slots are 8 point digital slots. See Appendix C for I/O register addressing of different base configurations.

PROTOCOL SELECT

Install the Protocol Select jumper (factory default) for Enhanced CRC-16 error checking of the DYNAMIC message.

Remove the Protocol Select jumper for standard Block Check Character error checking.

NOTE: This jumpers has no effect when the built-in Dynamic protocol is not used.

I/O CONFIGURATION

Input or Output modules may be installed in any order and are automatically identified by the RTU. If the built-in Dynamic protocol is used, the first I/O module installed next to the CPU should be an output module.

DYNAMIC PORT (PORT 1) BAUD RATE CONFIGURATION

The F3-RTU supports baud rates of 300, 1200, 2400, 4800, 9600, 19200, 38400, and 57600 on Port 1. One of the eight possible baud rates is selected using the 3 Baud jumpers. A "n" in the following table designates that a shunt should be installed in this position. A "¨" indicates that the shunt in this position should be removed.

NOTE: If port 1 is controlled by BASIC (SYSTEM(19)=NOT(0)) then configure port 1 baud rate using the SETPORT statement. If port 1 is controlled by Dynamic (SYSTEM(19)=0) then configure port 1 baud rate using these jumpers.

	300	57600	1200	2400	4800	9600	19200	38400
Baud 0	¨	n	¨	n	¨	n	¨	n
Baud 1	¨	¨	n	n	¨	¨	n	n
Baud 2	¨	¨	¨	¨	n	n	n	n

COMMUNICATION PORT RECEIVER CONFIGURATION

Port 1 and Port 2 are jumper selectable for RS-232 or RS-422/485 operation.

To configure Port *n*, locate the 3 position jumper labeled "RXD*n*". To use the RS-232 receiver on port *n*, place the shunt on the two pins labeled "232" as shown below. To use the RS-422/485 receiver, place the shunt as shown below.

RS232 Jumper Positions	RXD2 . . .	RXD1 . . .
RS422/485 Jumper Positions	RXD2 . . .	RXD1 . . .

COMMUNICATION PORT TRANSMITTER CONFIGURATION

On ports 1 and 2, both the RS-232 and RS-422/485 transmitters operate simultaneously. This permits local RS-232 monitoring of the RS-422/485 port for troubleshooting.

If you do not wish to use either the RS-232 or the RS-422/485 drivers then simply leave them disconnected.

The RS-422/485 transmitters operate automatically on the programming port when the module is in the COMMAND mode. This enables programming the module using an RS-422/485 connection.

During normal BASIC program execution the RS-422/485 transmitters are tri-stated to a high impedance state. This permits RS-422 and RS-485 multidrop connections. To turn the transmitters ON during a PRINT statement use the *mulitdrop* parameter in the SETPORT statement.

COMMUNICATION PORT CONNECTIONS

PORT 1 - PROGRAMMING/COMMUNICATION

PIN	SYMBOL	DESCRIPTION
1	RXD1+	DATA INPUT HIGH, RS-422
2	TXD1	DATA OUTPUT, RS-232
3	RXD1	DATA INPUT, RS-232
4	RTS1	DTE HAS DATA TO XMIT, RS-232
5	CTS1	DTE MAY XMIT DATA, RS-232
6	RXD1-	DATA INPUT LOW, RS-422
7	GND*	SIGNAL GROUND
8	TXD1-	DATA OUTPUT LOW, RS-422
9	TXD1+	DATA OUTPUT HIGH, RS-422

PORT 2 - COMMUNICATION/PROGRAMMING

PIN	SYMBOL	DESCRIPTION
1	RXD2+	DATA INPUT HIGH, RS-422
2	TXD2	DATA OUTPUT, RS-232
3	RXD2	DATA INPUT, RS-232
4	N/C	+5 V @ 100 mA
5	N/C	No Connection
6	RXD2-	DATA INPUT LOW, RS-422
7	GND*	SIGNAL GROUND
8	TXD2-	DATA OUTPUT LOW, RS-422
9	TXD2+	DATA OUTPUT HIGH, RS-422

* Signal ground on both ports is the same as the 9 VDC common.

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

APPENDIX A: QUICK START

INITIAL MODULE OPERATION USING ABM COMMANDER PLUS

1. Review the ABM Commander Plus On-line instructions.

2. Run ABM Commander Plus by typing ABM from the DOS prompt.

3. Enter "Set communication parameters" to set the desired communication baud rate for ABM Commander Plus.

4. Enter Terminal and On-line Edit mode in the ABM Commander software.

5. Install the F3-RTU Programmable Remote Terminal Unit in the CPU slot of the DL305 base. Connect the communications cable from one serial port of your computer to port 2 on the F3-RTU module before power is applied. Turn ON the power to the base. See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS for cable wiring diagrams.

7. The module will respond with the sign on message and the command prompt. If you do not receive the sign on message, please follow the trouble shooting procedure in APPENDIX B.

```
F A C T S  E x t e n d e d  B A S I C  P l u s
Programmable Serial 305 Bridge CPU
F3-RTU, Version ..., (c)Copyright ....

AUTOSTART Mode, Program, Baud
Mode = 0, Edit
Program = 0
Port 1 Baud = 38400
(Port 2 = 9600) Programming

1 stored programs, 63769 program storage bytes free

PRM 0
READY
>
```

NOTE: The '>' character is the command prompt from the FACTS Extended BASIC interpreter. All COMMANDS must be entered at this prompt.

EDITING A PROGRAM

Once you receive the ">" BASIC COMMAND mode prompt enter the following:

```
>AUTOSTART 0, 0, baud rate  (baud rate entered step 4 above)

Mode = 0, Edit
Program = 0
Port 1 Baud = (baud rate that Port 1 jumpers specify)
(Port 2 = (baud rate entered above) ) Programming

>10 p2.                       Enter a small program
>65535 p2.
>RESET         Cycling the power will reset the module.
```

FACTS Extended BASIC will respond with the sign-on message and the command prompt.

```
>LIST          enter the list command
10     PRINT2
65535  PRINT2

PRM 0
READY
>
```

Note that mode zero uses the stored baud rate. The program in the edit buffer, PROGRAM 0, is always retained during loss of power.

SAVING A PROGRAM

```
>NEW

>10 P2."MY FIRST PROGRAM"
>SAVE     saves program in program memory

Saving program 1

1 stored programs, 65509 program storage bytes free

PRM 0
READY
>10 P2. "SECOND PROGRAM"
>SAVE          saves program in program memory

Saving program 2

2 stored programs, 65843 program storage bytes free

PRM 0
READY
>
```

AUTO RUN MODE

```
>AUTOSTART 1, 2     Run program 2 after a reset

Mode = 1, RUN (CLEAR)
Program = 2
Port 1 Baud = (baud rate that Port 1 jumpers specify)
(Port 2 = 9600) Programming
(Port 3 = (last baud rate that port 3 was set to) )

>RESET
SECOND PROGRAM


PRM 2
READY
>
```

The module now begins execution of the second stored program following a reset. Enter a PRM0 command followed by a LIST command to confirm that the program in the edit buffer (PROGRAM = 0) is still present.

DELETING A PROGRAM

```
>DELPRM 2

1 stored programs, 65509 bytes free

>RESET          Once again, reset the module.

SECOND PROGRAM       program 3 is now program 2

READY
```

CANCEL AUTO RUN MODE

>AUTOSTART 0   Back to edit mode using stored baud rate

Mode = 0, Edit
Program = 0
Port 1 Baud = (*baud rate* that Port 1 jumpers specify)
(Port 2 = 9600) Programming
(Port 3 = (*last baud* rate that port 3 was set to))

CHANGING THE PROGRAMMING PORT

When interfacing to two external devices, you will want to change the RS-232 programming port from Port 2 to Port 1. This is done using COMMAND@ as shown below.

```
>SETPORT 1,9600,N,8,1,N  Store baud rate you will use at Port 1

>COMMAND@1     Move your programming cable to Port 1
```

APPENDIX B: TROUBLE SHOOTING

UNABLE TO ESTABLISH COMMUNICATION WITH F3-RTU

1. Apply power and verify "ON" light. This indicates the power supply is good.

2. Verify "uP" light is OFF and if so go to step 3. If this light is flashing slowly then there is a problem with the processor board. In this case, verify that all the EPROM leads are fully installed in the socket and that they are not bent over under the chip.

3. If the Port 2 RXD LED (R2) flashes when data is entered on the terminal or sent from the host computer then go back to step 2. If the LED does not flash then use a RS-232 break-out box to determine if the problem is in the cable or the computer.

4. Remove the module and position the "CLR" jumper as shown.

Clear Jumper Placement

§ §

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@1, remove LOCKOUT, and clear stored AUTOSTART information.

5. Install the module and install the cable to the computer.

6. Power up the base.

7. The module will now respond with the sign on message.
F A C T S   E x t e n d e d   B A S I C   P l u s
...
READY
> (">" prompt character indicates BASIC is in COMMAND mode)

8. Type AUTOSTART 0,0

9. Remove power and the module. Reposition the "CLR" jumper.

Clear Jumper Placement

§ §

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@1, remove LOCKOUT, and clear stored AUTOSTART information.

APPENDIX C: BASE CONFIGURATIONS

I/O CONFIGURATION

Input or Output modules may be installed in any order and are automatically identified by the RTU. If the built-in Dynamic protocol is used, the first I/O module installed next to the CPU should be an output module.

BASES

Note: Slot 0 is referred to as the first slot available for an I/O module. In a CPU base it is first slot beside the CPU. In an Expansion base it is the first slot beside the Power Supply.

```
  8   7   6   5   4   3   2   1   0   C
                                      P          10 Slot CPU
 700/100                     EXP/CPU  U


          6   5   4   3   2   1   0   C
                                      P          8 Slot CPU
                                      U


                      3   2   1   0   C
                                      P          5 Slot CPU
                     1,3/2            U


  9   8   7   6   5   4   3   2   1   0
                                                 10 Slot EXP
 700/100                     EXP/CPU


                      4   3   2   1   0
                                                 5 Slot EXP
                     1,3/2
```

Valid Base Configurations

10 Slot CPU Base
8 Slot CPU Base
5 Slot CPU Base
10 Slot CPU Base and a 5 Slot EXP Base
10 Slot CPU Base and a 10 Slot EXP Base
8 Slot CPU Base and a 5 Slot EXP Base
5 Slot CPU Base and a 5 Slot EXP Base
5 Slot CPU Base and two 5 Slot EXP Base

Jumpers and Switches

10 Slot Base

Jumper Blocks
SW1 -> EXP/CPU - Located between Slot 0 and 1 of a CPU Base
SW2 -> 700/100 - Located between Slot 7 and 8 of a CPU Base

Switches
None

8 Slot Base

Jumper Blocks
None

Switches
None

5 Slot Base

Jumper Blocks
None

Switches
Base 1,3/Base 2 - Located between Slot 2 and 3 of a CPU Base

In the following diagrams, slots which have both a high byte and a low byte in the I/O register reference may contain either 8 or 16 point I/O modules. If an 8 point module is used, the high byte I/O register reference is not used. If an I/O register contains only a low byte reference then only an 8 point module may be installed in that slot. All analog modules must be installed in a slot with both a high byte and a low byte I/O register reference (16 point slot).

10 Slot CPU Base Addressing

```
  8   7   6   5   4   3   2   1   0   C
                                      P          10 Slot CPU
 700/100                    EXP/CPU   U

                                      I/O Register
 na  15  13  11   9   7   5   3   1   High Byte
 16  14  12  10   8   6   4   2   0   Low Byte
```

8 Slot CPU Base Addressing

```
          6   5   4   3   2   1   0   C
                                      P          8 Slot CPU
                                      U

                                      I/O Register
         13  11   9   7   5   3   1   High Byte
         12  10   8   6   4   2   0   Low Byte
```

5 Slot CPU Base Addressing

```
                      3   2   1   0   C
                                      P          5 Slot CPU
                     1,3/2            U

                                     I/O Register
                      7   5   3   1   High Byte
                      6   4   2   0   Low Byte
```

10 Slot CPU and 5 Slot Expansion Base

10 Slot CPU

8	7	6	5	4	3	2	1	0	CPU
700/100							EXP/CPU		
									I/O Register
na	15	13	11	9	7	5	3	1	High Byte
16	14	12	10	8	6	4	2	0	Low Byte

5 Slot EXP

4	3	2	1	0	
1,3/2					
					I/O Register
na	na	na	na	na	High Byte
21	20	19	18	17	Low Byte

10 Slot CPU and 10 Slot Expansion Base

10 Slot CPU

8	7	6	5	4	3	2	1	0	CPU
700/100							EXP/CPU		
									I/O Register
na	15	13	11	9	7	5	3	1	High Byte
na	14	12	10	8	6	4	2	0	Low Byte

10 Slot EXP

9	8	7	6	5	4	3	2	1	0	
700/100							EXP/CPU			
										I/O Register
na	na	na	na	na	na	na	na	na	na	High Byte
22	21	20	19	18	17	16	na	na	na	Low Byte

8 Slot CPU and 5 Slot Expansion Base

8 Slot CPU

6	5	4	3	2	1	0	CPU

							I/O Register
13	11	9	7	5	3	1	High Byte
12	10	8	6	4	2	0	Low Byte

5 Slot EXP

4	3	2	1	0

1,3/2

					I/O Register
na	na	na	na	na	High Byte
20	19	18	17	16	Low Byte

5 Slot CPU Base and 5 Slot Expansion Base

5 Slot CPU

3	2	1	0	CPU

1,3/2

				I/O Register
7	5	3	1	High Byte
6	4	2	0	Low Byte

5 Slot EXP

4	3	2	1	0

1,3/2

					I/O Register
na	na	na	11	9	High Byte
na	13	12	10	8	Low Byte

5 Slot CPU Base and Two 5 Slot Expansion Bases

```
 3   2   1   0   C
                 P          5 Slot CPU
1,3/2            U

                I/O Register
na  na  na  na   High Byte
 6   4   2   0   Low Byte


 4   3   2   1   0
                            5 Slot EXP
1,3/2

                    I/O Register
na  15  13  na  na  High Byte
 1  14  12  10   8  Low Byte


 4   3   2   1   0
                            5 Slot EXP
1,3/2

                    I/O Register
na  na  na  na  na  High Byte
11   9   7   5   3  Low Byte
```

APPENDIX D: DYNAMIC Protocol Summary

This appendix contains a summary of the DYNAMIC implementation on the F3-RTU. For a complete description of the protocol, please refer to the Texas Instruments "DYNAMIC PROTOCOL REFERENCE GUIDE, Designing SCADA Systems, December 1986".

TRANSACTION CONTROL FIELDS

TCF	Description
0	EXECUTE
1	DATA REQUEST
2	SEQUENCED OUTPUT
3	DIRECT OUTPUT
4	RESERVED
5	COMMAND/CONTROL
6	RESERVED
7	NACK
8	RESERVED
9	NOT SUPPORTED (EXTENDED DATA REQUEST)
10	RESERVED
11	RESERVED
12	NOT SUPPORTED (USER PROCESSING)
13	RESERVED
14	RESERVED
15	RESERVED

Dynamic Data Types for DATA REQUEST, TCF=1

Data Type	Description
0	RESERVED
1	ANALOG INPUT, RAW COUNTS
2	DISCRETE INPUT
3	ANALOG AND DISCRETE INPUT (1+2)
4	PULSE COUNT INPUT
5	PULSE DURATION INPUT
6	NOT SUPPORTED (PULSE PERIOD INPUT)
7	RESERVED
8	NOT SUPPORTED (HIGH SPEED COUNTER)
9	ALL INPUT DATA (1+2+4+5)
10	RESERVED
11	DISCRETE OUTPUT
12	NOT SUPPORTED (PULSE OUTPUTS)
13	ANALOG OUTPUT, RAW COUNTS
14	RESERVED
15	RESERVED

Both exception and full mode reporting is supported for all data types. Exception mode responses are optimized by the RTU to minimize the response message. The RTU will report all exception data using full mode format if the resulting message size for that data type is reduced.

Dynamic Data Types for DIRECT/SEQUENCED OUTPUT, TCF=0,2,3

Data Type	Description
0-10	ILLEGAL
11	DISCRETE OUTPUT
12	NOT SUPPORTED (PULSE OUTPUTS)
13	ANALOG OUTPUT, RAW COUNTS
14	RESERVED
15	RESERVED

Dynamic Function Codes for COMMAND/CONTROL, TCF=5

Function	Description	
00	RESERVED	
01	NOT SUPPORTED (CONFIGURE/ABSOLUTE)	
02/03	LOGICAL CHANNEL INPUT/OUTPUT	
	Channel Number	Description
	01-15	RESERVED
	16	BASIC USER PROGRAM 0
	17	FOR BASIC USER PROGRAM 1
	18-23	FOR 512K USER MEMORY, 64K EACH LC
	24	RESERVED
	25	FOR STATUS AND ERRORS
	26	FOR PID LOOP PARAMETERS
	27-63	RESERVED
	64-75	FOR HISTORICAL DATA/DATA LOGGING
	76-FF	RESERVED
04	NOT SUPPORTED	(COUNTER FREEZE/RESET)
05	NOT SUPPORTED	(COUNTER THAW/RESET)
06	NOT SUPPORTED	(CONFIGURE/MODIFY) CONFIGURATION IS DONE BY BASIC
07	RESET LATCHED STATUS BITS IN RTU STATUS WORD	
08/09	START/KILL TASK	
	Job Name	Description
	Bx	BASIC program number "x"
	Lx	RESERVED (FOR Ladder Logic program number "x")
	Px	RESERVED (FOR PID Loop number "x")
0A	NOT SUPPORTED (SET REAL TIME CLOCK)	
0B	NOT SUPPORTED (READ REAL TIME CLOCK)	
0C	READ HARDWARE CONFIGURATION	
0D	NOT SUPPORTED (READ SOFTWARE CONFIGURATION)	
0E	NOT SUPPORTED (ISSUE DAYFILE MESSAGE)	
0F	NOT SUPPORTED (PRINT STRING TO RTU SERIAL I/O PORT)	
10	NOT SUPPORTED (ASSUME DEFAULT CONFIGURATION)	
11	RESERVED	
12	RESET RTU	
13	NOT SUPPORTED (SEND MESSAGE)	
14	NOT SUPPORTED (WAIT/RETRIEVE MESSAGE)	
15	NOT SUPPORTED (CONFIGURE/ABSOLUTE)	
16	NOT SUPPORTED (SET POUT POLARITY)	
17	NOT SUPPORTED (SET INTERLOCK)	
18	NOT SUPPORTED (CLEAR INTERLOCK)	
19	NOT SUPPORTED (REQUEST DATA BASE ACCESS TOKEN)	
1A	NOT SUPPORTED (SKIP ON ERROR)	

1B	NOT SUPPORTED (RETURN DATA BASE ACCESS TOKEN)
1C	READ RTU STATUS WORD
1D	NOT SUPPORTED (ESCAPE TO DIRECT OUTPUT)
1E	NOT SUPPORTED (ESCAPE TO DIRECT REQUEST)
1F	RESERVED
20	NOT SUPPORTED (SET FILTERS/PROCESSING OPTIONS)
21	NOT SUPPORTED (SOFTWARE INTERRUPT)
22	NOT SUPPORTED (REQUEST BROADCAST ACKNOWLEDGEMENT)
23	NOT SUPPORTED (WRITE CONFIGURATION MARK)
24	NOT SUPPORTED (READ CONFIGURATION MARK)
25	NOT SUPPORTED (TEST INTERLOCK)
26	NOT SUPPORTED (START CAP)

RTU Status Word

Bit	Description	
0	1 = RTU USER MEMORY BATTERY BACK-UP LOW	
1	ALWAYS 0	(RESERVED)
2	ALWAYS 1	(AC POWER PRESENT)
3	ALWAYS 0	(EXTERNAL ALARM PRESENT)
4	1 = BROADCAST MESSAGE ACKNOWLEDGE	
5	ALWAYS 0	(SOFTWARE CONFIGURATION PRESENT)
6	ALWAYS 0	(OUTPUT ACKNOWLEDGE)
7	1 = RTU RESET (POWER-UP FLAG)	
8	ALWAYS 0	(COUNTERS FROZEN)
9	1 = SEQUENCED OUTPUT ACKNOWLEDGED	
10	1 = MORE DATA AVAILABLE	
11	ALWAYS 0	(POINT DATA ADDRESSING MODE, PDAM = PHYSICAL)
12	ALWAYS 1	(MODE 40/41 = 8641 EMULATION)
13	ALWAYS 0	(BOARD TYPE DESCRIPTION, M057/M057)
14	ALWAYS 0	(SBX-B PRESENT)
15	ALWAYS 0	(CARD DATA ADDRESSING MODE, CDAM = PHYSICAL)

NACK and COMMAND/CONTROL Error Codes

Error Code	Description
01	INVALID BYTE COUNT
02	UNKNOWN DATA TYPE
03	ILLEGAL SLOT NUMBER (ONLY 15 SLOTS SUPPORTED)
04	UNKNOWN POINT
06	UNKNOWN TCF
07	MESSAGE TOO LONG
08	PARAMETER OUT OF RANGE
0A	BAD COMMAND ALIGNMENT - ONLY A SINGLE BYTE WAS FOUND WHERE A FUNCTION CODE WORD (COMMAND/CONTROL MESSAGE) WAS EXPECTED
0B	ILLEGAL FUNCTION CODE
0C	OUT OF DATA
0D	CONFLICTING CONFIGURATION
0E	INVALID LOGICAL CHANNEL LENGTH
0F	INVALID I/O PORT
10	PREVIOUS I/O REQUEST INCOMPLETE (CCM FUNCTION CODE 15H)
11	MISSING 8087
12	SYSTEM HALTED
13	SKIP
14	ILLEGAL INTERRUPT
15	ILLEGAL USER FUNCTION PROCESSOR
17	TSN MISMATCH
30	EXECUTE WITHOUT SEQUENCED OUTPUT PENDING
31	EXECUTE TSN WAS INCONSISTENT WITH THAT OF A PENDING SEQUENCED OUTPUT
32	EXECUTE TWO-SECOND TIME-OUT EXCEEDED
33	REDUNDANT EXECUTE

DYNAMIC SPECIFIC SYSTEM CODES

The following BASIC SYSTEM statement codes are unique to the DYNAMIC protocol implementation on the F3-RTU.

These codes are primarily used to configure features unique to the FACTS RTU. For example, the Un-solicited Report By Exception (URBE) feature built in to the DYNAMIC scanner and protocol driver is not available on the Texas Instruments RTUs. Please contact FACTS Engineering for additional information and help on implementing these new features.

Syntax SYSTEM (*code*) = *expr*
var = SYSTEM(*code*)

code	DESCRIPTION
10	DYNAMIC RECEIVE, WAIT FOR LAST CHARACTER TIME LIMIT COUNT, 10 MSEC PER COUNT. DEFAULT IS 3 CHAR. TIMES.
11	RETURNS DYNAMIC DEVICE ADDRESS PROGRAMMED USING THE J2 JUMPERS (SEE PAGE 18).
12	MAXIMUM NUMBER OF URBE COMMUNICATION TIRES PERMITTED. DEFAULT IS 3.
13	NUMBER OF URBE COMMUNICATION TIRES THIS TRANSACTION.
15	URBE COMMUNICATION TIME-OUT IN 10 MSEC UNITS 655.35 TO .01 SEC. DEFAULT IS 3 SEC.
16	TICKS, NUMBER OF 10 MSEC INTERRUPTS BEFORE NEXT I/O UPDATE AND DYNAMIC MESSAGE PROCESSING.
17	URBE_ENABLE, ENABLE URBE
18	URBE_GLOBAL_FLAG, URBE GLOBAL FLAG
19	SET FOR ABM ON PORT 1, CLEAR FOR DYNAMIC ON PORT 1 (PORT 1 RETURNS TO DYNAMIC PORT AFTER POWER CYCLE)
20	START OF SLOT 0 DATA BASE
21	MSB CH 1 AIN RBE DEAD BAND LOCATION OFFSET FROM SYSTEM(20)
22	MSB CHANNEL 1 AIN/AOUT LOCATION OFFSET FROM SYSTEM(20) (IF AIN THEN 8TH BIT = RBE, 6TH BIT = URBE ENABLE. IF AOUT THEN 5TH BIT = NEW DATA FLAG FOR I/O UPDATE)

23 MSB RBE ENABLE LOCATION OFFSET FROM SYSTEM(20). URBE enable location, ENURBELOC = ENRBE+2

24 NON-VOLATILE COUNTER FOR DYNAMIC RECEIVED PARITY ERRORS. MUST BE CLEARED BY BASIC.

25 NON-VOLATILE COUNTER FOR DYNAMIC RECEIVED FRAME CHECK ERRORS. COUNTS EITHER BCC OR CRC-16 ERRORS. MUST BE CLEARED BY BASIC.

26 NON-VOLATILE COUNTER FOR NUMBER OF INCOMPLETE DYNAMIC MESSAGES RECEIVED. SYSTEM(10) = TIME ALLOWED BEFORE A FRAME IS ABORTED. MUST BE CLEARED BY BASIC.

27 NON-VOLATILE COUNTER FOR DYNAMIC URBE MESSAGES WHICH WHERE ABORTED DUE TO NO HOST RESPONSE. SYSTEM(15) = TIME BETWEEN RETRIES. SYSTEM(12) = NUMBER OF RETIRES BEFORE ABORTING THE URBE MESSAGE. MUST BE CLEARED BY BASIC.

28 SPECIFY THE DESTINATION ADDRESS FOR URBE MESSAGE. DEFAULT IS 0.

29 EXTERNAL MEMORY MAP LOOK-UP TABLE IN CODE MEMORY. SLOT 0 DATA BASE LOCATIONS BYTE OFFSETS FROM SYSTEM(20) (OTHER SLOTS = SLOT 0 + 100H x SLOT_NUMBER). READ ALL MEMORY ADDRESSES EXCEPT PULSE RATE CONFIGURATION USING CBY(SYSTEM(29)+OFFSET). THE LOCATION OF THE PULSE RATE CONFIGURATION VALUES (PPR, PINP2, AND PRIMN) ARE READ AT SLOT 0 + 100H x (SLOT_NUMBER+16).

	OFFSET	
HIGH XCONF	0	
ENRBE	1	
PULDIR	2	
CNTSTAT	3	
PULSEDUR	4	
ACNT	5	
PINP	6	
LOWDATA	7	
HIGHDATA	8	
URBE	9	UNSOLICITED RBE ENABLE
TSRBE	10	TIME STAMP RBE ENABLE
TSURBE	11	TIME STAMP UNSOLICITED RBE ENABLE
PPR	12	PULSE RATE RBE ENABLE
PINP2	13	PULSE RATE COUNT DEAD BAND
PRIMN	14	PULSE RATE INPUT MINIMUM
ANSR	15	ANALOG AVERAGING SAMPLE TIME
ASUM	16	ANALOG ACCUMULATION VALUE

30 RETURNS BEGINNING OF USER DEFINED LOGICAL CHANNEL CONFIGURATION TABLE LOCATED IN SRAM BANK 1 (WORD(65536) TO WORD(65536*2)). THE CONFIGURATION TABLE MAY HAVE 0 TO 50 ENTRIES BEGINNING AT SYSTEM(30). EACH ENTRY CONTAINS THE FOLLOWING THREE ITEMS (5 BYTES/ENTRY).

BEGINNING ADDRESS OF THE LOGICAL CHANNEL (WORD)
SIZE OF THE LOGICAL CHANNEL IN BYTES (WORD)
LOGICAL CHANNEL NUMBER (BYTE)

31 ENABLE/DISABLE DYNAMIC HARDWARE HANDSHAKING. WHEN ENABLED DYNAMIC WILL ASSERT RTS AND WAIT UNTIL UP TO A WATCH-DOG TIME-OUT (~1 SEC.) FOR CTS BEFORE TRANSMITTING DATA. THERE IS NO RTS OFF DELAY AT THE END OF THE TRANSMISSION. DEFAULT IS ENABLED

32 EXTERNAL MEMORY MAP LOOK-UP TABLE IN CODE MEMORY FOR THE FOLLOWING ITEMS. CBY(SYSTEM(32)+OFFSET) RETURNS ABSOLUTE MEMORY ADDRESS. ADDRESS IS FOR MSB OF DATA. DATA IS STORED IN MEMORY MSB:LSB WITH THE LSB AT THE HIGHEST MEMORY ADDRESS.

	OFFSET	NUMBER OF BYTES
HOLDTXD	0	2
KEEPALIVE	2	2
HOSTSTAT	4	2
SOURCE_LAN_ADDR	6	4
DEST_WAN_ADDR	8	10
SOURCE_WAN_ADDR	10	10
TIME_TO_LIVE	12	2
SECOND_OF_DAY	14	3
PACKET_TIME	16	4
UPDATE_SOD	18	1
TURN_AROUND_DELAY	20	1
LCSTAT	22	2

HOLDTXD — READ HOLD TRANSMIT TIME FOR URBE. THIS IS THE TIME IN 100 msec UNITS THAT BASIC SHOULD WAIT BEFORE SENDING AN URBE MESSAGE BACK TO THE HOST.

KEEPALIVE — KEEP ALIVE TIME FOR URBE. THIS IS THE MAXIMUM TIME IN 100 msec UNITS THAT BASIC CAN WAIT BEFORE SENDING AN URBE MESSAGE.

HOSTSTAT — HOST STATUS WORD.

SOURCE_LAN_ADDR	FOUR BYTE LAN ADDRESS OF THE NODE FROM WHICH A FRAME IS BEING SENT. DEFAULT = 32 28 77 88.
DEST_WAN_ADDR	TEN BYTE WAN ADDRESS OF THE DEVICE TO WHICH THE FRAME IS BEING SENT. DEFAULT = ALL ZEROS.
SOURCE_WAN_ADDR	TEN BYTE WAN ADDRESS OF THE DEVICE FROM WHICH THE FRAME WAS SENT. DEFAULT = ALL ZEROS.
TIME_TO_LIVE	TWO BYTE TIME TO LIVE IN LAYER 3. DEFAULT = 0.
SECOND_OF_DAY	THREE BYTE TIME SINCE MIDNIGHT OF CURRENT DAY IN SECONDS.
PACKET_TIME	FOUR BYTE TIME SINCE MIDNIGHT OF JANUARY 1, 1970 MAINTAINED BY PACKET RADIO NETWORK. BASIC CAN WRITE ZEROS TO THESE LOCATIONS IN ORDER TO SEE WHEN THE RTU HAS RECEIVED A NEW TIME-STAMP.
UPDATE_SOD	SECOND OF DAY CONTROL BYTE. UPDATE_SOD = 1, RTU HAS WRITTEN NEW PACKET_TIME UPDATE_SOD = 4, FREEZE PACKET_TIME UPDATE_SOD = 8, RTU MAKES SECOND_OF_DAY = TIME$
TURN_AROUND_DELAY	TURN ARROUND DELAY VALUE (TADV) FOR PACKET RESPONSE MESSAGES. DEFAULT IS 1, NO TURN ARROUND DELAY. DELAY = (TADV-1) x 10 x TICKS msec.
LCSTAT	TELLS BASIC ABOUT LOGICAL CHANNELS WRITTEN TO BY THE HOST. MSB = LAST CHANNEL NUMBER WRITTEN LSB, BIT 0 = WRITE COMPLETED (CLEARED BY BASIC) LSB, BIT 1 = LAST WRITE STATUS OVER-WRITTEN (BIT 0 WAS STILL SET)

Example:

```
1000   S32=SYSTEM(32)
1010   PICK(DWAN,H)=CBY(S32+8) : PICK(DWAN,L)=CBY(S32+9)
1020   FOR I=0 TO 9
1030   BYTE(DWAN+I)=I
1040   NEXT I
1050   PRINT2 "DESTINATION WAN ADDRESS SET TO: ",
1060   FOR I=0 TO 9
1070   PRINT2 "0",I, SPC (1),
1080   NEXT I

>RUN
DESTINATION WAN ADDRESS SET TO: 00 01 02 03 04 05 06 07 08 09
```

APPENDIX E: Modbus RTU Protocol

This document describes the FACTS Engineering Modbus RTU protocol[1] support provided for the F3-RTU.

The hardware requirements are:

1) One F3-RTU.

2) The FACTS Engineering CPUMOD.ABM program. Minimal BASIC programming changes are required to change default operating parameters.

3) Any valid combination of PLC ***Direct*Ô** and FACTS discrete and analog modules may be used.

F3-RTU PROGRAMMABLE CPU RESOURCE REQUIREMENTS

The following F3-RTU resources are used by the Modbus slave program. All additional resources are available to the user's optional BASIC program.

Serial Ports - 1   Program Memory - 6K   Data Memory - 1.5K

Installation:

1) Set configuration parameters in the ASCII text file CPUMOD.ABM.

2) Install the F3-RTU in the CPU slot of the DL305 base. Download the program CPUMOD.ABM using ABM Commander. The CPUMOD.ABM file is located in the ABM60\CPU-RTU directory that ABM Commander 6.xx creates.

[1] Based on Gould document PI-MBUS-300 Rev B

F3-RTU JUMPER CONFIGURATION

The configuration jumpers shown in Chapter 5 for DYNAMIC Remote Address and Protocol Select only apply to the DYNAMIC protocol built in to the F3-RTU firmware. The position of these jumpers will not effect the operation of the BASIC Modbus program.

MODBUS CONFIGURATION

All configuration data is read from the first section of code in the CPUMOD.ABM program. In order to change these parameters, use an ASCII text editor to modify these lines. The affected lines are shown below with the defaults.

```
2    P1=0 : REM          Modbus Port is Port 2
3    DEBUG=NOT(0) : REM  Debug Mode is ON
4    RA=1 : REM          Remote Station Address is 1
5    BAUD=9600 : REM     Baud rate of Modbus Port is 9600 baud
6    MDROP=0 : REM       Multidrop is OFF
7    RTSCTS=0 : REM      Hardware Handshaking is OFF
8    WFL=10 : REM        Wait for Last Character Time-Out is 10ms
9    AAR=0 : REM         Automatic Address Recognition is OFF
10   CN=0 : REM          Radio Keying ON Delay is 0ms
11   CF=0 : REM          Radio Keying OFF Delay is 0ms
```

Modbus Port

The Modbus Port is selected by variable P1. If assigned a value of NOT(0) the Modbus Port is Port 1. If assigned a value of 0 the Modbus Port is Port 2.

Debug Mode

Debug Mode can be enabled by assigning a value of NOT(0) to the variable DEBUG. The Debug mode can be disabled by assigning a value of 0 to the variable DEBUG. If enabled and the Modbus Port is Port 2 then all Modbus communications will be echoed out Port 1. In this mode all messages received and all replies transmitted are echoed out of Port 1 at 9600 baud (no parity, 8-bit word, 1 stop bit). Data bytes are displayed in hexadecimal. Received and transmitted messages are preceded by "RXD =" and "TXD =" respectively. This mode is only recommended during startup or troubleshooting.

Remote Station Address

The Modbus Remote Station Address from 1 to 247 is read from the variable RA. If an invalid address is found the remote address will default to 1.

Communication Rate

The Communication Rate for the Modbus communication port is read from the variable BAUD. If an invalid code is found the baud rate will default to 1200. The maximum baud rate for Port 2 is 9600.

RS422/485 Multidrop

RS422/485 Multidrop is enabled by assigning the value NOT(0) to the variable MDROP. RS422/485 Multidrop is disabled by assigning the value 0 to the variable MDROP. If enabled the RS422/485 transmitters will only be enabled when a Modbus response is sent. Turn this feature ON for for hardwired multi-drop network configurations.

Hardware Handshaking

Hardware Handshaking is enabled by assigning the value NOT(0) to the variable RTSCTS. Hardware Handshaking is disabled by assigning the value 0 to the variable RTSCTS. If enabled then RTS of the Modbus Port will be asserted when the F3-RTU Modbus is ready to transmit a message. The F3-RTU will not transmit a message until CTS is asserted on the Modbus Port.

If disabled then RTS of the Modbus Port will come on for the time specified by CN before sending a Modbus response. RTS will stay on for the time specified by CF after sending a Modbus response.

Wait for Last Character Time-Out

The variable WFL is the Wait for Last Character Time-Out. This is useful for adjusting out radio noise when the carrier is dropped. This variable is in increments of 10 milliseconds. If zero is in this location the Wait for Last Character Time-Out will default to 10 milliseconds.

Automatic Address Recognition

Automatic Address Recognition is enabled by assigning the value NOT(0) to the variable AAR. Automatic Address Recognition is disabled by assigning the value 0 to the variable MDROP. Valid for Port 1 only. If enabled it forces an 11-Bit data frame with an 8-bit word and 1 stop bit. The 9th parity bit is used to distinguish between address and data bytes. The hardware checks this bit, if it is a 1 and the address byte matches the subsequent data bytes are loaded and processed.

Use this feature when all remotes in the network are FACTS engineering modules and the host supports AAR. The FACTS Modbus Master Module F4-MAS-MB supports the AAR feature.

Radio Keying Delays

The model F3-RTU Modbus program built-in radio keying relay on-delay and off-delay times are read from the variables CN and CF respectively.

RTS of the Modbus Port will come on for the time specified by CN before sending a Modbus response. RTS will stay on for the time specified by CF after sending a Modbus response.

ADDRESSING

The Modbus host reads and writes the F3-RTU Modbus Program using the Modbus addresses shown in the "F3-RTU Modbus Program Address Map" on the next page.

For Modbus register data types (REG.) the values shown are the range of valid starting and ending 16-bit register values as they appear in the Modbus protocol message.

For Modbus discrete coil data types (COIL) the values shown are the range of valid starting and ending discrete point values as they appear in the Modbus protocol message.

When using a host computer package which has a built-in Modbus driver or server, you may need to offset the address range for the particular Modicon PLC configured for the server.

The table below shows the starting address offset required when a typical server is configured for either a 484 or 584/984 type PLC.

Starting Address Offset for Modicon PLCs

Data Type	Starting 484 Address	Starting 584 984 Address	Modbus Message Address
Read/Write Coil	1	1	0
Read Only Coil	1001	10001	0
Read/Write Reg.	4001	40001	0

MEMORY MAP

F3-RTU Modbus Program Address Map:

F3-RTU Data Type	Quantity	Modbus Data Type	Modbus Addr. Range	Function Code
Discrete Inputs	184	COIL	0-B8(Hex) 0-183	2
Discrete Outputs	184	COIL	0-B8(Hex) 0-183	1,2,5,15
Analog Inputs	128	REGISTER	0-80(Hex) 0-127	3,4
Analog Outputs	128	REGISTER	0-80(Hex) 0-127	3,4,6,16

Each Address for the Coil Data type represents one discrete input or discrete output.

Each Address for the Register Data Type represents one Analog Input or Analog Output.

The F3-RTU Modbus Program distinguishes between inputs and outputs. If a Modbus write request attempts to write to an input, the Modbus program will return an exception code 02 (Illegal Data Address). You can load and run the ABM program IOCFG to verify your I/O configuration.

Jumper settings distinguish between Discrete and Analog Slots.

See Appendix C for base configurations and addressing.

Supported Modbus Function Codes

Code	Meaning	Data Type
01	Read group of inputs or outputs	Discrete Inputs and Outputs
02	Read group of inputs or outputs	Discrete Inputs and Outputs
05	SET/RST single output	Discrete Outputs
15	SET/RST group of outputs	Discrete Outputs
03	Read HEX value of 1 or more Analog Inputs or Outputs	Analog Inputs and Outputs
04	Read HEX value of 1 or more Analog Inputs or Outputs	Analog Inputs and Outputs
06	Write HEX value to a single Analog Output	Analog Outputs
16	Write HEX to a group of Analog Outputs	Analog Outputs
22	Read from ABM memory	Bank 1
23	Write to ABM memory	Bank 1
26	Read or Write TIME and DATE	

ADDITIONAL FUNCTION CODES

FACTS Engineering has defined additional function codes for setting the ABM Clock Calendar, Reading ABM Memory, and Writing ABM Memory. These function codes were added to enable historical data logging, time stamping and other user BASIC programmable features. In order to use these function codes the host software should implement them as described below.

When using these function codes the Modbus program should be run out of PRM0 (BANK0 Memory). Data will be read and written from bank 1 memory.

FUNCTION CODE 22 - Read from ABM memory (Bank 1)

This function code can be used to read data that the user's BASIC program has written to Bank 1 memory.

A data start address of 0 will access the first location in Bank 1 memory which is WORD(65536). The ending address is equal to (Starting address + Number of registers - 1)*2.

The Modbus addresses for Bank 1 ABM memory are 0-65535. Each address represents a BYTE of memory for a total of 64K bytes.

If the number of registers is greater than 125 then the slave will return an illegal data address exception code (02). If the Ending address is greater than 65535 DEC (FFFF HEX), then the slave will send the illegal data address exception code (02).

ADDR	FUNC CODE	H.O. ADDR	L.O. ADDR	H.O. # OF REGS	L.O. # OF REGS	CRC
01	16	00	FF	00	02	CRC

The above example is used to read 2 words or 4 bytes starting with WORD(512+65536) through WORD(514+65536).

ADDR	FUNC	BYTE CNT	DATA REG H.O.	DATA REG L.O.	DATA REG H.O.	DATA REG L.O.	CRC
01	16	04	29	2E	4F	52	CRC

The chart above is an example of the Response String if there was a value of 10542 in WORD(512) and 20306 in WORD(514).

FUNCTION CODE 23 - Write to ABM Memory (Bank 1)

This function code can be used to write data to Bank 1 memory where a user's BASIC program can use it.

A data start address of 0 will access the first location in Bank 1 memory which is WORD(65536). The ending address is equal to (Starting address + Number of registers - 1)*2.

The Modbus addresses for Bank 1 ABM memory are 0-65535 which corresponds to WORD(65536) to WORD(131072). Each address represents a BYTE of memory for a total of 64K bytes.

If the number of registers is greater than 125 then the slave will return an illegal data address exception code (02). If the Ending address is greater than 65535 DEC (FFFF HEX), then the slave will send the illegal data address exception code (02).

```
A   F  HO   LO  NUMBER BYT HO  LO  HO  LO   C
D   U   A   A     OF   CNT  D   D   D   D   R
D   N   D   D    REGS       A   A   A   A   C
R   C   D   D               T   T   T   T

01  18  00  FF  00  02  04  04  57  08  AE CRC
```

Fig. 1 Command String

The above Command String will set WORD(512) to 1111 and WORD(514) to 2222.

NOTE: Writing to ABM memory may overwrite a stored programs in the module. Make sure all programs are backed-up on floppy or hard disks.

```
ADDR FUNC H.O. L.O.   NUMBER   CRC
          ADDR ADDR     OF
                       REGS.

 01   18   00   FF   00   02   CRC
```

The Response String above will echo the Address, Function Code, Starting Address high and low order and the Number of registers.

FUNCTION CODE 26 (Set date and time)

ADDR	FUNC CODE	RD/ WRT	MTH	DAY	YEAR	DAY OF WEEK	HOUR	MIN.	SEC.	CRC
01	1A	FF	2	A	5D	3	D	19	1E	CRC

The above Command String will set the date to Feb. 10, 1993 and the time to 3:25:30 p.m. The time is set in 24 hour time. (13=1 p.m. 14=2 p.m. etc..). If the RD/WRT field is FF then the time and date will be written to the BASIC module. If it is 00 then the time and date will only be read and put into the Response String and the trailing data in the Command String will be ignored.

The limitations on this command are defined by the TIME$ and DATE$ statements. If you send a command to set the Month to 13, the response will be an exception code. If the Command is Global, address of 00, the module will set the time and date but will not send a response.

ADDR	FUNC CODE	RD/ WRT	MTH	DAY	YEAR	DAY OF WEEK	HOUR	MIN.	SEC.	CRC
01	1A	FF	2	A	5D	3	D	19	1E	CRC

If you are writing the response string will return the same values as the Command String. If you are reading then the values will be that of the current date and time.

FAST RESPONSE FEATURE

The F3-RTU Modbus program, CPUMOD.ABM, includes a fast response feature which in most applications, reduces the time required to poll a network. The Fast Response mode is automatic. No user programming is required.

Fast Response Feature, Description of Operation:

The module automatically remembers the locations read during the last valid read block poll, function codes 3 or 4. While other RTUs are being polled, the module continuously reads these locations. If another read block poll message is received by the module for the same locations, then the module immediately sends this information back to the SCADA master. In the case where the same block of memory is always read from the F3-RTU, the module's response will have a turn-around delay time of 0.

F3-RTU MODBUS PROGRAM RESPONSE TIME

BASIC CoProcessor Modbus response time can be estimated using the following formula.

DB = Data Bytes = number of I/O points to R/W divided by 8
                  number of registers to R/W times 2
ML = Message length = 13 + DB
CT = Communication Time (in seconds) = 10/baud rate x ML
AT = Access Time (in seconds) = PLC scan time
KT = Keying Time (in seconds) = RTS on-delay + RTS off-delay
CC = CoProcessor Compilation Time
RT = Response Time = KT x 2 + AT + CT + CC

Examples:

RS-422 network at 57600 baud, KT=0, 16 I/O R or W, 4 register R or W.

For 15 msec scan time, RT = 15 + 1.7 + 5 = 22 msec

Radio network at 1200 baud, KT=.015, 16 I/O R or W, 4 register R or W.

For 15 msec scan time, RT = .03 + 15 + 83 + 5 = 103 msec

APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

RS-232 STANDARD

RS-232-C (RS-232) is an interface standard from the Electronic Industries Association (EIA). The standard names and defines 20 communication signals, assigned to separate pins in a 25-pin connector. The five unassigned pins may carry nonstandard signals required by any individual system.

Each signal is transmitted as a positive or negative electric current between 3 and 15 volts (usually 12 volts). The signal assigned to each pin flows in one direction only. Signals **output**, for example, from a computer must **input** to a terminal, and vice versa.

RS-232 signals travel over a serial interface cable that may have up to 25 wires. Since most signals are not required for simple communication, cables have as few as 2 or 3 wires. As shown in the following cabling diagrams, jumpers often are installed at one or both of the connectors to ensure that flow control signals are satisfied.

The signals flow between two types of interface ports, data communication equipment (DCE) and data terminal equipment (DTE). The pin names are the same for both DCE and DTE equipment, however, the direction of signal flow is reversed.

RS-232 DTE and DCE Pin Names and Signal Flow

			Signal Direction		
Pin	Abrev.	Name	DCE	DTE	Description
1	FG	Frame Ground	None	None	
2	TXD	Transmit Data	Input	Output	DTE output data path
3	RXD	Receive Data	Output	Input	DCE output data path
4	RTS	Request to Send	Input	Output	DTE has data to XMIT
5	CTS	Clear to Send	Output	Input	DTE may XMIT data
6	DSR	Data Set Ready	Output	Input	DCE has data to XMIT
7	SG	Signal Ground	None	None	
8	DCD	Data Carrier Detect	Output	Input	Modem has carrier
20	DTR	Data Terminal Ready	Input	Output	DCE may XMIT data
22	RI	Ring Indicator	Output	Input	

IBM COMPUTER CABLES

```
   IBM AT                        F3-RTU
        PIN                     PIN

   TXD   2  <                    2   TXD
   RXD   3                    >  3   RXD
   RTS   7
   CTS   8
   GND   5                       7   GND
   DCD   1
   DTR   4
   DSR   6

  9-PIN MALE                   9-PIN MALE
  DCE CONNECTOR                DTE CONNECTOR


IBM PC, PC-XT                    F3-RTU
        PIN                     PIN

    RI   22
   TXD   2                    >  3   RXD
   RXD   3  <                    2   TXD
   RTS   4
   CTS   5
   GND   7                       7   GND
   DSR   6
   DCD   8
   DTR   20

 25-PIN FEMALE                  9-PIN MALE
 DTE CONNECTOR                 DTE CONNECTOR
```

IDENTIFYING A COMMUNICATION PORT AS DCE OR DTE

With an unknown RS-232 port powered, measure the dc voltage between pin-2 and ground (pin-7) and pin-3 and ground. If the most negative pin is pin-2 then the port is DTE. If the most negative pin is pin-3 then the port is DCE. Improper connection of pins 2 and 3 will not damage the interface.

RS-232 DCE CONNECTION WITH SOFTWARE OR NO HANDSHAKE

```
F3-RTU MODULE, DTE                     DCE DEVICE
               PIN                     PIN

     TXD   2                         >   2   TXD
     RXD   3  <                          3   RXD
                                         4   RTS
                                         5   CTS
     GND   7                             7   GND
                                         6   DSR
   9-PIN FEMALE                         20   DTR
   CONNECTOR
                                      TYPICAL 25-PIN
                                      CONNECTOR
```

RS-232 DTE CONNECTION WITH SOFTWARE OR NO HANDSHAKE

```
F3-RTU MODULE, DTE                     DTE DEVICE
               PIN                     PIN

     TXD   2                         >   3   RXD
     RXD   3  <                          2   TXD
                                         5   RTS
                                         4   CTS
     GND   7                             7   GND
                                         6   DSR
   9-PIN FEMALE                          8   DCD
   CONNECTOR                            20   DTR

                                      TYPICAL 25-PIN
                                      CONNECTOR
```

RS-232 WITH HARDWARE HANDSHAKE

```
F3-RTU MODULE, DTE          MODEM OR OTHER DCE DEVICE
                          REQUIRING HARDWARE HANDSHAKING

        PIN                    PIN

   TXD   2                 >   2   TXD
   RXD   3  <                  3   RXD
   RTS   4                 >   4   RTS
   CTS   5  <                  5   CTS
   GND   7                     7   GND
                               6   DSR
9-PIN FEMALE                  20   DTR
CONNECTOR
                            TYPICAL 25-PIN
                            CONNECTOR
```

```
F3-RTU MODULE, DTE         DTE DEVICE REQUIRING
                           HARDWARE HANDSHAKING
        PIN                    PIN

   TXD   2                 >   3   RXD
   RXD   3  <                  2   TXD
   RTS   4                 >   6   DSR
   CTS   5  <                 20   DTR
   GND   7                     7   GND
                               8   DCD
9-PIN FEMALE                   5   CTS
CONNECTOR                      4   RTS

                             TYPICAL 25-PIN
                             CONNECTOR
```

PARALLEL PRINTER CABLING

```
Parallel Printer     Serial/Parallel Converter

   Centronics          Centronics     DTR   20
    parallel            parallel      RXD   3  <
   interface           interface      GND   7

                 36-PIN
         Centronics Interface

                            F3-RTU Module

                               GND   7
                               TXD   2
                               CTS   5

                             9-PIN MALE
                             DTE CONNECTOR
```

RS-422/485 STANDARD

The RS-485 transceivers on F3-RTUs are compatible with both RS-422 and RS-485 signals.

RS-422 uses high current differential outputs and is specified to 4000 feet at 10 Megabaud. Lower speed communications, such as 19.2K baud, may use substantially longer cables.

RS-485 is an upgraded version of EIA RS-422-A and offers higher current tri-state drivers which are internally protected from bus contentions caused by multiple drivers on the same line. RS-485 drivers will also withstand higher voltages on their outputs when disabled (high impedance state). RS-485 is specified for multiple transmitter and multiple receiver systems as well as single and multi-drop RS-422 applications. The RS-422 specification permits only one driver and 10 receivers on a line. The RS-485 standard allows up to 32 drivers and receivers on the same transmission line.

RS-422/485 COMMUNICATION

Most CoProcessors have one RS-422/485 communication interface some have two. To select a port for RS232 or RS422/485 data reception mode, please refer to "JUMPER DESCRIPTIONS AND LOCATIONS" in the chapter for the CoProcessor module that you are using. Transmissions from a selectable port are always available at RS-232 and RS-422/485 signal levels simultaneously.

RS-422/485 POINT-TO-POINT CABLING

```
RS-422 or RS-485 DTE                F3-RTU
Terminal or BASIC Module          PIN  Port 1

        RXD (+)   <                 9   TXD+
        RXD (-)   <                 8   TXD-
        TXD (-)                 >   6   RXD-
        TXD (+)                 >   1   RXD+
        GND                         7   GND
```

RS-422/485 MULTI-DROP MADE EASY

Four wire RS-422 multiple transmitter multi-drop networks and all 2 wire RS-485 connections require that the transmitters float when not in use.

To enable the RS-422/485 transmitters only when PRINTing, use SETPORT to select multi-drop mode "M". Use the multi-drop option when the F3-RTU is a slave in a master/slave configuration or when a peer to peer configuration is required.

To leave the RS-422/485 transmitters ON even when not PRINTing, use SETPORT to select point to point mode "P". Use the point to point option when the F3-RTU is a single master in a master/slave or point to point configuration. This configuration provides the greatest noise immunity because the RS-422/485 drivers remain enabled and prevent noise from being received by the the slave devices on the network.

Example   Configure Port 1 for 9600 baud, no parity, 8 bit word, 1 stop bit, software XON/XOFF handshaking, and multi-drop RS-422/485 mode.

```
SETPORT 1, 9600, N, 8, 1, S, M
```

RS-485 TWO WIRE MULTI-DROP

```
    F3-RTU
RS-422/485 Port


  RTS    4
  TXD+   9           l       120 Ù
  TXD-   8         l
  RXD-   6  <      l
  RXD+   1  <        l       120 Ù     l
  GND    7       l
                               l

                               Ñ Only use one earth ground
                                 for Network signal common.

  F3-RTU
Port 1  PIN

  TXD+   9           l
  TXD-   8         l
  RXD-   6  <      l
  RXD+   1  <        l
  GND    7       l



  F3-RTU
Port 1  PIN

  TXD+   9           l
  TXD-   8         l
  RXD-   6  <      l
  RXD+   1  <        l
  GND    7       l



  F3-RTU
Port 1  PIN

  TXD+   9           l       120 Ù
  TXD-   8         l
  RXD-   6  <      l
  RXD+   1  <        l       120 Ù      l
  GND    7       l
```

RS-422/485 FOUR WIRE MULTI-DROP

```
    F3-RTU
  RS-422/485 Port
                                62 Ù

  RTS    4
  TXD-   8                      62 Ù       l
  TXD+   9
  RXD-   6  <          l
  RXD+   1  <            l
  GND    7       l

                                  Ñ Only use one earth ground
                                    for Network signal
                                    common.

  F3-RTU
Port 1  PIN

  RXD-   6         l
  RXD+   1           l
  TXD-   8  <          l
  TXD+   9  <            l
  GND    7       l


  F3-RTU
Port 1  PIN

  RXD-   6         l
  RXD+   1           l
  TXD-   8  <          l
  TXD+   9  <            l
  GND    7       l


  F3-RTU
Port 1  PIN

  RXD-   6         l                 62 Ù
  RXD+   1           l
  TXD-   8  <
  TXD+   9  <                        62 Ù      l
  GND    7       l
```

Cable Shielding

Shielding improves noise immunity (magnetic field protection). It is important to ground the shield at the receiver end only. Grounding the receiver end only provides the least high frequency signal attenuation and the best rejection of unwanted signals. Grounding both ends of the shield will cause magnetic field induced noised currents to flow through ground. Noise may then appear on the data lines due to transformer like coupling with the shield. If the cable shield is used as the system ground conductor then placing a 100 Ù resistor in series with the shield and the ground connection will reduce noise producing ground currents.

Connecting Cables and Line Termination

A dual twisted pair plus ground connection is recommended for 4-wire RS-422 networks. Proper termination of the balanced transmission line is required to prevent data errors. A typical AWG 22 solid wire with .060 inch plastic cover, twisted 4.5 times per foot has a characteristic impedance of about 120 Ù. Thus the selection of the two 62 Ù line-to-ground terminating resistors. Line-to-ground termination is preferred to the often shown line-to-line 120 Ù termination. In noisy or long line applications the much better line-to-ground common-mode rejection capability is particularly important. In multidrop networks, the line must be terminated at the extreme ends only as shown in the two previous diagrams. Addition of intermediate terminations will adversely load the line. If both the transmit and receive ends of the same twisted pair are terminated, double the value of the termination resistors.

Floating Data Lines Noise Prevention

The RS-422/485 drivers at the host should remain enabled to prevent noise from being received by the the slave devices on the network. To prevent noise reception at the host when there is no slave transmitting, add a pair of network biasing resistors to the host as shown in the two previous diagrams. This will pull-up the floating transmit line from the slaves to the RS-422/485 idle state (RXD+ to RXD- > .45 V). The equivalent of this can be done in a CoProcessor using the "P" parameter in the SETPORT statement.