Automationdirect.com™

Direct Logic 305

305 BASIC Modules

F3-AB128
F3-AB128-T

Order Number: F3-AB-M

TRADEMARKS

™***Automationdirect.com*** is a Trademark of ***Automationdirect.com***

™CoProcessor is a Trademark of FACTS Engineering, Inc.

COPYRIGHT

Last Issued Date: May 1998
Current Issued Date: September 1999

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 Manufacturers 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.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

This manual describes details specific to each of the 305 BASIC Module part numbers. This document should be used to supplement the FACTS Extended BASIC User's Reference when programming the FACTS Engineering 305 BASIC modules.

305 BASIC Modules are installed in a 16 point I/O slot of a DL305 PLC base. The 305 BASIC module communicates to the PLC CPU using the TRANSFER statement and PLC CPU ladder logic. The 305 BASIC module communicates to external devices using the built in serial port(s) or built in telephone modem depending on which module that you have. Check the feature overview to verify that you have the correct module for your application.

I/O REFERENCE DEFINITIONS

All 305 BASIC module use 8 PLC input and 8 PLC output reference points for addressing. The module may be placed in any 16 point slot. The table below defines the 16 references for slots 0-7.

I/O Reference Definition Table

Low 8 bits of Slot Reference (To PLC CPU from ABM)	Slot							
	0	1	2	3	4	5	6	7
Bit Weight 1 -->	0	10	20	30	40	50	60	70
Bit Weight 2 -->	1	11	21	31	41	51	61	71
Bit Weight 4 -->	2	12	22	32	42	52	62	72
Bit Weight 8 -->	3	13	23	33	43	53	63	73
Bit Weight 16 -->	4	14	24	34	44	54	64	74
Bit Weight 32 -->	5	15	25	35	45	55	65	75
Bit Weight 64 -->	6	16	26	36	46	56	66	76
Bit Weight 128 -->	7	17	27	37	47	57	67	77
Register Reference	R0	R1	R2	R3	R4	R5	R6	R7
High 8 bits of Slot Reference (From PLC CPU to ABM)								
Bit Weight 1 -->	100	110	120	130	140	150	160	170
Bit Weight 2 -->	101	111	121	131	141	151	161	171
Bit Weight 4 -->	102	112	122	132	142	152	162	172
Bit Weight 8 -->	103	113	123	133	143	153	163	173
Bit Weight 16 -->	104	114	124	134	144	154	164	174
Bit Weight 32 -->	105	115	125	135	145	155	165	175
Bit Weight 64 -->	106	116	126	136	146	156	166	176
Bit Weight 128 -->	107	117	127	137	147	157	167	177
Register Reference	R10	R11	R12	R13	R14	R15	R16	R17

Note: On some CPUs Slot 6 references (IO60-67,160-167 and R6,R16) are valid for 16 point I/O modules providing the instructions DSTR5 (F55) and DOUT5 (F65) are not used. Use instead two DSTR1 (F51) and two DOUT1 (F61) instructions or use only slots 0 - 5.

Not all CPUs support a 305 I/O BASIC Module in slot 7.

CHAPTER 2: TRANSFER INSTRUCTION

The TRANSFER statement is used by all 305 BASIC modules to communicate to the DL305 PLC CPU. This statement requires ladder logic in the PLC CPU. This chapter describes the TRANSFER statement and gives examples of how to use it.

TRANSFER

Function	TRANSFER outputs a byte to and inputs a byte from the PLC CPU
Syntax	var = TRANSFER (expr)
Shorthand	T.
Usage	TRANSFER outputs the value (0 to 255) of expr to the PLC CPU. At the same time TRANSFER inputs a value from the PLC CPU and assigns the value to var. If the optional expr is omitted then TRANSFER will input a value without changing the state of the outputs to the PLC CPU.

To guarantee synchronized data transfer with the PLC CPU, the TRANSFER takes place immediately after the PLC completes its' I/O update. Consecutive TRANSFERs will take place on consecutive scans (providing minimal BASIC code exists between each TRANSFER).

Communication Diagram For Consecutive TRANSFERs

PLC CPU	ABM
I/O Update ↓	Wait for end of PLC I/O Update
Solve Ladder Logic ↓	Read/Write PLC I/O with TRANSFER
I/O Update ↓	Wait for end of PLC I/O Update
Solve Ladder Logic	Read/Write PLC I/O with TRANSFER

Example: Input the PLC CPU data register 400 in three scans

```
100  REM TRANSFER code to get register 400 from PLC
110  REM Code will be read by PLC CPU during its' next I/O update
120  REM
125  DIM REG(254)
130  CODE = 128
140  DUMMY = TRANSFER (CODE)
150  REM
160  REM   Wait one scan while PLC CPU inputs code
170  REM
180  DUMMY = TRANSFER
190  REM
200  REM   PLC solves ladder logic for code and Outputs value
210  REM   of Register 400 to ABM before next TRANSFER.
220  REM
230  REG(40) = TRANSFER : REM   Register 400 read
```

Note that the dimensioned array REG(40) was used instead of REG400 since BASIC would interpret the variables REG400 and REG410 as the same variable. See CHAPTER 2, GETTING STARTED WITH FACTS EXTENDED BASIC in the FACTS Extended BASIC Reference Manual (FA-BASIC-M).

Consecutive TRANSFERs can be used to reduce communication time as shown in the following examples.

Input 8 PLC CPU data registers in 10 scans.

```
15 DIM REG(254)
20 DUMMY = TRANSFER (128) : REM Code to get register 400
25 DUMMY = TRANSFER (129) : REM Code to get register 401
30 REG (1) = TRANSFER(130) : REM PLC CPU returns reg. 400
35 REG (2) = TRANSFER(131) : REM Input 401, Output code for 403
40 REG (3) = TRANSFER(132) : REM Input 402, Output code for 404
45 REG (4) = TRANSFER(133) :REM Input 403, Send code for T601 MSB
50 REG (5) = TRANSFER(134) : REM Input 404,Send code for T601 LSB
55 REG (6) = TRANSFER(135) : REM Input T601 MSB,Send code for 405
60 REG (7) = TRANSFER : REM      ABM inputs Timer 601 LSB
65 REG (8) = TRANSFER : REM      PLC returns last reg. value, 405
70   END
100  REM        Register          REG(index)      Code
110  REM        400                     1         128
120  REM        401                     2         129
130  REM        402                     3         130
140  REM        403                     4         131
150  REM        404                     5         132
160  REM        T 601 MSB               6         133
170  REM        T 601 LSB               7         134
180  REM        405                     8         135
```

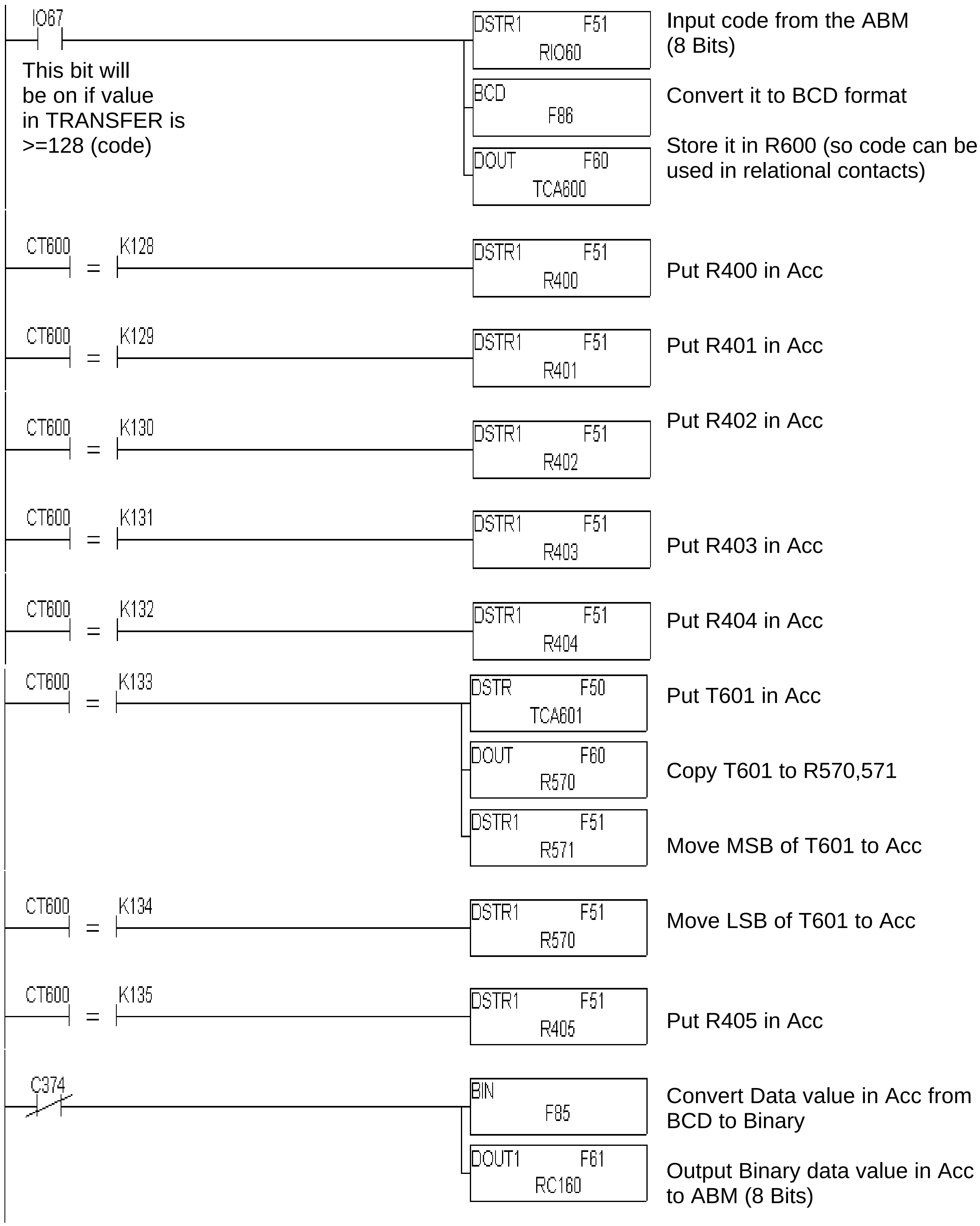
Notice that the code values used in the previous examples are greater than 127. This is done to make it easy for the ladder logic to distinguish codes from data values which would be in BCD and less than 100 in the PLC CPU.

Warning: Consecutive TRANSFERs yield the fastest possible PLC CPU data exchange. Incorrect operation can occur if a PLC I/O update takes place between TRANSFERs. This can only happen if additional BASIC statements are executed in between "consecutive" TRANSFERs. Therefore, ONTIME and ONPORT interrupts should be prevented from interrupting consecutive TRANSFERS.

One method would be to disable the interrupts during the TRANSFER sequence.

To preserve interrupt response times, set a flag (BASIC variable) during the interrupt subroutine. This flag would be cleared at the start of consecutive TRANSFERs and examined at the end of the TRANSFER sequence. If the flag is set at the end of the sequence then an interrupt has occurred and the TRANSFER sequence should be repeated.

The following example ladder logic (45 addresses) could be used with both of the previous examples. The references in the example are for a BASIC module placed in slot 6. To modify this example for a different slot replace references I/O67, R6, and R16 with the references for the slot where the BASIC module is installed.



Load a PLC CPU data register from BASIC

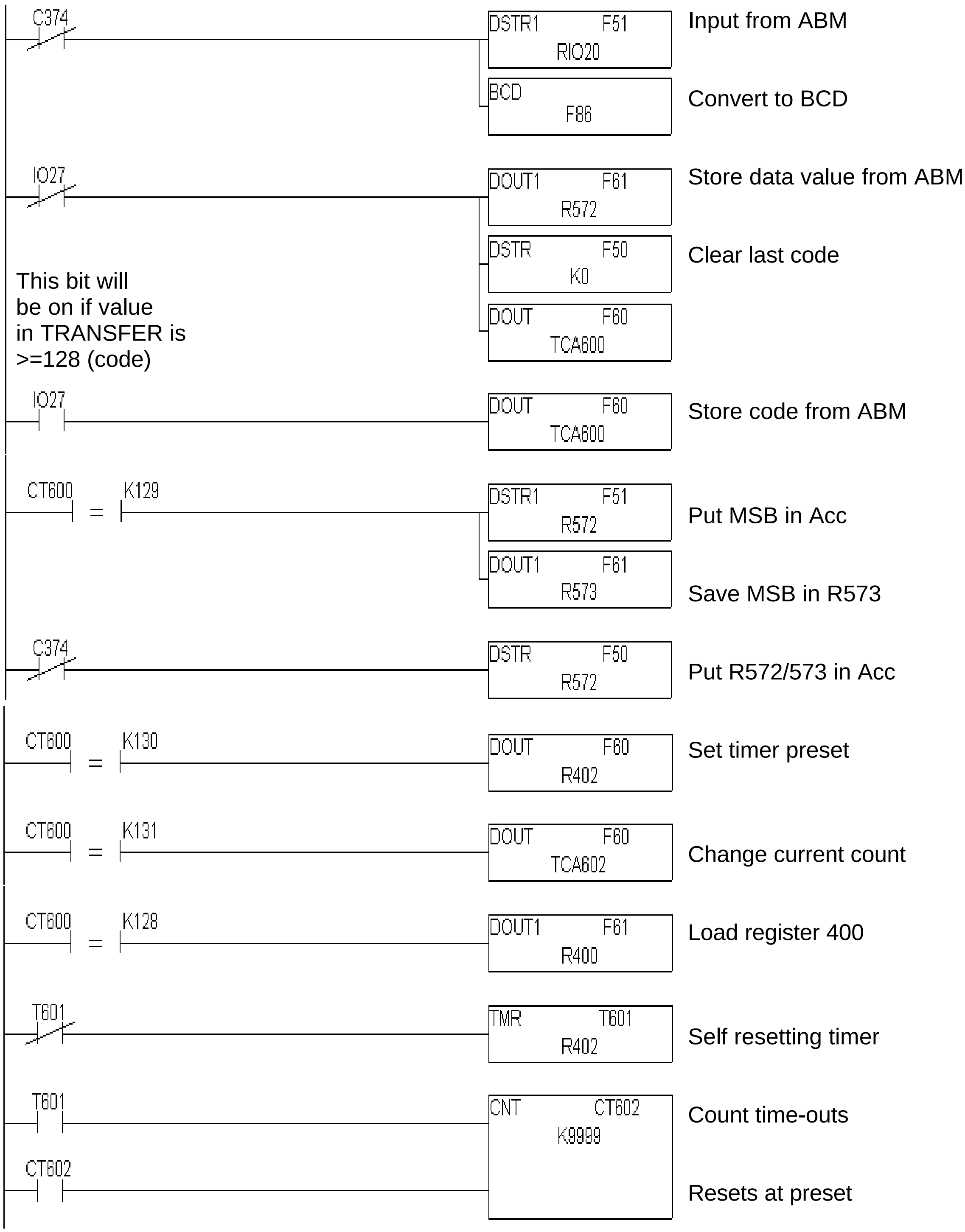
```
10   NUM = 99 : REM   Number to output 0 - 99
15   CODE = 128 : REM   Code to load register 400
20   DUMMY = TRANSFER (NUM)
30   DUMMY = TRANSFER (CODE)
40   END
```

Load two PLC CPU data register pairs. A register pair may be a timer/counter preset or accumulative value register or a 16 point I/O module such as a 12 bit analog output module.

```
10   LSB = 12 : REM   Least significant two BCD digits
20   MSB = 34 : REM   Most significant two BCD digits
30   LOAD_HIGH = 129 : REM   Code to store MSB
40   REM   Code to load timer 601 preset reg. pair 402/403
50   CODE = 130
60   DUMMY = TRANSFER (MSB) : REM   Output MSB data value
70   DUMMY = TRANSFER (LOAD_HIGH) : REM   Save it
80   DUMMY = TRANSFER (LSB) : REM   Output LSB data value
90   DUMMY = TRANSFER (CODE) : REM   Load the register pair
100  REM   Code to load current count for counter 602
110  CODE = 131
120  DUMMY = TRANSFER (40) : REM   MSB
130  DUMMY = TRANSFER (LOAD_HIGH)
140  DUMMY = TRANSFER (95) : REM   LSB
150  DUMMY = TRANSFER (CODE) : REM   Current count is now 4095
160  END
```

Helpful Hint: The variables LSB10 and LSB20, CODE15 and CODE25 or REG400 and REG410 will return the same value due to the way that BASIC stores variables. To avoid this problem, use dimensioned variables such as CODE(index) or REG(index).

The following example ladder logic (41 addresses) could be used with both of the previous examples. The references in the example are for a BASIC module placed in slot 2. The references used are for example only. Any references valid for a 16 point I/O module may be used.



PLC CPU OUTPUT TO BASIC MODULE

Install an eight point input module (such as an I/O Simulator) in slot one with the 305 I/O BASIC module in slot 0, and enter the following ladder logic.

The following 8 rungs of ladder logic are equivalent to the rung above.
Enter the following BASIC program to display a number which represents the status of the PLC inputs 10 through 17.


```
>10 P1. USING(###),TRANSFER,CR,
>20 GOTO 10
>RUN
```

The displayed number will be from 0 to 255 inclusive. Press <Ctrl-C> to stop program execution. The number displayed is equal to the sum of the binary weights for each PLC CPU output (100-107) which was ON.

BASIC MODULE OUTPUT TO PLC CPU

Install an eight point output module in the third I/O module slot, slot 2. and enter the following ladder logic (In this example, the ABM is still in slot 0, first I/O module next to CPU).

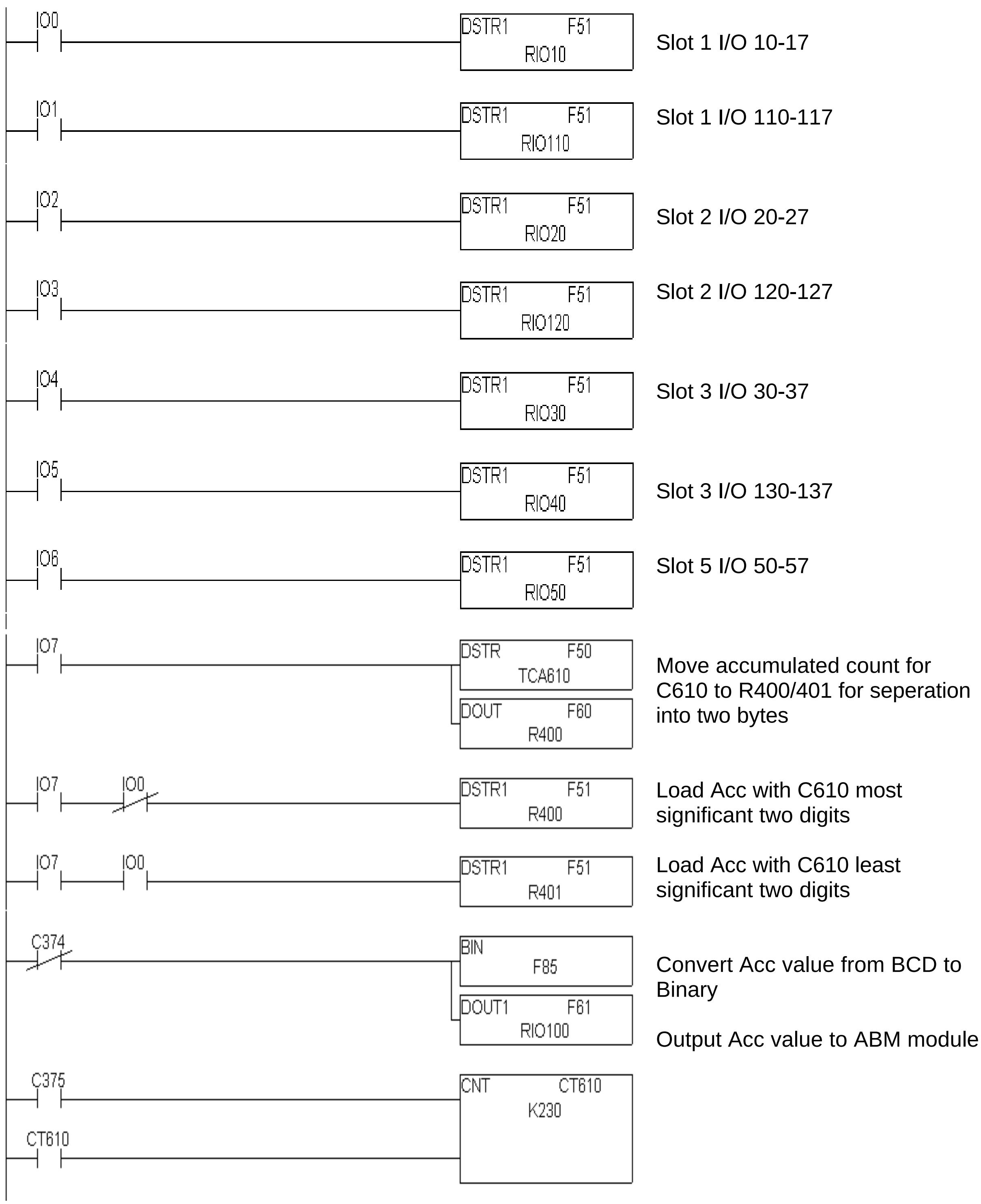
The following 8 rungs of ladder logic are equivalent to the rung above.

Enter the following BASIC program and use it with the previous ladder logic to control the status of the PLC outputs 20 through 27.

```
>10 INPUT1 "Energize or De-energize (E/D) ",$(0)
>20 IF UCASE$($(0))<>"E".AND.UCASE$($(0))<>"D" THEN END
>25 IF UCASE$($(0))="E" THEN OFF=0 ELSE OFF=NOT(0)
>30 INPUT "Enter output reference 20 - 27 ",$(0)
>32 IF $(0)="" THEN END
>35 OUT = VAL($(0)) - 20
>40 IF OUT<0.OR.OUT>7 THEN 30
>50 IF OFF THEN BIT(OUT)=0 ELSE BIT(OUT)=1
>60 DMY = TRANSFER (BITS)
>70 GOTO 10
```

LOADING SEVERAL PLC CPU REGISTERS INTO THE BASIC MODULE

The following ladder logic will output, at the request of the BASIC program, the status of 16 point I/O modules in slots one and two, the status of eight point I/O modules in slots three, four, and five, plus the accumulated valve of counter 610.



The following BASIC program can be used with the previous ladder logic program to control the transfer of data from the PLC CPU.

```
>10  DIM REG(8)
>20  A = TRANSFER(1)
>25  A = TRANSFER(2)
>30  REG(1) = TRANSFER(4)
>40  REG(2) = TRANSFER(8)
>50  REG(3) = TRANSFER(16)
>60  REG(4) = TRANSFER(32)
>70  REG(5) = TRANSFER(64)
>80  REG(6) = TRANSFER(128)
>85  REG(7) = TRANSFER(129)
>90 COUNT = TRANSFER + TRANSFER*100 : REM Counter LSB + MSB x 100
>95   PRINT1 "I/O STATUS FOR FIRST 56 POINTS"
>100  FOR INDEX = 1 TO 7
>110  BITS = REG(INDEX)
>120  FOR PT = 0 TO 7
>130  PRINT BIT(PT),SPC(2),
>140  NEXT PT : PRINT1 : NEXT INDEX
>160  PRINT "Counter 610 current value is ",COUNT
```

POWERFUL YET SIMPLE TO USE - BASIC/LADDER COMMUNICATION

The following example program (EXAM-305.ABM on the ABM Commander disk) can be used as the core of almost any application. It enables you to utilize all of the capabilities of the ABM even if you have little PLC data instruction experience.

The example demonstrates reading and writing 4 data registers (which in the example are timer/counter presets) and reading and changing the accumulated values in a timer and a counter. Also the status of an 8 point input module is read and the SET/RST of a coil is controlled by the ABM.

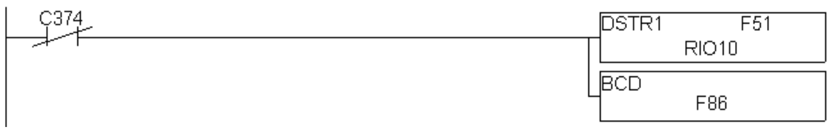
An important concept to understand is that the ABM "instructs" the PLC. This greatly simplifies the communication task. ABM outputs to the PLC will ALWAYS be read on the next PLC scan, therefore, the ABM ALWAYS knows how to interpret the data returned by the PLC CPU. As a result, no receipt of data confirmation is required by the PLC or the BASIC module. In an application where the PLC CPU is the master and must control the ABM, the BASIC module "instructs" the PLC to send the control information (one register could represent 8 PLC CPU instructions).

Use rungs 1 through 4, 13, 14, and 16 in your application. Then for each read or write to any register or register pair use one additional rung of logic similar to those in the example. Don't over look the power of the PLC data instructions. Registers references can be used for discrete I/O modules, Timer/Counter presets and accumulated values, data registers and more.

F A C T S   E X T E N D E D   B A S I C   I N T E R F A C E

Ladder logic for CPU communication

Power On
One Shot



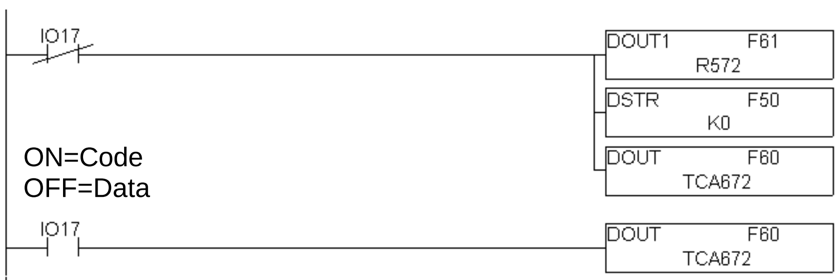
ABM Output to CPU

Input 17 is used to distinguish between BCD data (0-99) and command or instruction codes (128-255).

BASIC will always output data first. If 8 bit data is required it will be stored in register 572. If 16 bit data is required it will be stored in register pair 572/573.

This rung puts data into temporary storage register 572 and cancels the last ABM command or instruction code.

ON=Code
OFF=Data



Temp. Storage LSB

ON=Code
OFF=Data

ABM Command sequencer

Command used when transferring 16 bit data.

Moves data from LSB to MSB position in temporary data storage register pair 572/573.

Comand from ABM

Temp. storage LSB

Temp. storage LSB

ABM command to move data from temporary storage register pair 572/573 to data register pair 400/401. This is the preset for timer 600 in this example.

Command
from ABM

Temp. storage LSB/MSB

TMR 600 preset

Command to set preset of counter 601.

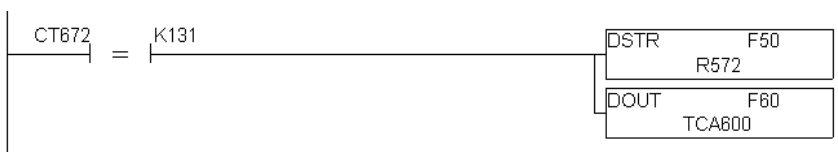
Command
from ABM

Temp. storage LSB/MSB

CNT 601 Preset

Command to change accumulate register for timer 600.

Command
from ABM



Temp. storage LSB/MSB

TMR 600 accumulated value

Command to change accumulate register (current count) for counter 601. Note that counter 601 could be a 99 or 9999 step additional ABM command sequencer if more than 128 command codes were required in an application.

Command
from ABM

Temp. storage LSB/MSB

CNT 601 accumulated value

MOVE DATA FROM 305 CPU TO ABM

Put data register pair 400/401 into temporary data storage register pair 572/573. This takes a snap shot of the current 16 bit value. Then load the two least significant BCD digits into the accumulator for output to the ABM. In this example 400/401 is the preset for timer 600.

Command
from ABM

Timer 600 preset

Temp. storage LSB/MSB

Temp storage LSB

Put the LSB of the preset for counter 601 into the accumulator.

Command
from ABM

CNT 601 preset

Temp. storage LSB/MSB

Temp storage LSB

Put the LSB of the accumulate register for timer 600 into the accumulator (2 least significant BCD digits of current time).

Command
from ABM

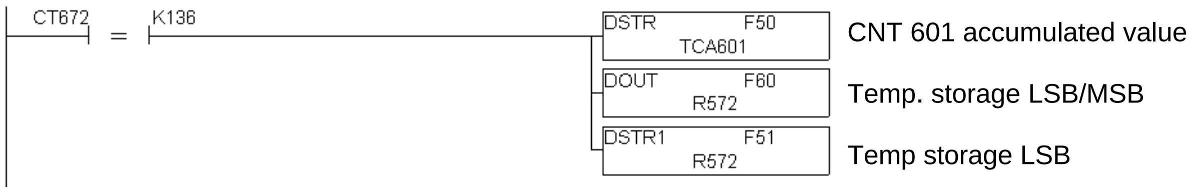
Timer 600 accumulated
value

Temp. storage LSB/MSB

Temp storage LSB

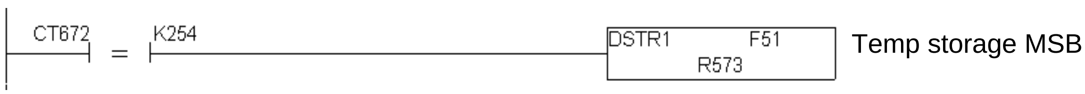
Put LSB of counter 601 accumulate register into the accumulator.

Command
from ABM



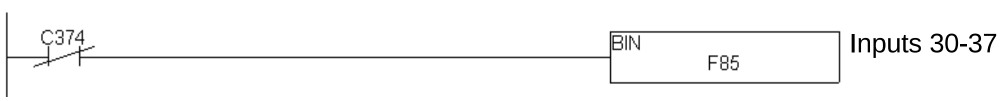
Put MSB of data into the accumulator for output to the ABM (data is the 2 most significant BCD digits associated with commands 133-136).

Command
from ABM



Convert BCD data in accumulator to binary for output to ABM.

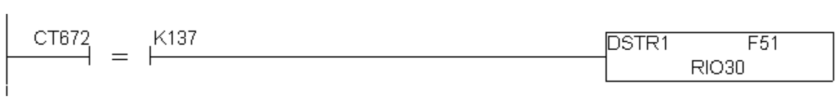
Power On
One Shot



Put status of inputs 30-37 in accumulator.

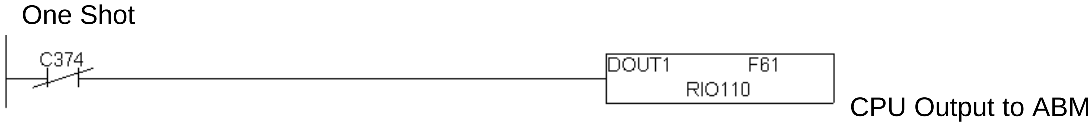
ABM Commands to input PLC I/O registers should be after the BCD to binary conversion rung (previous rung) since I/O register references are already in binary.

Command
from ABM



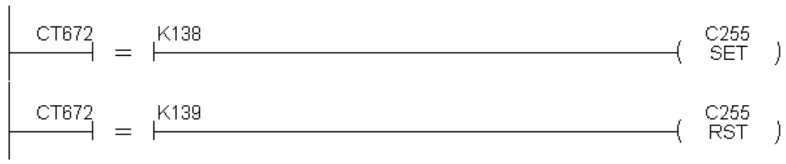
Always output the data in the accumulator to the ABM.

Power On
One Shot



C O N T R O L   3 0 5   C P U   O U T P U T S   F R O M   A B M

Command
from ABM



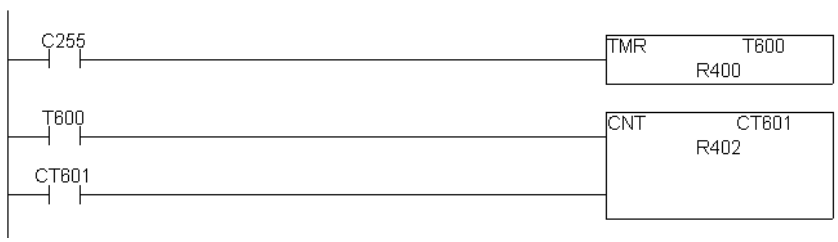
SET/RST by ABM

SET/RST by ABM

S T A R T   O F   U S E R ' S   L A D D E R   L O G I C

The ABM interface logic can be placed anywhere in the program. If the ABM must force (over-ride) inputs then those commands would be placed before the contacts are used in the program. If the ABM must force outputs then those commands must be placed after the output coil in the program. In most applications the ABM will not force I/O.

SET/RST
by ABM



BASIC Program for CPU communication

```
10    REM
20    REM FACTS Extended BASIC Program EXAM-305.ABM
30    REM
40    REM Demonstrates speed and simplicity of PLC CPU interface
50    REM
60    GOTO 1000
70    REM
80    REM Place often called subroutines at beginning for
90    REM maximum speed
200   REM
210   REM Routine to load a PLC CPU 16-bit register
220   REM
230   MSB=INT(N/100)
240   LSB=N-MSB*100
250   DUMMY=TRANSFER(MSB)
260   DUMMY=TRANSFER(128) :  REM Move MSB data to register 573
270   DUMMY=TRANSFER(LSB)
280   DUMMY=TRANSFER(REG_CODE) : REM Move preset data into reg.
290   RETURN
1000  REM
1010  REM         Input 8 PLC CPU 8-bit registers
1015  REM
1020  REM Registers could be I/O, internal relays, data
1030  REM registers, or T/C preset or accumulate registers.
1035  REM
1040  CLOCK 1 : TIME=0 : DBY(71)=0 :  REM Reset timer
1050  DUMMY=TRANSFER(133) :  REM Code to get register 400
1060  DUMMY=TRANSFER(254) :  REM Code to get register 401
1070  REG(1)=TRANSFER(134) :  REM Code to get register 402
1080  REG(2)=TRANSFER(254) :  REM Code to get register 403
1090  REG(3)=TRANSFER(135) :  REM Code to get LSB of reg 600
1100  REG(4)=TRANSFER(254) :  REM Code to get MSB of reg 600
1110  REG(5)=TRANSFER(136) :  REM Code to get LSB of reg 601
1120  REG(6)=TRANSFER(254) :  REM Code to get MSB of reg 601
1130  REG(7)=TRANSFER :  REM LSB of reg 601 stored in REG(7)
1140  REG(8)=TRANSFER :  REM MSB of reg 601 stored in REG(8)
1150  T=TIME
1160  PRINT1 "Eight PLC CPU 8-bit data registers input in ",
1170  PRINT1 T*1000," milli-seconds"
1175  T1=(REG(2)*100+REG(1))/10
1180  PRINT1 "Timer 600 preset is ",T1
1190  PRINT1 "Counter 601 preset is ",REG(4)*100+REG(3)
1195  A1=(REG(6)*100+REG(5))/10
1200  PRINT1 "Timer 600 current time is ",A1
1210  PRINT1 "Counter 601 current count is ",REG(8)*100+REG(7)
1220  IF A1>T1 THEN DUMMY=TRANSFER(139) :  REM   Reset timer 600
```

```
1300   REM
1310   REM Input and decode status of an 8 bit reg or I/O module
1320   REM
1325   TIME=0 : DBY(71)=0 :  REM  Set TIME and TIME fraction to 0
1330   DUMMY=TRANSFER(137) :  REM Get status of inputs 30-37
1340   DUMMY=TRANSFER :  REM PLC CPU gets the code this scan
1345   BITS=TRANSFER :  REM PLC CPU executes the code this scan
1350   T=TIME
1355   PRINT1 T*1000," milli-seconds to input and decode 8 PLC",
1360   PRINT1 " CPU Plus references"
1365   REM
1370   REM   Display Bit status for this register
1375   REM
1380   FOR I=0 TO 7
1385   PRINT1 "Input 3",I," = ",
1387   IF BIT(I) THEN  PRINT1 " ON" ELSE  PRINT1 "OFF"
1390   NEXT I
1395   REM
1400   REM   Set new preset values for TMR 600 and CNT 601
1405   REM
1410   SETINPUT 0,0,13,0,10000,5000
1420   INPUT1 "Please enter new preset for Timer 600 ",$(0)
1430   IF INPLEN=0 THEN  GOTO 1480
1440   N=VAL($(0))
1450   IF N>=.1.AND.N<=999.9 THEN N=N*10 ELSE  GOTO 1420
1460   REG_CODE=129
1470   GOSUB 200 :  REM Move new preset data into 400/401
1480   DUMMY=TRANSFER(138) :  REM   SET Coil 255 to start timer
1520   INPUT1 "Please enter new preset for Counter 601 ",$(0)
1530   IF INPLEN=0 THEN  END
1540   N=VAL($(0))
1550   IF N<0.OR.N>9999 THEN  GOTO 1520
1560   REG_CODE=130
1570   GOSUB 200 :  REM Move new preset data into 402/403

6000   REM
6001   REM              BASIC Module Command Code Summary
6002   REM
6010   REM     Code    Reg.     Description
6020   REM     128     573     Move data from register 572 to 573

6030   REM     129   400/401   Move 572/573 to 400/401
6040   REM     130   402/403   Move 572/573 to 402/403
6050   REM     131     600     Move 572/573 to TMR 600 acc reg
6060   REM     132     601     Move 572/573 to CTR 601 acc reg
6070   REM     133   400/401   Move 400/401 to 572/573
6080   REM     134   402/403   Move 402/403 to 572/573
6090   REM     135     600     Move TMR 600 acc reg. to 572/573
6100   REM     136     601     Move CTR 601 acc reg. to 572/573
6110   REM     137     013     Move status of 130-137 to acc
6120   REM     138     255     SET the PLC CPU output coil 255
6130   REM     139     255     RST the PLC CPU output coil 255
6140   REM     254     573     Move MSB data from 573 to acc

7000   REM   Copyright(c) FACTS Engineering, Inc. 6/28/1989
```

CHAPTER 3: F3-AB64

F3-AB64 GENERAL SPECIFICATIONS

Mounting Requirement	- 16 Point I/O Slot - Up to 8 modules per CPU base
Power Consumption	- 75 mA typical - 90 mA @ 9 Vdc maximum (supplied by base)
Operating Environment	- 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	- 64K Total (32K Data, 32K Program)
Physical Connectors	- Two 9 Pin D type connectors - Port 1 (top 9 Pin) - Port 2 and 3 (bottom 9 Pin)
Indicator LEDs	- RXD 1 and RXD 2
Port 1	- RS232/422/485 Jumper Selectable - 57600 Baud Maximum
Port 2	- RS232 - 9600 Baud Maximum
Port 3	- RS232 - 9600 Baud Maximum
Additional Features	- Battery Backed Calendar/Clock - Programmable from Port 1 or Port 2

See APPENDIX G: MEMORY MAPS for details of the 64K memory map.

NOTE: This module has been obsoleted and superseded by the F3-AB128. See Chapter 4 for details on the F3-AB128.

F3-AB64 DESCRIPTION

This DL305 family compatible ASCII/BASIC Module features 64K of non-volatile memory, three serial ports, real-time battery backed calendar clock, floating point math, and the FACTS Extended BASIC interpreter.

64K bytes of nonvolatile memory allows multiple program storage and execution, DL305 CPU register expansion and retentive data storage and retrieval.

Port 1 is a high performance 57.6K baud fully configurable RS-232 or RS-422/485 serial interface. Port 2 is a 9600 baud fully configurable RS-232 serial interface. Port 3 is a 9600 baud fully configurable RS-232 serial interface. All three ports have 255 character type-a-head input buffers for simultaneous communication with three or more external devices.

The real-time battery-backed calendar clock maintains time and date when power outages occur. Time based BASIC interrupts can be programmed to .005 of a second.

Floating point math solves complex formulas to 8 significant digits.

The FACTS Extended BASIC interpreter has many features and statements that simplify control oriented programming.

- ✓ Program from Port 1 or Port 2 (COMMAND@)
- ✓ Flexible bit manipulation instruction (BITS and PICK)
- ✓ Serial port and timer interrupts (ONPORT and ONTIME)
- ✓ Extensive serial port control (SETPORT, SETINPUT, PRINT, INPUT, INPLEN, INLEN)
- ✓ Extensive string manipulation instructions (MID$, LEFT$, RIGHT$, REVERSE$, ASC, CHR$, LCASE$, UCASE$, STR$, VAL, HEX$, OCTHEX$, DATE$, TIME$)
- ✓ Debugging tools (TRACE, STOP, CONT)
- ✓ Program chaining (GOPRM)
- ✓ Statements and control structures common to most BASICs

F3-AB64 IMPORTANT INFORMATION

Commands and Statements Not Supported

The following commands and statements that are described in the FACTS Extended BASIC User's reference are not supported by this module.

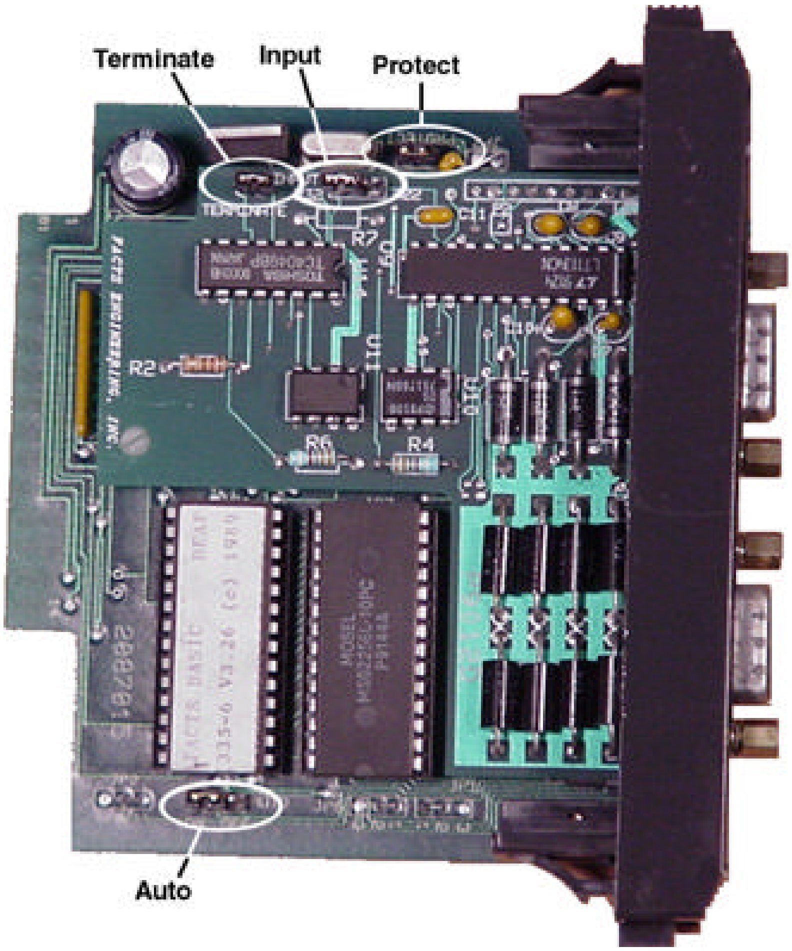
- OCTHEX$
- REVERSE$
- SYSTEM

Differences from FACTS Extended BASIC User's Reference (FA-BASIC-M)

RIGHTS$ - String Operator

Function	RIGHT$ returns a n character string beginning with the last character
Syntax	string variable = RIGHT$(string expression, n)
See Also	LEFT$, MID$
Usage	n is an expression and specifies the number of characters of string expression to be returned. n must be in the range 0 to 254. RIGHT$ returns a string consisting of the last through the nth character of its string argument. If n is greater than or equal to the length of string expression then all of string expression is returned. If n is 0 then RIGHT$ returns the null string. RIGHT$ allows you to reverse the order of all or part of a string in a single statement.
Example	>PRINT1 RIGHT$("SDRAWKCAB",20) BACKWARDS >PRINT1 RIGHT$("N20G45",2) 54

F3-AB64 JUMPER DESCRIPTIONS AND LOCATIONS



PROTECT

The PROTECT jumper is used to hardware write protect saved programs (program memory). Placing this jumper on both posts allows a program to be saved (this is the default factory setting). Placing this jumper on a single post DOES NOT allow program memory to be written to.

The error message UNABLE TO VERIFY will be generated if the SAVE command is executed while this jumper is installed on one post.

INPUT 422/232

The INPUT 422/232 jumper specifies the communication interface type for port 1. Placing this jumper in the 422 position selects the RS422/485 interface. Placing this jumper in the 232 position selects the RS232 interface (this is the default factory setting).

TERMINATE

The TERMINATE jumper is used to place a 120 ohm terminating resistor between RXD- and RXD+ on the port 1 RS422/485 interface. Placing the jumper on both posts includes the termination (this is the default factory setting). Placing this jumper on a single post removes the termination resistor.

If your application uses a multi-drop RS422/485 configuration then only the F3-AB64s at the extreme ends of the network should have this jumper installed. If your application requires different termination, then the TERMINATE jumper should be placed on a single post.

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS.

AUTO OFF/ON

The AUTO OFF/ON jumper specifies the AUTOSTART mode that the module will use at reset. Placing the jumper in the OFF position selects AUTOSTART mode 255. Placing the jumper in the ON position allows the module to use the last stored AUTOSTART parameters (this is the default factory setting).

See the AUTOSTART command in the FACTS Extended BASIC User's Reference.

The error message UNABLE TO VERIFY will be generated when the AUTOSTART command is entered with the AUTO ON/OFF jumper in the OFF position.

CAUTION: Placing this jumper in the OFF position will erase program 0, all stored variables, cancel a COMMAND@2, remove LOCKOUT, and clear stored AUTOSTART information.

F3-AB64 PORT PINOUTS

PORT 1 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	RXD1+	DATA INPUT HIGH, RS-422
2	TXD1	DATA OUTPUT, RS-232
3	RXD1	DATA INTPUT, 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 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	N/C	NOT USED, RESERVED
2	TXD2	DATA OUTPUT, RS-232
3	RXD2	DATA INTPUT, RS-232
4	RTS2/**TXD3 ***	DTE HAS DATA TO XMIT, RS-232
5	CTS2/**RXD3 ***	DTE MAY XMIT DATA, RS-232
6	N/C	NOT USED, RESERVED
7	GND	SIGNAL GROUND
8	N/C	NOT USED, RESERVED
9	N/C	NOT USED, RESERVED

***** The Port 3 TXD/RXD signals appear at unused hardware handshaking lines, RTS/CTS of Port 2. If Port 3 is not used then the lines may be used for hardware handshaking.

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

CHAPTER 4: F3-AB128

F3-AB128 GENERAL SPECIFICATIONS

Mounting Requirement	- 16 Point I/O Slot - Up to 8 modules per CPU base
Power Consumption	- 205 mA @ 9 Vdc maximum (supplied by base)
Operating Environment	- 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 and 3 (top 9 Pin) - Port 2 (bottom 9 Pin)
Indicator LEDs	- RXD 1 and RXD 2
Port 1	- RS232/422/485 Jumper Selectable - 57600 Baud Maximum
Port 2	- RS232 - 9600 Baud Maximum
Port 3	- RS232 - 9600 Baud Maximum
Additional Features	- Battery Backed Calendar/Clock - Programmable from Port 1 or Port 2

See APPENDIX G: MEMORY MAPS for details of the 128K memory map.

F3-AB128 DESCRIPTION

This DL305 family compatible ASCII/BASIC Module features 128K of non-volatile memory, three serial ports, real-time battery backed calendar clock, floating point math, and the FACTS Extended BASIC interpreter.

128K bytes of nonvolatile memory allows multiple program storage and execution, DL305 CPU register expansion and retentive data storage and retrieval.

Port 1 is a high performance 57.6K baud fully configurable RS-232 or RS-422/485 serial interface. Port 2 is a 9600 baud fully configurable RS-232 serial interface. Port 3 is a 9600 baud fully configurable RS-232 serial interface. All three ports have 255 character type-a-head input buffers for simultaneous communication with three or more external devices.

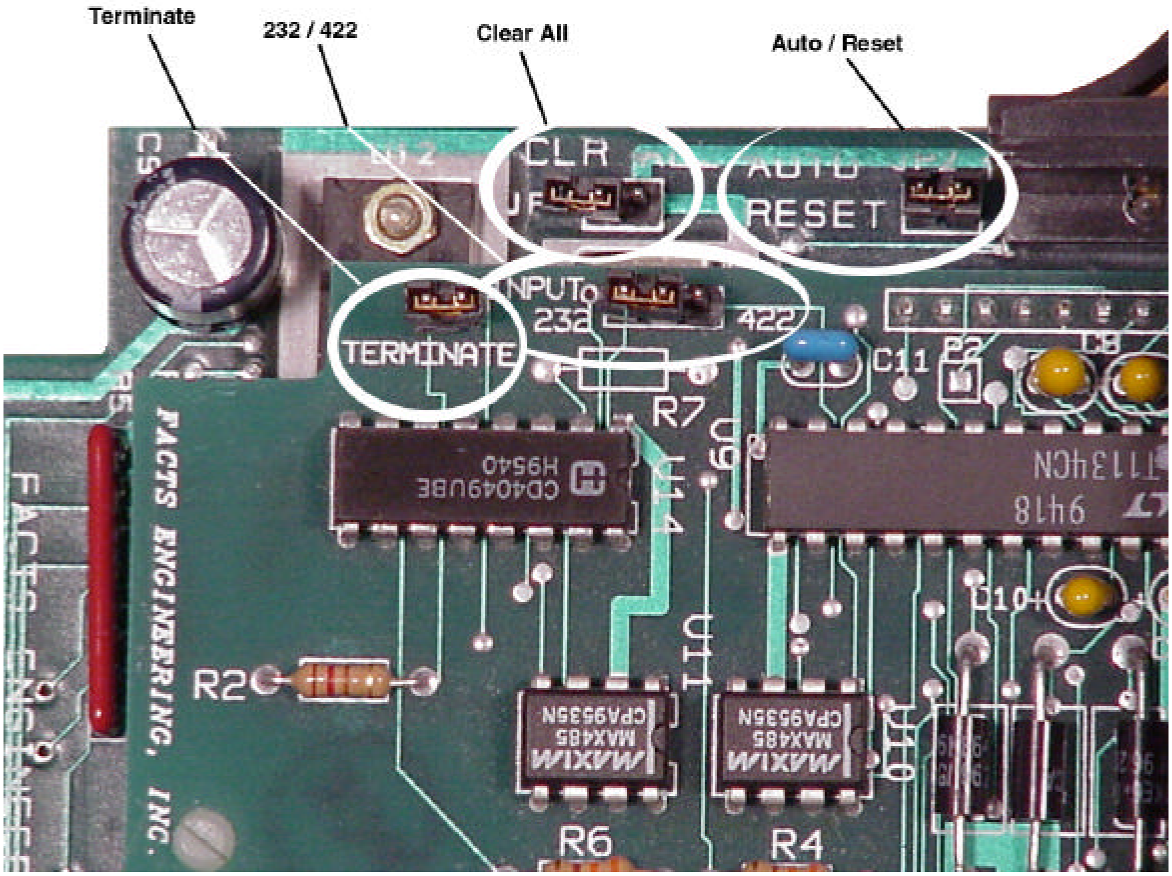
The real-time battery-backed calendar clock maintains time and date when power outages occur. Time based BASIC interrupts can be programmed to .005 of a second.

Floating point math solves complex formulas to 8 significant digits.

The FACTS Extended BASIC interpreter has many features and statements that simplify control oriented programming.

- ✓ Program from Port 1 or Port 2 (COMMAND@)
- ✓ Flexible bit manipulation instruction (BITS and PICK)
- ✓ Serial port and timer interrupts (ONPORT and ONTIME)
- ✓ Extensive serial port control (SETPORT, SETINPUT, PRINT, INPUT, INPLEN, INLEN)
- ✓ Extensive string manipulation instructions (MID$, LEFT$, RIGHT$, REVERSE$, ASC, CHR$, LCASE$, UCASE$, STR$, VAL, HEX$, OCTHEX$, DATE$, TIME$)
- ✓ Debugging tools (TRACE, STOP, CONT)
- ✓ Program chaining (GOPRM)
- ✓ Statements and control structures common to most BASICs

F3-AB128 JUMPER DESCRIPTIONS AND LOCATIONS



AUTO/RESET (JP2)

The AUTO/RESET jumper specifies the action the module will take when the PLC/CPU goes into program mode. Placing this jumper on both posts causes the module to remain RESET while the PLC CPU is in the program mode, (the module is not accessible until the PLC CPU is in run mode). this is the factory default. Placing the jumper on a single post causes the module to reset only at power up (the module is accessible as long as power is applied.

NOTE: If JP2 is placed on a single post and the PLC CPU is in program mode, executing a TRANSFER instruction will return a value of 32768 (8000H). The TRANSFER instruction will return a number between 0 and 255 if the PLCis in the run mode.

CLR ALL (JP1)

The CLR ALL jumper is used to select the AUTOSTART mode that the module uses at reset. Placing the jumper on both posts selects AUTOSTART mode 255. Placing the jumper on a single post allows the module to use the last stored AUTOSTART parameters (this is the default factory setting).

See the AUTOSTART command in the FACTS Extended BASIC User's Reference.

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@2, remove LOCKOUT, and clear stored AUTOSTART information.

INPUT 232/422

The INPUT 422/232 jumper specifies the communication interface type for port 1. Placing this jumper in the 422 position selects the RS422/485 interface to be used on port 1. Placing this jumper in the 232 position selects the RS232 interface to be used on port 1 (this is the default factory setting).

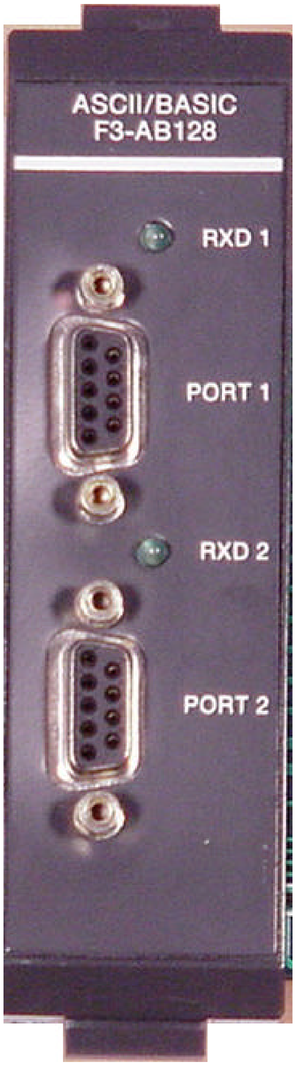
TERMINATE

The TERMINATE jumper is used to place a 120 ohm terminating resistor between RXD- and RXD+ on the port 1 RS422/485 interface. Placing the jumper on both posts includes the termination (this is the default factory setting). Placing this jumper on a single post removes the termination resistor.

If your application uses a multi-drop RS422/485 configuration then only the F3-AB128s at the extreme ends of the network should have this jumper installed. If your application requires different termination, then the TERMINATE jumper should be placed on a single post.

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

F3-AB128 PORT PINOUTS



PORT 1 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	RXD1+	DATA INPUT HIGH, RS-422
2	TXD1	DATA OUTPUT, RS-232
3	RXD1	DATA INTPUT, RS-232
4	RTS1/**TXD3 ***	DTE HAS DATA TO XMIT, RS-232
5	CTS1/**RXD3 ***	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

* The Port 3 TXD/RXD signals appear at unused hardware handshaking lines, RTS/CTS of Port 1. If Port 3 is not used then the lines may be used for hardware handshaking.

PORT 2 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	N/C	NOT USED, RESERVED
2	TXD2	DATA OUTPUT, RS-232
3	RXD2	DATA INTPUT, RS-232
4	RTS2	DTE HAS DATA TO XMIT, RS-232
5	CTS2	DTE MAY XMIT DATA, RS-232
6	N/C	NOT USED, RESERVED
7	GND	SIGNAL GROUND
8	N/C	NOT USED, RESERVED
9	N/C	NOT USED, RESERVED

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

CHAPTER 5: F3-AB128-T

F3-AB128-T GENERAL SPECIFICATIONS

Mounting Requirement	- 16 Point I/O Slot - Up to 8 modules per CPU base
Power Consumption	- 205 mA @ 9 Vdc maximum (supplied by base)
Operating Environment	- 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	- One 9 Pin D type connector (Port 1 and 3) - RJ-12 Line and Phone Jacks (Port 2)
Indicator LEDs	- RXD 1, RXD 2, and RXD 3
Port 1	- RS232/422/485 Jumper Selectable - 57600 Baud Maximum
Port 2	- Telephone Modem - 2400 Baud Maximum
Port 3	- RS232 - 9600 Baud Maximum
Additional Features	- Battery Backed Calendar/Clock - Programmable from Port 1 or Port 2

See APPENDIX C: TELEPHONE MODEM SPECIFICATIONS for details on the built in telephone modem.

See APPENDIX D: TELEPHONE MODEM AT COMMANDS for details on the AT command set.

See APPENDIX G: MEMORY MAPS for details on the 128K memory map.

F3-AB128-T DESCRIPTION

This DL305 family compatible ASCII/BASIC Module features three serial ports including a built-in full duplex, 300/1200/2400 baud PSK/FSK, asynchronous telephone modem (Port 2). Port 3 is a 9600 baud fully configurable RS-232 serial interface. Port 1 is a high performance 57.6K baud fully configurable RS-232 or RS-422/485 serial interface. The modem is Bell 212A, Bell 103, CCITT V.21, and CCITT V.22 and V.22 bis compatible.

The auto dialer includes the capability for DTMF tone generation and call progress monitoring (detects no dial tone, ring, and busy).

The modem can be programmed to auto answer calls based on the number of rings, the time and date or other application specific requirements.

Either port 1 or the modem port can be software configured to be the programming port. This feature enables remote programming (tele-servicing) and simplified software development.

Direct connection to the telephone line is made via a standard RJ-12 PHONE jack on the face of the module. When the modem is not in use, a standard telephone may be connected to the second RJ-12 LINE jack on the front of the module.

BASIC statements and industry standard "AT" commands are used to control the modem and communicate with other modems over the public switched telephone network (PSTN).

All three ports have 255 character type-a-head input buffers for simultaneous communication with three or more external devices.

Modem Operation

Command and on-line operating modes of the modem are enabled with the DTR2 statement.

```
10   DTR2 = 0 : REM Disable modem (Hang-up and exit on-line mode)
20   DTR2 = 1 : REM Enable modem (Enter command mode)
```

Modem "AT" Commands

"AT" command strings are PRINTed at Port 2 to control modem operation. These commands are subdivided into three major groups:

Configuration	Configuration commands affect the modem operation by changing the contents of "S-registers" (software registers). The B command which selects CCITT or BELL protocols is an example of this type of command. As a result of the B command, bit six of S-register 14 is set or cleared depending on which protocol is chosen.
Action	Action commands instruct the modem to perform an action Immediately. Examples are the "A", go off hook in Answer mode, command and the "D", dial, command.
Diagnostic	Diagnostic commands initiate Local Analog and Digital Loopback tests and Remote Digital Loopback test (Detailed information on diagnostic commands is available from FACTS Engineering).

The "AT" prefix begins every command string with the exception of "+++", escape, and the "A/", repeat, commands. "AT" (attention code) allows the modem to determine the PORT 2 data rate and parity setting (see SETPORT statement in the FACTS Extended BASIC User's Reference). The modem will adopt these parameters until a new "AT" command is received.

Multiple commands can be included in a single PRINT statement. Space characters are allowed between commands to improve readability. A command string must be terminated with the ASCII carriage return character (default value in S-register 3). If a line feed character follows the carriage return character it is ignored. A result code is returned by the modem following execution of the command (see RESULT CODES table).

The command buffer can hold up to 40 characters before connection is made. It holds 10 characters after connection is established.

Example

```
10 PRINT2 "AT DT 1(234)567-8910" : REM Touch tone  dialing
```

The punctuation symbols and spaces are optional and are ignored by the auto-dialer.

Modem "AT" Commands Summary

AT	Precedes all command lines except "+++", escape, and "A/", repeat
A	Answer call without waiting for ring
A/	Repeat previous command line
B	CCITT V.22 and V.21 mode
B1	Bell 212A and 103 mode (default)
C/C0	Transmit carrier off
C1	Transmit carrier on (default)
D	Dial a number (0-9 ABCD*#)
E	Turn Echo off
E1	Turn Echo on (default)
H	Go on hook or hang up, PHONE jack connected to LINE (default)
H1	Go off hook
Q	Return result codes (default)
Q1	Do not return result codes
Sn=x	Write x in S-register n
Sn?	Read S-register n
V/V0	Enable short form numeric result codes (default)
V1	Enable full word result codes
X/X0	CONNECT result code enabled (default)
X1	CONNECT 1200/2400 result codes is enabled, no call progress monitoring
X2	Same as X1 except dial tone detection is enabled
X3	Same as X1 except busy signal detection is enabled
X4	ALL CONNECT result codes enabled including NO DIALTONE and BUSY
+++	The default escape code

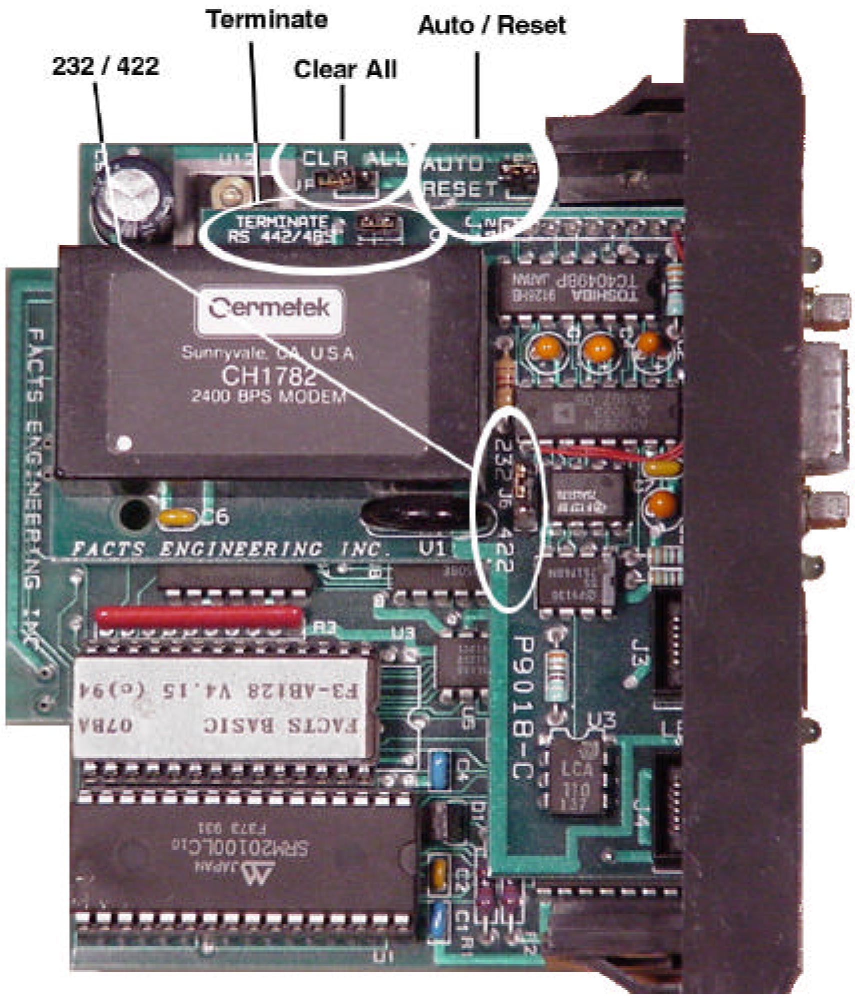
See APPENDIX D: TELEPHONE MODEM AT COMMANDS for a detailed description of the AT command set.

Result Codes

Result codes are strings returned by the modem in response to various "AT" commands and telephone line conditions.

Word Code	Result Code Description	Digit Code
OK	Command executed	0
CONNECT	Connected at 300 or 1200 bps	1
RING	Ring signal detected on telephone line	2
NO CARRIER	Carrier signal not detected or lost	3
ERROR	Illegal command or error in command line	4
CONNECT 1200	Connected at 1200 bps	5
NO DIALTONE	Dial tone was not detected	6
BUSY	Busy signal was detected	7
CONNECT 2400	Connected at 2400 bps	10

F3-AB128-T JUMPER DESCRIPTIONS AND LOCATIONS



AUTO/RESET (JP2)

The AUTO/RESET jumper specifies the action the module will take when the PLC/CPU goes into program mode. Placing this jumper on both posts causes the module to remain RESET while the PLC CPU is in the program mode, (the module is not accessible until the PLC CPU is in run mode). this is the factory default. Placing the jumper on a single post causes the module to reset only at power up (the module is accessible as long as power is applied.

NOTE: If JP2 is placed on a single post and the PLC CPU is in program mode, executing a TRANSFER instruction will return a value of 32768 (8000H). The TRANSFER instruction will return a number between 0 and 255 if the PLCis in the run mode.

CLR ALL (JP1)

The CLR ALL jumper is used to select the AUTOSTART mode that the module uses at reset. Placing the jumper on both posts selects AUTOSTART mode 255. Placing the jumper on a single post allows the module to use the last stored AUTOSTART parameters (this is the default factory setting).

See the AUTOSTART command in the FACTS Extended BASIC User's Reference.

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@2, remove LOCKOUT, and clear stored AUTOSTART information.

INPUT 232/422

The INPUT 422/232 jumper specifies the communication interface type for port 1. Placing this jumper in the 422 position selects the RS422/485 interface to be used on port 1. Placing this jumper in the 232 position selects the RS232 interface to be used on port 1 (this is the default factory setting).

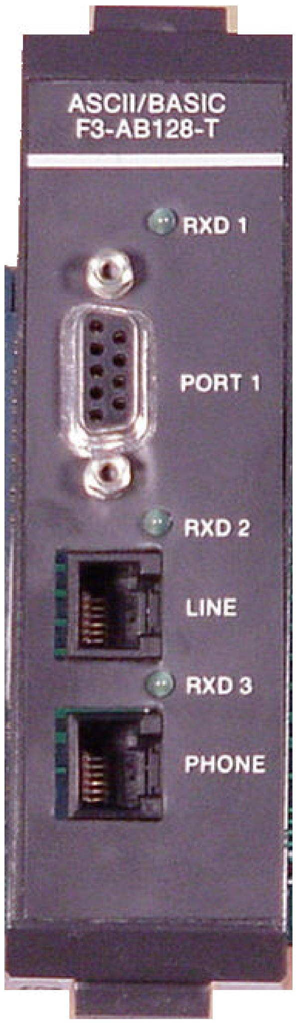
TERMINATE

The TERMINATE jumper is used to place a 120 ohm terminating resistor between RXD- and RXD+ on the port 1 RS422/485 interface. Placing the jumper on both posts includes the termination (this is the default factory setting). Placing this jumper on a single post removes the termination resistor.

If your application uses a multi-drop RS422/485 configuration then only the F3-AB128s at the extreme ends of the network should have this jumper installed. If your application requires different termination, then the TERMINATE jumper should be placed on a single post.

See APPENDIX F: RS232 AND 422/485 WIRING DIAGRAMS

F3-AB128-T PORT PINOUTS



PORT 1 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	RXD1+	DATA INPUT HIGH, RS-422
2	TXD1	DATA OUTPUT, RS-232
3	RXD1	DATA INTPUT, RS-232
4	RTS1/**TXD3 ***	DTE HAS DATA TO XMIT, RS-232
5	CTS1/**RXD3 ***	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

***** The Port 3 TXD/RXD signals appear at unused hardware handshaking lines, RTS/CTS of Port 1. If Port 3 is not used then the lines may be used for hardware handshaking.

CHAPTER 6: F3-AB128-R

F3-AB128-R GENERAL SPECIFICATIONS

Mounting Requirement	- 16 Point I/O Slot - Up to 8 modules per CPU base
Power Consumption	- 205 mA @ 9 Vdc maximum (supplied by base)
Operating Environment	- 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	- RXD 1 and RXD 2
Port 1	- RS232 - 57600 Baud Maximum
Port 2	- Leased Line / Radio Modem - 1200 Baud Maximum
Port 3	- RS232 - 9600 Baud Maximum
Additional Features	- Battery Backed Calendar/Clock - Programmable from Port 1 or Port 2

See APPENDIX E: LEASED LINE / RADIO MODEM SPECIFICATIONS.

See APPENDIX F: MEMORY MAPS for details on the 128K memory map.

NOTE: This module has been obsoleted and should not be used in new applications.

F3-AB128-R DESCRIPTION

This DL305 family compatible ASCII/BASIC Module incorporates a 300/600/1200 baud FSK, asynchronous modem. The modem is Bell 103/113/108, Bell 202, CCITT V.21, and V.23 compatible with fast response times for direct connection to leased-line networks and radios. BASIC statements are used to control the modem and communicate with external devices.

Modem operation is selected by jumper placement for either a 2-wire or 4-wire connection. Communication at 300 baud full-duplex or 600/1200 baud half-duplex is achieved using a two wire interface. Full-duplex operation at 300 to 1200 baud is supported over a four wire interface.

DC isolation and common-mode rejection between the modem and the telephone line or radio is achieved with transmit and receive transformers.

Transmit and receive circuits are balanced to match the telephone network load impedance, which in the U.S.A. is typically 600Ω.

Although the modem is not FCC or DOC (Canada) approved for direct connection to the Public Switched Telephone Network (PSTN), transmit level and receive sensitivity may be fixed by jumper placement to conform to FCC Part 68. Alternate jumper placement permits transmit level adjustment and receive attenuation for non-telephone standard interfaces.

Hazard protection is provided by Metal-oxide varistors (MOVs) and back to back zener diodes which effectively clamp transients at the supply rails. An external 0.5A Slow Blow Fuse is recommended for each transformer to protect against power line shorts to the phone lines. Use of a Slow Blow type fuse will prevent safe transients from blowing the fuse.

FULL-DUPLEX, 2 or 4-WIRE OPERATION

General Description:

In a 4-wire leased-line or radio connection, the transmit signals are isolated from the receive signals thus allowing the same frequencies to be used to communicate in both directions. This is necessary since a high speed single Bell 202 or CCITT V.23 channel uses more than half of the available bandwidth.

For the lower speed (300 baud) Bell 103 and CCITT V.21, bandwidth is shared between two channels. This permits 2-wire full-duplex operation (transmit and receive simultaneously).

Communication Protocol:

The calling modem asserts RTS which enables its' transmitter. After "Carrier Detect ON Delay" the called modem begins to receive data. After a longer time period, "RTS to CTS ON Delay", the calling modem begins to transmit data. In this manner, data may be transmitted in both directions simultaneously.

Lease-line Communication:

When hardware handshaking is enable for Port 2 with the SETPORT statement, communication timing is taken care of automatically as follows.

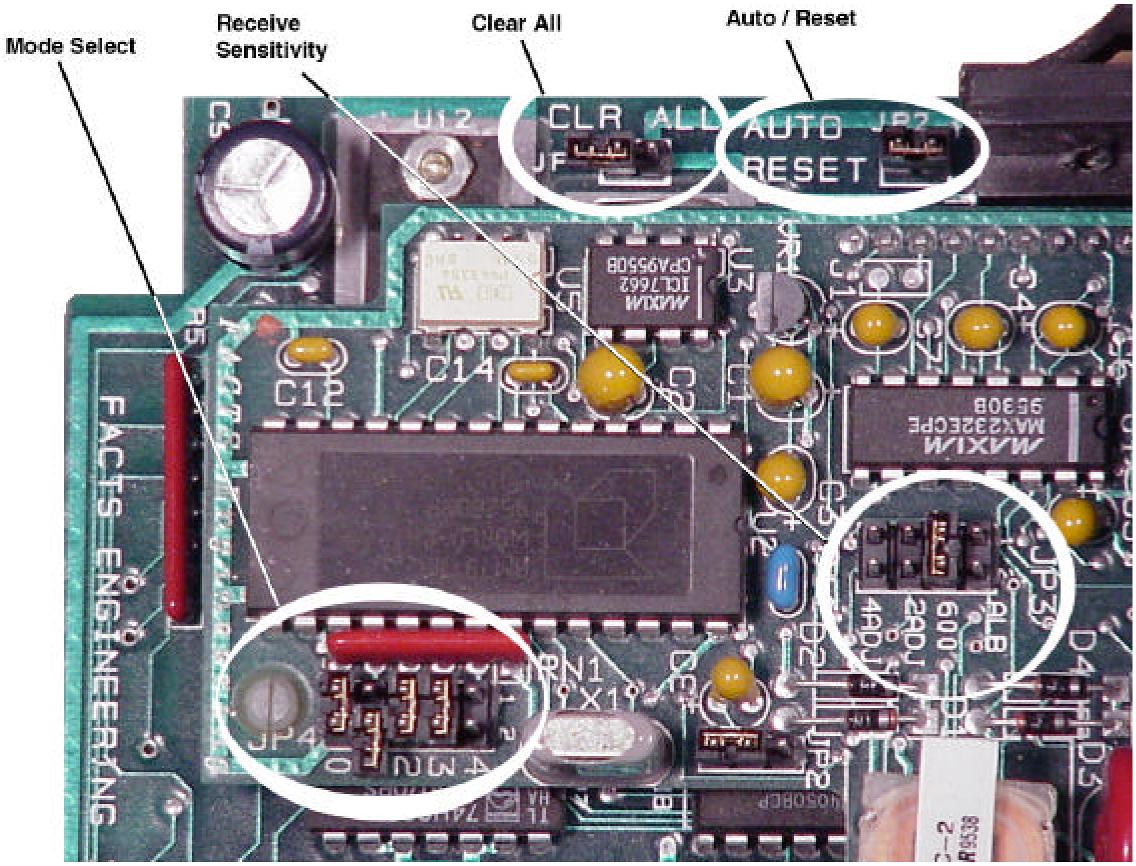
1) A PRINT2 statement will assert RTS. This will close the relay contact and begin carrier transmission.

2) After 8 msecs (25 msecs at 300 baud) data transmission of the PRINT statement list will begin.

3) Two character times after the last data bit has been transmitted, RTS will turn OFF thus dropping the carrier and opening the relay contact (RTS can be extended at the end of a PRINT2 statement by changing the contents of DBY(27), the default is DBY(27)=2).

4) Data reception begins automatically after a valid carrier signal has been detected for 3 to 5 msecs (10 to 16 msecs at 300 baud). Received data is placed into the 255 character Port 2 input buffer. Use the FACTS Extended BASIC statements ONPORT2, INPUT2, INLEN2, INKEY$2, and SETINPUT to process the received data.

F3-AB128-R JUMPER DESCRIPTIONS AND LOCATIONS



AUTO/RESET (JP2)

The AUTO/RESET jumper specifies the action the module will take when the PLC/CPU goes into program mode. Placing this jumper on both posts causes the module to remain RESET while the PLC CPU is in the program mode, (the module is not accessible until the PLC CPU is in run mode). this is the factory default. Placing the jumper on a single post causes the module to reset only at power up (the module is accessible as long as power is applied.

NOTE: If JP2 is placed on a single post and the PLC CPU is in program mode, executing a TRANSFER instruction will return a value of 32768 (8000H). The TRANSFER instruction will return a number between 0 and 255 if the PLCis in the run mode.

CLR ALL (JP1)

The CLR ALL jumper is used to select the AUTOSTART mode that the module uses at reset. Placing the jumper on both posts selects AUTOSTART mode 255. Placing the jumper on a single post allows the module to use the last stored AUTOSTART parameters (this is the default factory setting).

See the AUTOSTART command in the FACTS Extended BASIC User's Reference.

CAUTION: Placing this jumper on both posts will erase program 0, all stored variables, cancel a COMMAND@2, remove LOCKOUT, and clear stored AUTOSTART information.

MODE SELECTION (JP4)

Modem operating mode selection is made by positioning jumpers on a set of 5 pairs of pins labeled JP4. The pins are located below the crystal and are labeled along the board edge, "0 to 4". A "**Q**" in the following tables indicates that the jumper should be removed whereas a "**O**" indicates that it should be installed.

300 BAUD

Modem Operating Mode Description	4	3	2	1	0
Bell 103 orig., 300 bps, full-duplex, 2-wire	**O**	**O**	**O**	**O**	**O**
Bell 103 orig., 300 bps, full-duplex, 4-wire (Receive = Transmit Frequency, 1070/1270 Hz)	**Q**	**O**	**O**	**O**	**O**
Bell 103 answer, 300 bps, full-duplex, 2-wire	**O**	**O**	**O**	**O**	**Q**
Bell 103 answer, 300 bps, full-duplex, 4-wire (Receive = Transmit Frequency, 2025/2225 Hz)	**Q**	**O**	**O**	**O**	**Q**

600 BAUD

Modem Operating Mode Description	4	3	2	1	0
CCITT V.23 mode 1, 600 bps, half-duplex, 2-wire	**O**	**Q**	**O**	**O**	**O**
CCITT V.23 mode 1, 600 bps, full-duplex, 4-wire	**Q**	**Q**	**O**	**O**	**O**
CCITT V.23 mode 1, 600 bps, half-duplex, 2-wire with soft turn off	**O**	**Q**		**O**	**O**

1200 BAUD

Modem Operating Mode Description	4	3	2	1	0
Bell 202, 1200 bps, half-duplex, 2-wire	**O**	**O**	**O**	**Q**	**O**
Bell 202, 1200 bps, full-duplex, 4-wire	**Q**	**O**	**O**	**Q**	**O**
Bell 202 w/equ., 1200 bps, half-duplex, 2-wire	**O**	**O**	**O**	**Q**	**Q**
Bell 202 w/equ., 1200 bps, full-duplex, 4-wire	**Q**	**O**	**O**	**Q**	**Q**
CCITT V.23 mode 2, 1200 bps, half-duplex, 2-wire	**O**	**O**	**Q**	**Q**	**O**
CCITT V.23 mode 2, 1200 bps, full-duplex, 4-wire	**Q**	**O**	**Q**	**Q**	**O**
CCITT V.23 mode 2, 1200 bps, half-duplex, 2 wire with soft turn off	**O**	**Q**	**Q**	**Q**	**O**
CCITT V.23 mode 2 w/ equ., 1200 bps, half-duplex	**O**	**O**	**Q**	**Q**	**Q**
CCITT V.23 mode 2 w/ equ., 1200 bps, full-duplex	**Q**	**O**	**Q**	**Q**	**Q**
CCITT V.23 mode 2 w/ equ., 1200 bps, half-duplex, 2 wire with soft turn off	**O**	**Q**	**Q**	**Q**	**Q**

TRANSMIT LEVEL ADJUSTMENT POT

Position the jumper on the two pins nearest the transformers to enable transmit level adjustment. Access to the transmit level adjustment pot is made via the small hole above the Port 2 DE-9 connector. Turn the 25 turn pot clockwise to reduce the transmit level. To select the fixed -9 dBm transmit level, move the jumper on the three position header, JP2, towards the crystal.

RECEIVE SENSITIVITY ADJUSTMENT POT

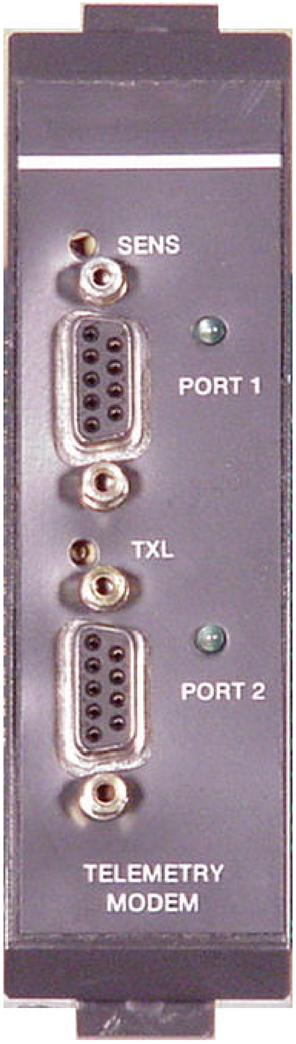
To enable 2-wire or 4-wire sensitivity adjustment, position a jumper on the JP3 pins labeled "2ADJ" or "4ADJ" respectively. Access to the receive sensitivity pot is made via the small hole above the Port 1 DE-9 connector. Turn the 25 turn pot counter-clockwise to attenuate (decrease) the receive sensitivity. To disable the pot (maximum sensitivity) position a jumper on the JP3 pins labeled "600".

COMMUNICATION PATH SELECTION (JP3)

Install a single jumper in one of the four sets of pins labeled JP3. This will selects the signal path as shown in the table below.

Communication Path	ALB	600	2ADJ	4ADJ
Data TXD to RXD (analog loop back)	**O**	**Q**	**Q**	**Q**
4-wire w/ no RXD sensitivity adjustment	**Q**	**O**	**Q**	**Q**
2-wire w/ RXD sensitivity adjustment	**Q**	**Q**	**O**	**Q**
4-wire w/ RXD sensitivity adjustment	**Q**	**Q**	**Q**	**O**

F3-AB128-R PORT PINOUTS



PORT 1 - PROGRAMMING/COMMUNICATION		
Pin	Signal Name	Description
1	N/C	NOT CONNECTED, RESERVED
2	TXD1	DATA OUTPUT, RS-232
3	RXD1	DATA INTPUT, RS-232
4	RTS1/**TXD3 ***	DTE HAS DATA TO XMIT, RS-232
5	CTS1/**RXD3 ***	DTE MAY XMIT DATA, RS-232
6	N/C	NOT CONNECTED, RESERVED
7	GND	SIGNAL GROUND
8	N/C	NOT CONNECTED, RESERVED
9	N/C	NOT CONNECTED, RESERVED

***** The Port 3 TXD/RXD signals appear at unused hardware handshaking lines, RTS/CTS of Port 1. If Port 3 is not used then the lines may be used for hardware handshaking.

PORT 2 - COMMUNICATION	
PIN	DESCRIPTION
1	MODEM TRANSMIT -
2	MODEM TRANSMIT +
3	RESERVED
4	MODEM RECEIVE -
5	MODEM RECEIVE +
6	NOT CONNECTED
7	RELAY OUTPUT N.O.
8	RELAY OUTPUT COM
9	NOT CONNECTED

TRANSMIT RELAY

An on-board Solid State relay provides a normally open contact closure which can be used to key a radio transmitter. Maximum load current is 100 mA from 0 to 300 V ac or dc. Maximum ON state resistance is 24Ω . The relay is controlled from BASIC using the DTR2 statement (DTR2 is same as modem RTS).

APPENDIX A: QUICK START

INITIAL MODULE OPERATION USING ABM COMMANDER PLUS

1. Run ABM Commander for Windows.

2. Review the ABM Commander for Windows Help/Instructions.

3. Connect the cable from the computer to the 305 BASIC module.. See APPENDIX F for wiring diagrams.

4. Turn ON the power to the PLC.

5. Select 'COMMAND MODE Connect to BASIC Module' from the main window. Select 'SYstem_Stats' from the COMMAND MODE menu. The 'SYstem_Stats' button will send a SPACE BAR character so the BASIC CoProcessor can correctly calculate the baud rate.

6. 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
   >   (">" character indicates BASIC is in COMMAND mode)

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

   NOTE: If using a F3-AB64 module you will not see the program number (PRM #) that is shown in these examples before the 'READY' message and command prompt.

   If you do not receive the sign on message, please follow the trouble shooting procedure in APPENDIX B.

7. The BASIC CoProcessor is now ready for programming and program upload/download.

EDITING A PROGRAM

User Action	Display Window
Select 'Auto' from the menu bar. Select Mode 0, Program 0, and Click 'OK'.	AUTOSTART 0,0 Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) >
Enter the following on the 'Command Line' field 10 p. <ENTER> 65535 p. <ENTER>	>10 p. >65535 p. >
Select 'ReseT' from the menu bar. Cycling the power to the PLC will also reset the BASIC CoProcessor.	RESET F A C T S E x t e n d e d B A S I C P l u s Series 305 Version 4.15 (c)Copyright FACTS Engineering, Inc. 1988 - 1994 AUTOSTART Mode, Program, Baud Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) 0 stored programs, 63275 program storage bytes free PRM 0 READY >
Select 'List' from the menu bar. Note that mode zero uses the stored baud rate. The program in the edit buffer, PROGRAM 0, is retained during loss of power in mode zero.	list 10 PRINT1 65535 PRINT1 PRM 0 READY >

SAVING A PROGRAM

User Action	Display Window
Select 'NeW' from the menu bar.	NEW >
Enter the following on the 'Command Line' field: 10 P."MY FIRST PROGRAM" <ENTER>	>10 p. "MY FIRST PROGRAM" >
Select 'SaVe' NOTE: The BASIC CoProcessor is shipped with a diagnostic program in PRM1 so the first SAVEd program will go into PRM2.	SAVE Saving program 2 2 stored programs, 63250 program storage bytes free PRM 0 READY >
Enter the following on the 'Command Line' field: 10 P."MY SECOND PROGRAM" <ENTER>	>10 p. "MY SECOND PROGRAM" >
Select 'SaVe'	SAVE Saving program 3 3 stored programs, 63224 program storage bytes free PRM 0 READY >

AUTO RUN MODE

User Action	Display Window
Select 'Auto' from the menu bar. Select Mode 1, Program 2, and Click 'OK'. This specifies that the BASIC CoProcessor will run program 2 after a reset.	AUTOSTART 1,2 Mode = 1, RUN (CLEAR) Program = 2 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) >
Select 'ReseT' from the menu bar. Cycling the power to the PLC will also reset the BASIC CoProcessor.	RESET MY FIRST PROGRAM PRM 2 READY >
Select 'Sel' from the menu bar. Click the 'Program 0' radio button then 'OK'.	>
Select 'List' from the menu bar. Confirm that the program in the edit buffer (PRM0) is still present.	list 10 PRINT1 "MY SECOND PROGRAM" PRM 0 READY >

DELETING A PROGRAM

User Action	Display Window
Select 'Del' from the menu bar. Enter '2' then click 'OK'. Click 'Yes' on the confirmation dialog.	DELPRM2 2 stored programs, 63249 program storage bytes free >
Select 'ReseT' from the menu bar. Cycling the power to the PLC will also reset the BASIC CoProcessor.	RESET MY SECOND PROGRAM PRM 2 READY >

CANCEL AUTO RUN MODE

User Action	Display Window
Select 'Auto' from the menu bar. Select Mode 0, Program 0, and Click 'OK'. This specifies that the BASIC CoProcessor will start up in edit mode after a reset.	AUTOSTART 0,0 Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) >

CHANGING THE PROGRAMMING PORT

When interfacing to a RS-422 or RS-485 device or when communicating with two or three external devices, you can change the RS-232 programming port from Port 1 to Port 2. This is done as shown below.

User Action	Display Window
In the 'Port Select' field (Bottom Left of the Command Window) select the 'Port 2' radio button.	No Change
In the 'Port Select' field click on the 'Command Port (ABM)' button. Click 'Yes' on the confirmation dialog.	No Change
Move cable from Port 1 to Port 2 then click 'OK' on the dialog prompting the cable change.	No Change
Select 'SYstem_Stats' from the menu bar.	>

APPENDIX B: TROUBLE SHOOTING

UNABLE TO ESTABLISH COMMUNICATION WITH BASIC COPROCESSOR

1. If the Port 1 RXD LED flashes when data is entered on the terminal then go 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.

2. If this is an F3-AB128, F3-AB128-R, F3-AB128-T then Power off the base, remove the module, and place the "CLR ALL" jumper on both posts.

   If this is a F3-AB64 then Power off the base, remove the module, and place the "AUTO OFF/ON" jumper in the OFF position.

   CAUTION: Installing this jumper will erase program 0, all stored data, cancel a COMMAND@2, remove LOCKOUT, and clear stored AUTOSTART information.

3. Run ABM Commander for Windows.

4. Review the ABM Commander for Windows Help/Instructions.

5. Connect the cable from the computer to the 205 CoProcessor module. See APPENDIX C for wiring diagrams.

6. Turn ON the power to the PLC.

7 Select 'COMMAND MODE Connect to BASIC Module' from the main window. Select 'SYstem_Stats' from the COMMAND MODE menu. The 'SYstem_Stats' button will send a SPACE BAR character so the BASIC CoProcessor can correctly calculate the baud rate.

8 If using an F3-AB128, F3-AB128-T, OR F3-AB128-R, the module will now respond with the sign on message.

```
FACTS Extended BASIC Plus
...
Stored AUTOSTART parameters sum check failed
Please re-enter AUTOSTART command
...

READY
>   (">" prompt character indicates BASIC is in COMMAND mode)
```

If using a F3-AB64, 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
Version 3.25 (C)Copyright FACTS Engineering, Inc. 1988-1993

AUTOSTART Mode, Program, Baud
Mode=255, NEW (space-bar)
Program =255

Port 1 Baud = 65535 Programming
(Port 2 = 65535)
(Port 3 = 65535)

0 stored programs, 8174 bytes free

READY
>   (">" prompt character indicates BASIC is in COMMAND mode)

9. Type the following command and press return.

   >AUTOSTART 0,0

10. Power off the base and remove the module. Place the "CLR ALL" jumper on a single post (or "AUTO ON/OFF" in the ON position.

11. Install the module and power up the base. 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)

APPENDIX C: TELEPHONE MODEM SPECIFICATIONS

TRANSMITTER SPECIFICATIONS

Description	Min	Typ	Max	Units
Carrier Output Level		-10	-9	dBm
Frequency Deviation	.01		.01	%

RECEIVER SPECIFICATIONS

Description	Min	Typ	Max	Units
Received Signal Range, OFF to ON	-45		0	dBm
Received Signal Range, ON to OFF	-48		0	dBm
Frequency Deviation Tolerance	-7		12	Hz

FREQUENCY PARAMETERS

300 Baud Modes

Modem Mode	Baud Rate	Transmit Frequency		Receive Frequency	
		Space Hz	Mark Hz	Space Hz	Mark Hz
Bell 103 Origin	300	1070	1270	2025	2225
Bell 103 Answer	300	2025	2225	1070	1270
CCITT V.21 Origin	300	1180	980	1850	1650
CCITT V.21 Answer	300	1850	1650	1180	980

1200/2400 Baud Modes

Description	Bell 212A	CCITT V.22
Baud Rate (bps)	1200/2400	1200/2400
Answer Band (Hz)	2400	2400
Originate Band (Hz)	1200	1200
Answer Tone (Hz)	2225	2100
Guard Tone (Hz)	--	1800

APPENDIX D: TELEPHONE MODEM AT COMMAND SET

DESCRIPTION OF MODEM AT COMMANDS

A Answer command

"A" command causes the modem to immediately go off-hook, answer the call and go into data transmission mode. Alternately the modem will automatically answer a call after the number of rings specified in S-register 0. By default, the modem will answer a call after 1 ring.

A/ Repeat command

"A/" command must be used without "AT" prefix and carriage return character at the end of the line. After receiving the "A/" command the modem repeats the execution of the last command stored in the command buffer. This command is normally not used.

Bn Bell/CCITT Protocol Compatibility

B/B0 Selects CCITT V.22 and V.21 modes of operation.
B1 Selects Bell 212A and 103 modes of operation (default).

Cn Enable carrier signal

C/C0 Turns the transmit carrier off.
C1 Enables the carrier signal (default).

D Dial telephone number

The "D" command causes the modem to go to the auto-dial mode. The parameters of the command include ASCII characters A to D, digits 0 to 9, *, and #. The dial modifiers are:

T	Touch-tone dial
P	Pulse dial
R	Reverse mode (change from originate mode to answer mode)
W	Wait for a dial tone
,	Pause before executing the next command in the dial string. waiting time is specified in S-register 8. Default is 2 seconds.
;	Go to command state after dialing. The line will be held for further commands. No connection will be made to another modem.
!	Go on-hook for 1/2 second (flash).
@	Wait for a ring back, followed by 5 seconds of silence before processing the next symbol in the dialing string. The waiting time is specified in S-register 7. Default is 30 seconds.

En Echo command

E1 Enables the modem to echo back to the BASIC module input buffer the characters transmitted out Port 2 (default).
E0 Disables local echo.

Hn Switch hook control

- H0 Causes the modem to go on-hook or hang up (default).
- H1 Causes the modem to go off-hook

Qn Result codes

- Q0 Enables result codes (default).
- Q1 Disables result codes.

Sn Direct register commands

Instead of stepping through a sequence of "AT" commands, the S-registers my be written directly. This reduces the time needed to configure the modem.

- Sn? Returns decimal value stored in S-register n.
- Sn=x Stores decimal equivalent of a binary number in S-register n (the range of x from 0 to 255)

Vn Select type of result code, verbal or numeric

- V0 Selects the short numeric form of the result code (one or two decimal digits) .
- V1 Selects a long verbal form of the result codes. Long form codes are preceded and terminated with both carriage return and line feed characters.

Xn Enable extended result code

X/X0 Requests the modem to send a CONNECT message when connection is established with the remote modem. In this mode, the modem dials regardless of the presence or absence of the dial tone. Dialing begins after waiting for the number of seconds specified in S-register 6 (default is 2 seconds). The modem does not recognize the dial tone or busy signal in this mode. This is the default.

X1 Requests the modem to send a full CONNECT 1200 message once connection is established at 1200 bps or a CONNECT message when both modems are operating at 300 bps. In this mode, the modem dials regardless of the presence or absence of the dial tone. Dialing begins after waiting for the number of seconds specified in S-register 6 (default is 2 seconds). The modem does not detect the dial tone or busy signal.

X2 Requests the modem to send a full CONNECT message (Connect 300 , or CONNECT 1200) once connection is established. The modem waits for the dial tone before it dials. If the dial tone is not detected within five seconds, the NO DIALTONE result code is returned. The busy signal is not detected.

X3 Requests the modem to send a full CONNECT message (CONNECT 300, or CONNECT 1200) once connection is established. In this mode, the modem dials regardless of the presence or absence of the dial tone. Dialing begins after waiting for the number of seconds specified in S-register 6 (default is 2 seconds). If busy signal is detected, the BUSY result code is returned.

X4 Requests the modem to send a full CONNECT message (CONNECT 300, or CONNECT 1200) once connection is established. The modem waits for the dial tone before it dials. If the dial tone is not detected within five seconds, the NO DIALTONE result code is returned. If busy signal is detected, the BUSY result code is returned.

&Dn Data Terminal Ready Option

D0 Modem ignores BASIC DTR2 statements (default).
D1 Modem assumes command state when DTR2 goes from 0 to 1.
D2 Modem hangs up, assumes command state and disables auto answer with the BASIC line DTR2=0 : DELAY 3000 : DTR2=1.
D3 Modem assumes software reset state when DTR2 goes from 0 to 1.

+++ The escape code

The escape code, preceded and followed by guard time intervals, returns the modem to the command mode from the on-line state without releasing the telephone line.

APPENDIX E: RADIO MODEM/LEASED LINE MODEM SPECIFICATIONS

TRANSMITTER SPECIFICATIONS

Description	Min	Typ	Max	Units
Fixed Carrier Output Level		-10	-9	dBm
Adjustable Carrier Output Level	-40		0	dBm
Frequency Accuracy (Except 202)	-0.4		0.4	Hz
Frequency Accuracy (202 Mark)	-1.0		1.0	Hz
Request to Send (RTS) to Clear to Send (CTS) ON Delay (std.)		25 8		ms @ 300 bps ms @ 1200 bps
Request to Send (RTS) to Clear to Send (CTS) OFF Delay (std.)		.5		msec

RECEIVER SPECIFICATIONS

Description	Min	Typ	Max	Units
Received Signal Range	-45		0	dBm
Carrier Detect ON Level		-42		dBm
Carrier Detect Hysteresis		5.5		dB
Frequency Deviation Tolerance	-16		16	Hz
Carrier Detect ON Delay	10 3		16 5	ms @ 300 bps ms @ 1200 bps
Carrier Detect OFF Delay	7 2		20 8.5	ms @ 300 bps ms @ 1200 bps

FREQUENCY PARAMETERS

Modem Mode	Baud Rate	Transmit Frequency		Receive Frequency	
		Space Hz	Mark Hz	Space Hz	Mark Hz
Bell 103 Origin	300	1070	1270	2025	2225
Bell 103 Answer	300	2025	2225	1070	1270
CCITT V.23 Mode 1	600	1700	1300	1700	1300
CCITT V.23 Mode 2	1200	2100	1300	2100	1300
CCITT V.23 Mode 2 Equalized	1200	2100	1300	2100	1300
Bell 202	1200	2200	1200	2200	1200
Bell 202 Equalized	1200	2200	1200	2200	1200

Equalization

The Equalized modes use fixed digital filters to compensate for both amplitude and group delay distortion (group delay is the derivative of the phase of a signal with respect to frequency). CCITT V.23 equalization is for a typical European phone line. Bell 202 Equalization is for a typical U.S. phone line.

Soft Turn-Off

When RTS is turned OFF at the end of a message, transients occur which may cause spurious space signals to be received at a remote modem (thus generating unwanted characters). During soft turn-off the modem transmits a soft carrier frequency of 900 Hz for 8 msec. This results in a steady Mark on the receive line of the remote modem. Soft turn-off is selectable for 1200 baud operation, full or half duplex.

Soft turn-off modes are only appropriate for leased-line communication (turning off RTS turns off the radio transmitter). Radio communication soft turn-off is accomplished be holding RTS ON for 1 or 2 character times after PRINTing the data (see BASIC example program).

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.

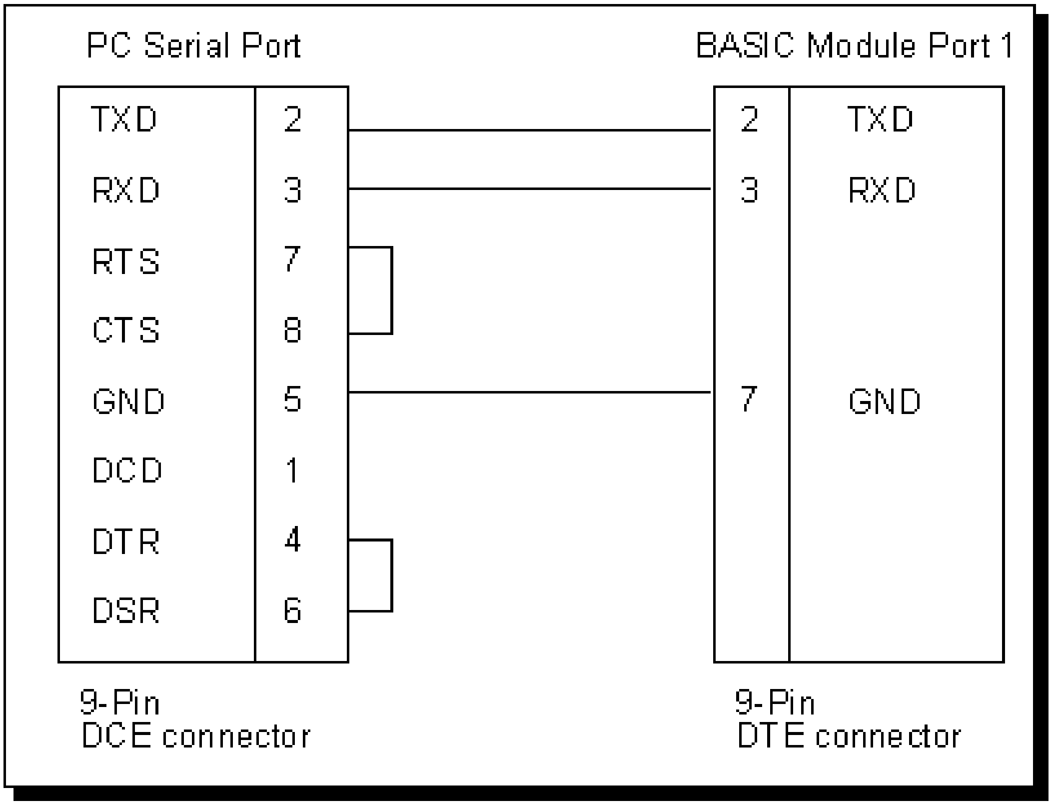
RS-232 signals travel over a serial interface cable that may have up to 25 wires. Since most signals are not required for simple communication, cables have as few as 2 or 3 wires. As shown in the following cabling diagrams, jumpers often are installed at one or both of the connectors to ensure that flow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

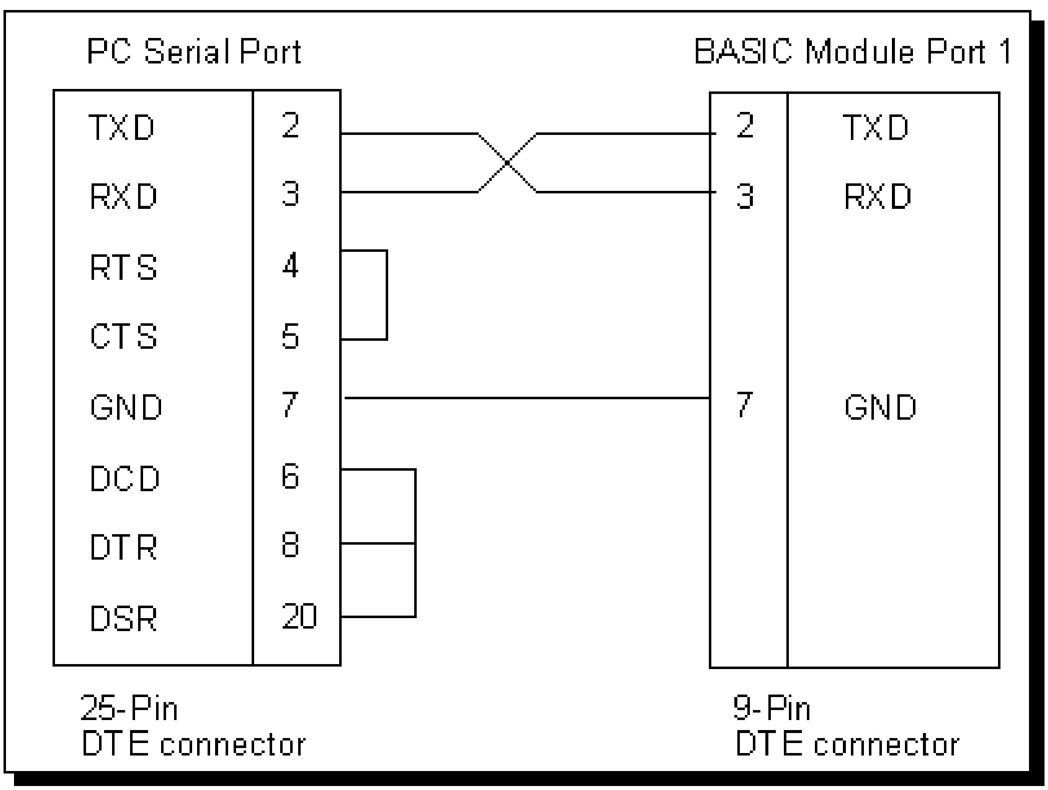
The signals flow between two types of interface ports, data communication equipment (DCE) and data terminal equipment (DTE). The pin names are the same for both DCE and DTE equipment, however, the direction of signal flow is reversed.

RS-232 DTE and DCE Pin Names and Signal Flow

Pin	Abrev.	Name	Signal Direction		Description
			DCE	DTE	
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	Input	Output	
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

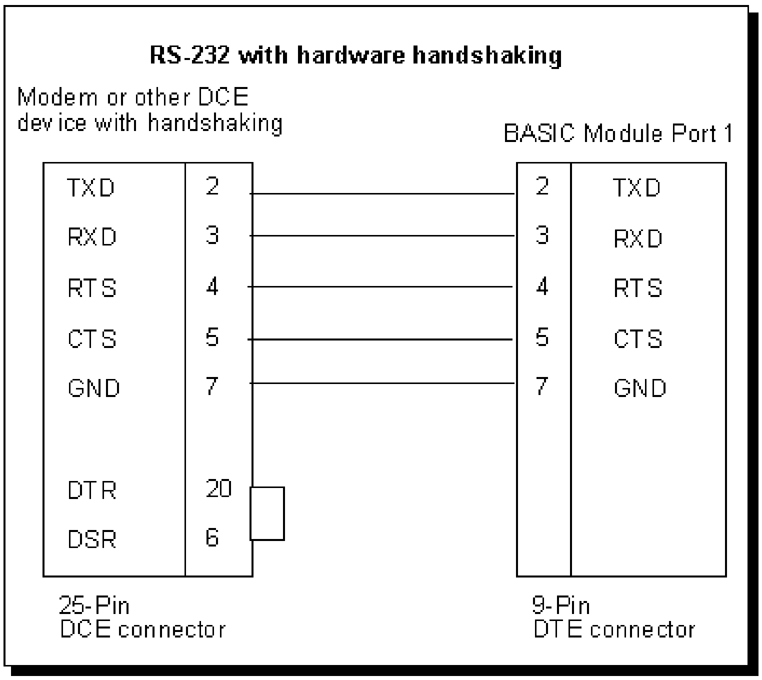


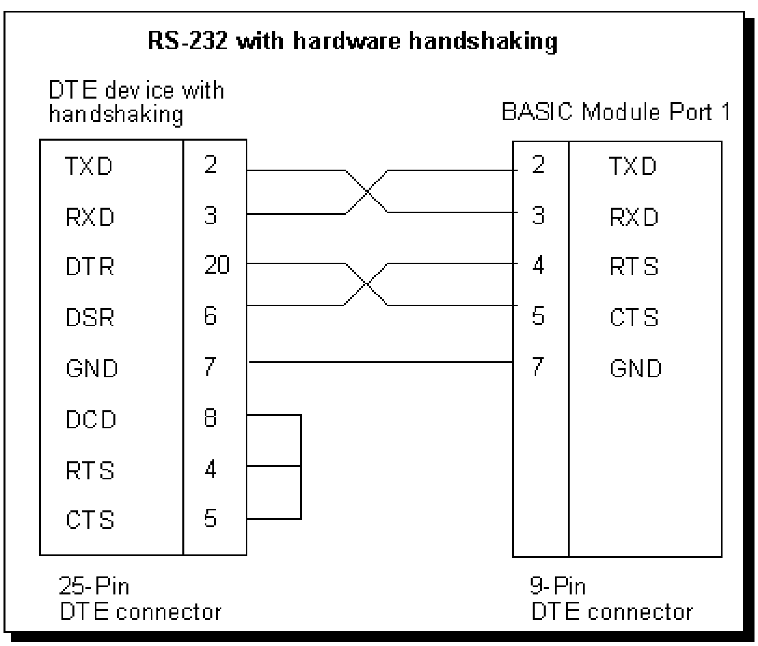


IDENTIFYING A COMMUNICATION PORT AS DCE OR DTE

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 WITH HARDWARE HANDSHAKE





RS-422/485 STANDARD

The RS-485 transceivers on CoProcessor's so equipped are compatible with both RS-422 and RS-485 signals.

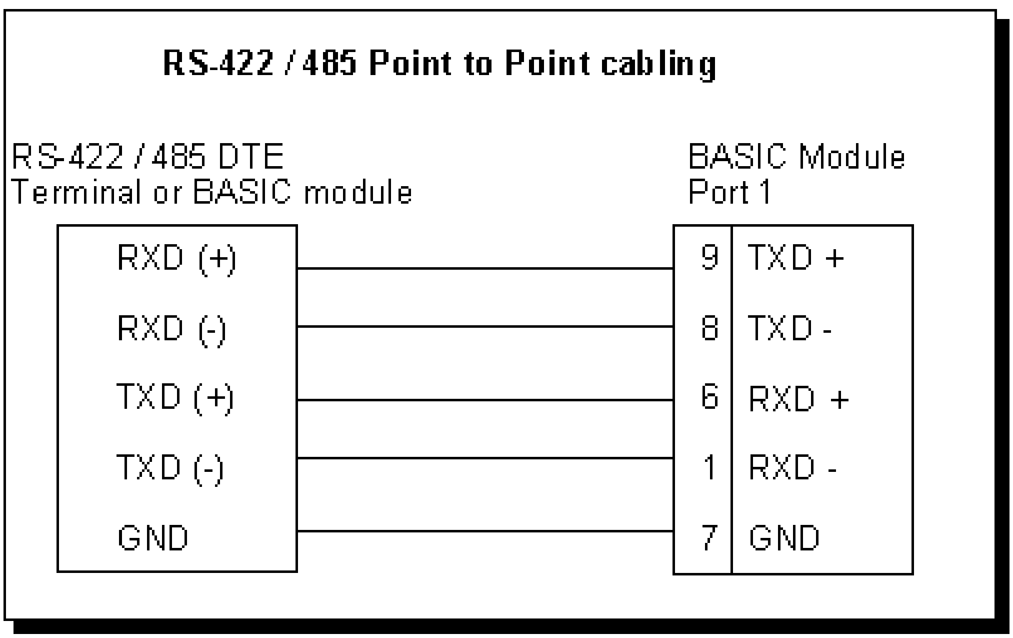
RS-422 uses high current differential outputs and is specified to 4000 feet at 10 Megabaud. Lower speed communications, such as 19.2K baud, may use substantially longer cables.

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-422/485 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.

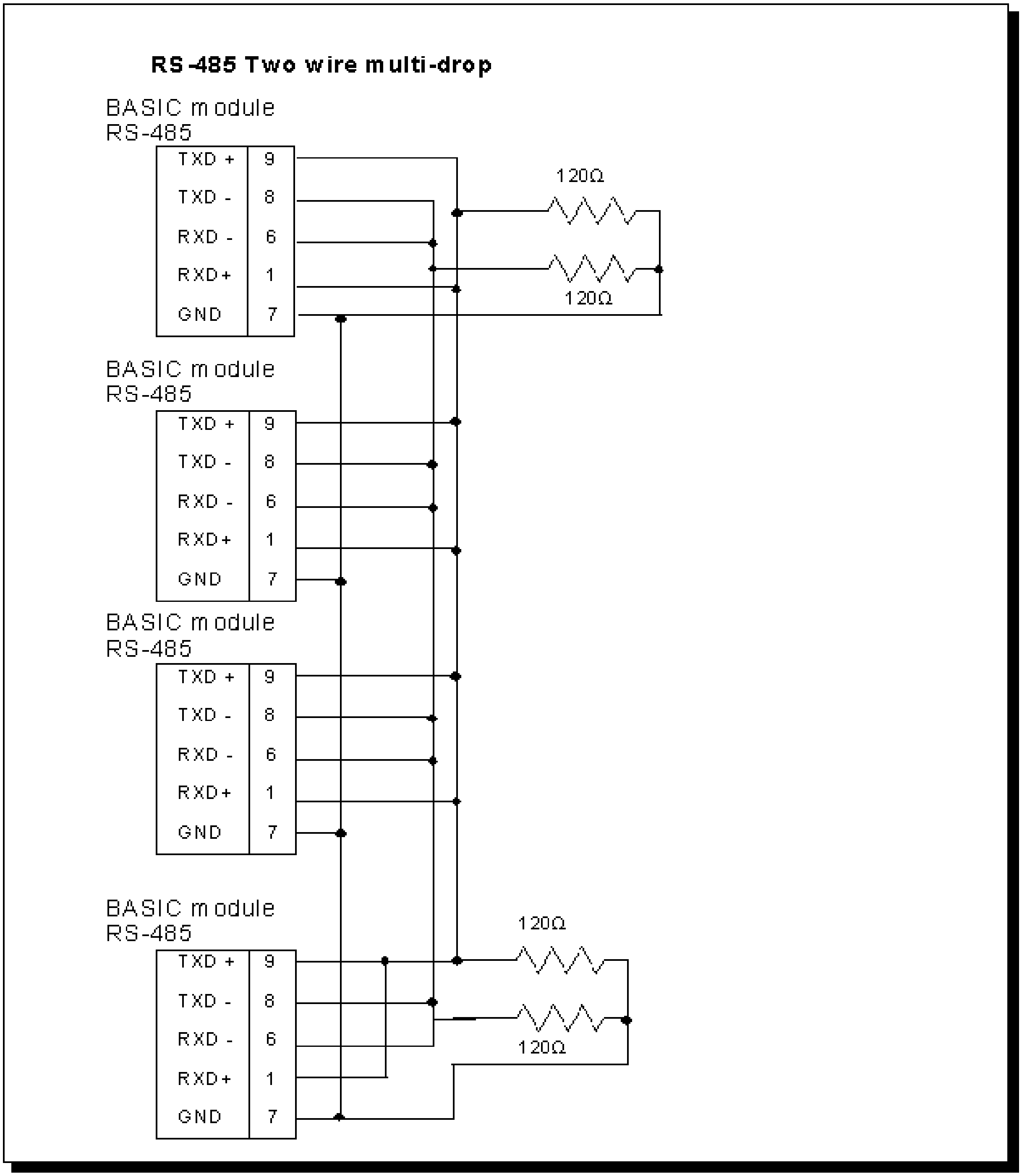
To enable the RS-422/485 transmitters only when PRINTing, use SETPORT to select multi-drop mode "M". Use the multi-drop option when the CoProcessor is a slave in a master/slave configuration or when a peer to peer configuration is required.

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 CoProcessor 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 slave devices on the network.

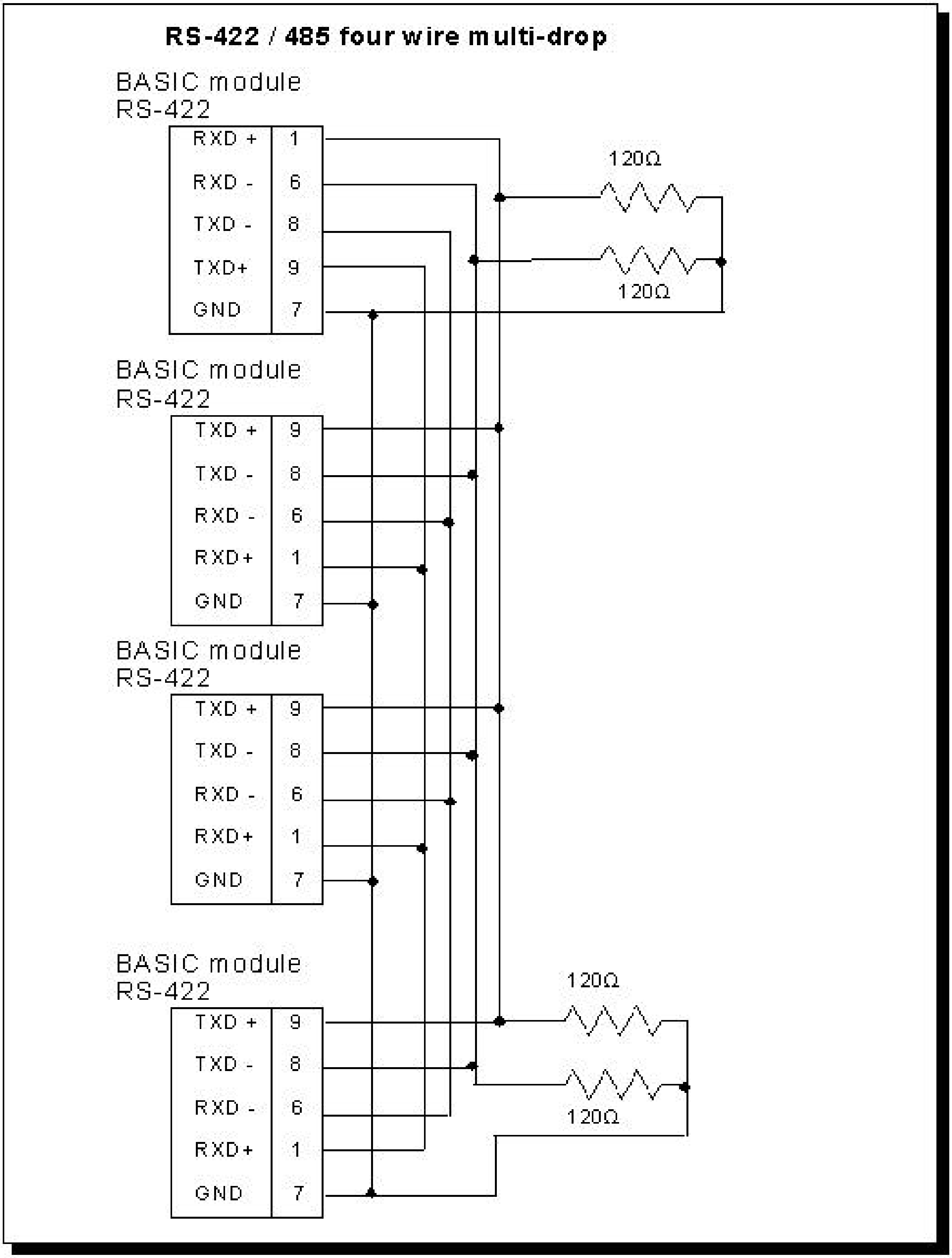
Example: Configure Port 1 for 9600 baud, no parity, 8 bit word, 1 stop bit, software XON/XOFF handshaking, and multi-drop RS-422/485 mode.

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

RS-485 TWO WIRE MULTI-DROP



RS-422 FOUR WIRE MULTI-DROP



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 Ω resistor in series with the shield and the ground connection will reduce noise producing ground currents.

Connecting Cables and Line Termination

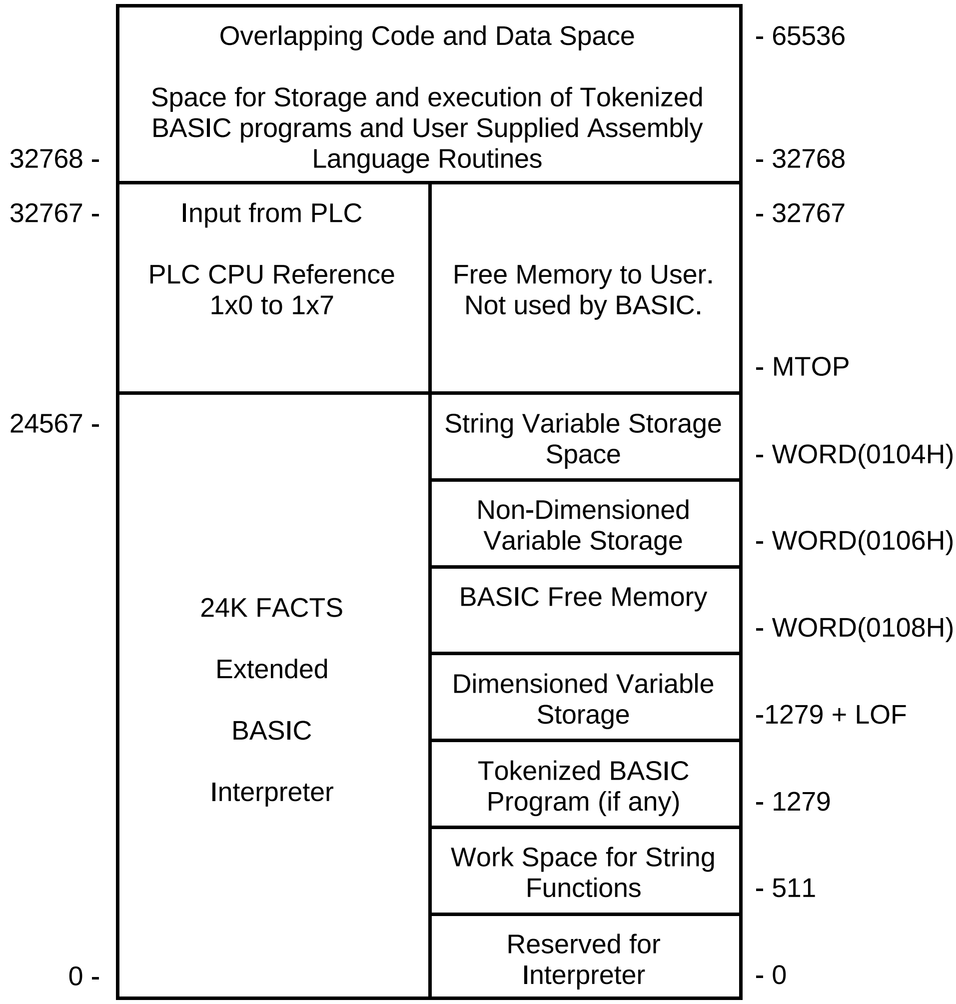
A dual twisted pair plus ground connection is recommended for 4-wire RS-422 networks. Proper termination of the balanced transmission line is required to prevent data errors. A typical AWG 22 solid wire with .060 inch plastic cover, twisted 4.5 times per foot has a characteristic impedance of about 120 Ω. Thus the selection of the two 62 Ω line-to-ground terminating resistors. Line-to-ground termination is preferred to the often shown line-to-line 120 Ω 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 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.

APPENDIX G: MEMORY MAPS

64K MEMORY MAP - F3-AB64



128K MEMORY MAP - F3-AB128, F3-AB128-T, F3-AB128-R