Preliminary Data Sheet

11/19/03

DTL-IFB-485

Interface Board for Serial-Input Electronic Loads

Features

- **Simplifies interfacing to DATEL's DTL2A, DTL2A-LC, DTL3A & DTL4A, 100W serial-input electronic loads**
- **Connects to PC's *serial* port using RS485 adapter card**
- **Simplifies burn-in rack and load bank connections• Microcontroller with a four-wire RS-485 interface**
- **Update loads individually or simultaneously**
- **Red/Green LEDs for fault/unit enabled**
- **WinBurn™ ActiveX Controls**

Ordering Information

MODEL	DESCRIPTION
DTL-IFB-485	RS-485 serial Interface Board with A/D for serial-input electronic loads. WinBurn™ Active X Control available upon request (DTLx485)
DTL-DB10-6	6' Cable from RS-485 port to DTL-IFB-485

General Description

The DTL-IFB-485 uses a microcontroller with a 4-wire RS-485 interface to facilitate programming the DTL2A/2A-LC/3A/4A serial-input (isolated) loads using simple ASCII command strings through the serial port of a PC. THE DTL-IFB-485 plugs in to a DTL2/3/4 load and provides an address for the load (range 0 to 255) set by two rotary hexadecimal switches on the DTL-IFB-485. The standard RS-485 specification provides for up to 32 devices connected up to 4,000 feet of cable length. The DTL-IFB-485 operates at either 9600 or 1200 baud. The DTL-IFB-485 uses special high-input impedance receivers, allowing 256 devices to be driven by a single RS-485 port. A driver card can be installed in a PC, just as a COM port (RS-232) would be and is programmed identically to a normal RS-232 port. RS-232/RS485 interface adapters can also be used. The DTL-IFB-485 provides a convenient way to manage up to 256 electronic loads dispersed over a large area, such as might be typically encountered in a burn-in system or battery load/life test system. The DTL-IFB-485 provides a 12-bit A/D option that can read back the load voltage.

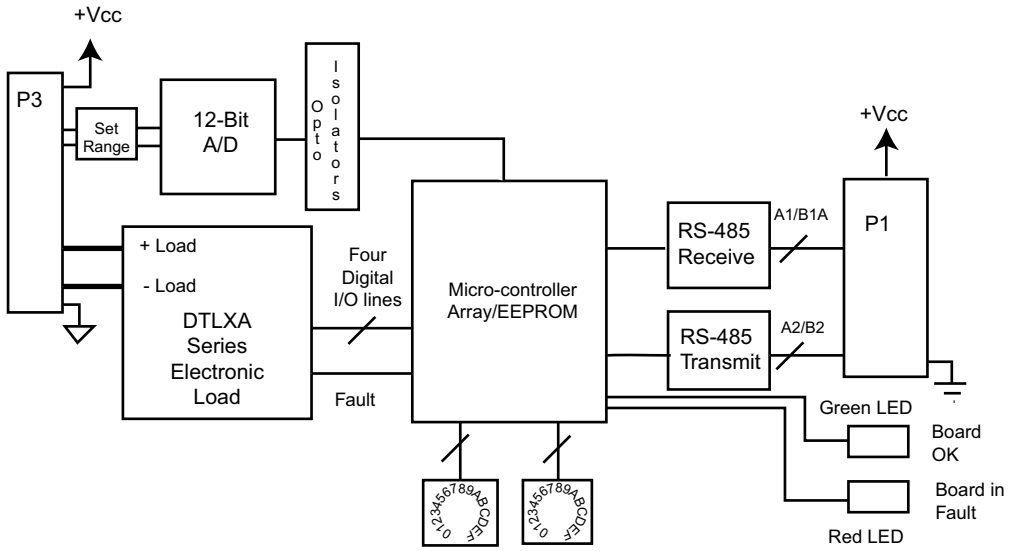
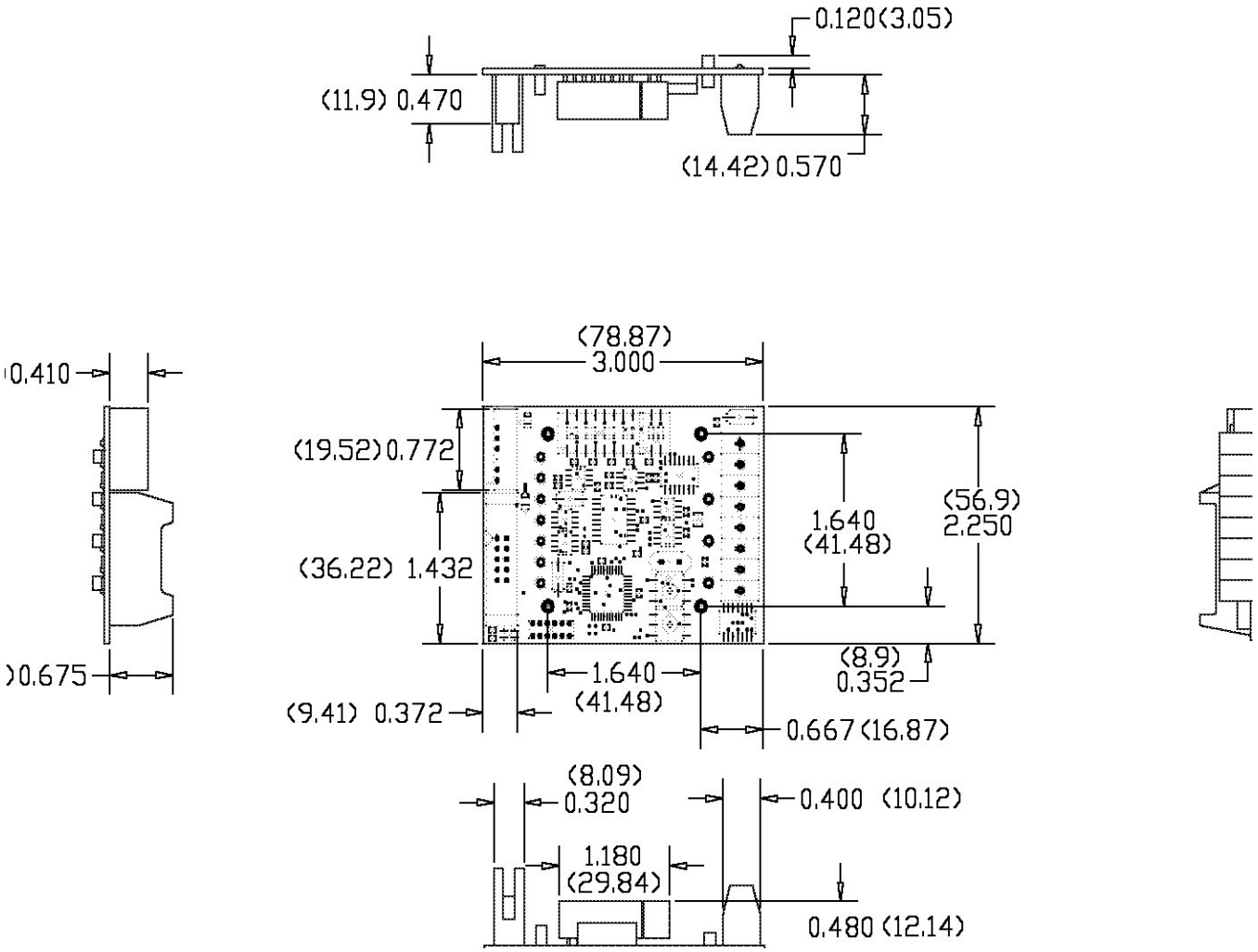


Figure 1. Simplified Schematic

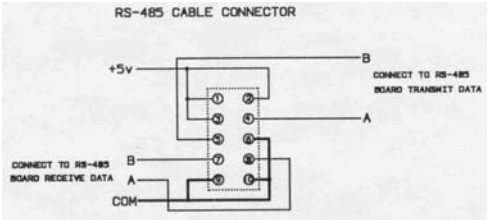
DATEL, Inc. Mansfield, MA 02048 (USA) * Tel: (508)339-3000, (800)233-2765 Fax: (508)339-6356 * Email: sales@datel.com * Internet: www.datel.com

Mechanical Specifications

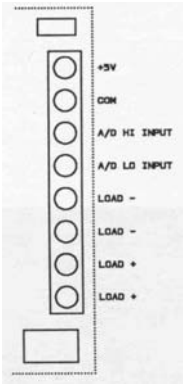


Note: (1) dimensions in inches (mm) (2) Mounting Holes 0.125 (3.16) diameter are on 1.640 (41.48) centers

I/O Connections: P1	
Pin	**Function**
1	+5 Volts Supply
2	+5 Volts Supply
3	+5 Volts Supply
4	RS485 Receive "A1"
5	RS485 Receive "B1"
6	Digital Ground
7	RS485 Transmit "B2"
8	RS485 Transmit "A2"
9	Digital Ground
10	Digital Ground



I/O Connections: P3	
Pin	**Function**
1	+5 Volts Supply
2	Digital Ground
3	A/D Hi Input
4	A/D Lo Input
5	Load -
6	Load -
7	Load +
8	Load +



Digital Input	Min.	Typ.	Max.	Units
Digital Inputs (pins 4 and 5 of Connector P1):				
V_{IL}			0.8	Volts
V_{IH}	2.0			Volts
I_{IL}			-0.8	mA
I_{IH}			1	mA

Digital Outputs	Min.	Typ.	Max.	Units
Digital Inputs (pins 4 and 5 of Connector P1):				
V_{IL}			0.4	Volts
V_{IH}	2.4			Volts
I_{IL}			-4	mA
I_{IH}			4	mA

Analog Input				
Input Voltage Range		4.096, 8.192, 40.96 (See A/D Setup Section)		Volts
Resolution		12		bits
4.096V		1		mV
8.192V		2		mV
40.96V		10		mV
Differential Linearity Error		±3/4		LSB
Integral Linearity Error		±2		LSB
Accuracy (not calibrated)		±1		% FSR
Accuracy (calibrated)		±0.1		% FSR
Offset Error (not calibrated,)		±3/4	±3	% FSR
Offset Error (calibrated)				% FSR
Gain Error (not calibrated)		±2	±8	LSB
Gain Error (calibrated)				% FSR

Power				
+5 Volts Supply (pin 1, 2, 3 of P1)	+4.75	+5.0	+5.25	Volts
Current (pin1,2, 3 without DTLXA Load		+100		mA

Environmental				
Operating Ambient Temperature Ta, where no derating required. Natural Convection, vertical mount				
Storage Temperature	–40		+105	°C
Humidity (Non-condensing)			95	**%**
Altitude Above Sea Level		10,000		feet

Physical		
Dimensions	2" x 3" x 0.675" (50.8 x 76.2 x 17.2mm)	
Socket extension below board	0.120	inches
Pin Material	Brass, solder coated	
Isolation, ± Load to IGround	500	Volts
Isolation, any pin to either grnd	500	Volts
Isolation, resistance	100	Mohm
Mounting	Four PCB-through-holes #4-40 clearance	
Weight	_ _ _ ounces (_ _ grams)	

Timing	Min.	Typ.	Max.	Units
Issue command/recieve response:		**3**0		msec

Absolute Maximum Ratings

These are stress ratings. Exposure of devices to any of these conditions may adversely affect long-term reliability. Proper operation under conditions other than those listed in the Performance/Functional Specifications Table is not implied.

Power Supply Voltage (pin3):	5.5 Volts
Digital Input Voltage (pins 4,5,6,7):	5.5 Volts
Output Reverse-Polarity Protection:	No protection
Output Overvoltage Protection:	No protection
Storage Temperature	–40 to +105°C
Lead Temperature (soldering, 10 sec.)	+300°C

USING THE DTL-IFB-485 SERIAL ADAPTER WITH DTL2A FAMILY 100W SERIAL ELECTRONIC LOADS

Overview

The DTL-IFB-RS485 allows up to 256 DTL2A series of serial 100W loads to be operated on a single RS-485 half duplex cable at distances up to 5,000 feet from the controlling computer. The DTL-IFB-485 uses serial communication and can be connected to a standard PC RS232 COM port with an appropriate RS232/RS485 adapter. Alternatively a plug-in RS485 board (PCI/ISA) can be inserted in the control computer. Control of the DTL2A Series load is via ASCII string commands with ASCII responses. The DTL-IFB-485 includes a small microcontroller that makes the protocol conversions between the ASCII commands and the DTL2A2 Series load. Commands and responses are detailed in Command Syntax section herein.

The DTL-IFB-485 also includes an uncommitted and optically isolated 12 bit analog input channel. Dip-switches on the board allow selection of unipolar ranges of 4.096V (resolution 1mV), 8.192V (resolution 2mV) and 40.96V (resolution 10mV). The analog input channel might typically be used for measuring the voltage across the DTL2A load from the U.U.T. or the current drawn using a shunt, but since it is a free isolated 2 terminal input, it can also be used for any other purpose.

The DTL-IFB-485 requires +5V power at 100mA. Each interface board includes screw connections for the 5V power and the common signal cable can also distribute power to multiple units.

RS-485 Communication

RS-485 is a similar serial communication standard to the more common RS-232 found on all personal computers (COM1 & COM2 ports). The primary difference is that RS-232 uses single ended transmit and receive data lines with a ±5V to ±15V signal level and as a result, is limited to about 25 feet transmission distance, whereas RS-485 uses balanced differential transmit and receive lines with differential thresholds of +/-200mV and up to +12V and -7V of common mode voltage. Because of the differential configuration, RS-485 is much less susceptible to noise than RS-232 and is specified to operate up to 5,000 feet away from the controller. True RS-485 is a full duplex system that transmits and receives on a common 2 wire pair with a ground wire (3 lines). The DTL-IFB-485 uses a variant of this that has separate transmit and receive pairs (often referred to as RS-422) plus ground and power distribution wires. To facilitate connecting multiple DTL-IFB-485 units and loads to one controller, a 10 way, 0.050" flat cable, is used to interconnect units and the controller:

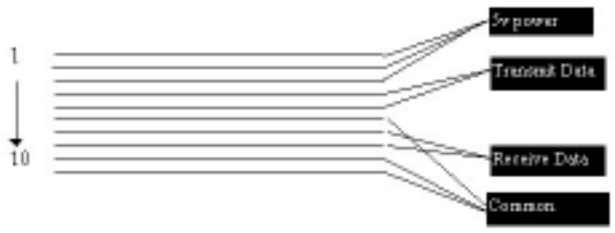


Figure 1. RS-485 Signal Lines

Six of the 10 lines are used for power distribution, and the remaining 4 are used as two pairs for the transmit and receive RS-485 differential signals. The use of standard flat 10 way I.D.C. cable allows socket connectors to be easily pressed on the cable at appropriate positions for the loads. Although 6 conductors are dedicated for power distribution, standard 28AWG cable has a resistance of 55 ohms/1,000 feet so the voltage drops incurred sending 5V power over a few tens of feet will be prohibitive. For longer distances, 5V power should be provided locally at each DTL-IFB-485 and connected via the screw connector.

Standard RS-485 provides for a fan out of 32 units on a bus. The DTL-IFB-485 uses special low power driver/receivers which only present a 1/8 unit load. Consequently up to 255 devices can be connected to a bus.

The data transmission rate is set on the DTL-IFB-485 using a 2 position dip switch. Baud rates of 9,600, 2,400, 1,200 and 300 can be selected as shown:

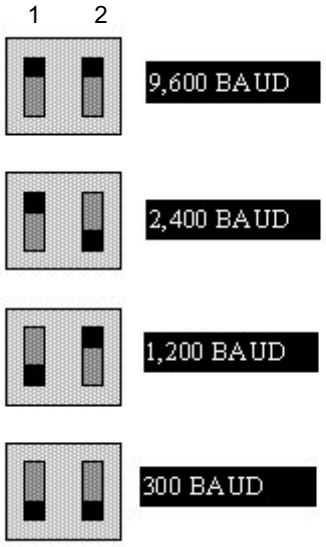


Figure 2. Data Transmission Speed Set by Dip-Switches

Generally using 9,600 baud will give the fastest response and will be the best choice. The data format should be 8 data bits, 1 stop bit and no parity.

For long distance transmission, over 100 feet, terminator resistors will improve the performance of the RS-485 bus. There are several ways of providing termination matching, but a common method is to place a 120 ohm resistor across the differential transmit and receive lines at the far end of the bus. The same configuration can be added to the transmitting end, but no more than 2 sets of 120 ohm resistors should be added to a single line.

The schematic of a simple RS-232 to RS-485 converter is shown in Figure 5, commercial equivalents can be purchased from many sources.

RS485 Communication (Cont'd)

Initial Set Up

If you are using a RS232 - RS485 adapter on a standard COM port, connect the transmit A (or -) terminal to line 4 of the flat cable, transmit B (or +) to line 5, receive A (or -) to line 8, receive B (or +) to line 7 and ground to line 6, 9 or 10. If you wish to connect the 5v power through the cable, connect the +5V to lines 1,2 & 3 and the common to lines 6, 9 & 10.

Each DTL-IFB-485 has two hexadecimal rotary dip switches that set the unit address. Each interface board must be set to a unique address, no two or more units on the same cable can have the same address otherwise data collisions will occur on the RS-485 receive bus. This will not cause any physical damage but will result in garbled data. Select any address(es) that is (are) convenient. Example: A123 is equal to 7B in Hex so rotary switch SW2 is set to "7" and SW3 is set to "B".

Set the baud rate dipswitches to 9,600 baud (as described in the previous section). If you wish to use a lower baud rate, then make changes in the terminal software settings (below) to match your selected speed.

Turn on the 5V power to the DTL-IFB-485's and interface adapter if used. To test things out, a standard dumb terminal program such as Windows Hyperterminal (step-through example starts on page 8 herein) can conveniently be used.

Command Syntax

The command syntac of the DLT-IFB-485 is very specific and must be adhered to for satisfactory operation. The microcontroller on board the DTL-IFB-485 has limited resources available for "command parsing", so its up to the user to provide properly structured command strings. All commands are case insensitive (i.e. A000_?R is the same as a000_?r). All commands must be followed by a carriage return (ASCII code 13) to execute.

The delimiter character (underscore character_) is critical between address and data or command. A1232345L (with no delimiter) will not execute and will produce an Error response because the data is in the wrong position in the string (it looks like 345L), whereas A123--2345L (with 2 delimiters) will produce an Error response as the parser is expecting a number in the 6th character position. The delimiter character can be any with the exception of carriage return (ASCII13) or backspace (ASCII 11?). Convenient delimiter characters that the parser ignores are typically underscore_, #, x (any letter, number or symbol on the keyboard) or Space. The parser has no capability of throwing away leading spaces, zeros, etc.

Any command that includes an address i.e. is unit specific, will generate a response, typically OK, FAULT or ERROR but certain global commands such as C (clear), L (load) or G_1234 (global data load) that affect all DTL-IFB-485's on the cable will not generate any response although they will light the activity LED's of all units. The family of commands and responses is as follows:

1. A123_3456
Writes decimal data 3456 to load with address 123. Stores data, but does not load the output register. Unit responds with OK, FAULT or ERROR. OK if operation is successful, FAULT if load is out of compliance (e.g. zero voltage applied) or ERROR if error in data (non-numeric or > 4095). No response if address is invalid (no module) or out of range >255. If address only is sent i.e. Axxx, a unit with the selected address will respond with OK (this is a way to find if units are correctly connected to system and responding).

2. A123_3456L
Writes decimal data 3456 to load with address 123 and loads DTL2A Series output register updating output current. Error response as in #1.

3. A234_?S
Returns fault status from #234. Responds OK or FAULT. FAULT condition corresponds to out of compliance or no voltage at output terminals of the load.

4. A006_?V
Returns A/D input channel voltage in 5 digit decimal string, range from 0.000 to 4.095/8.190/40.95V (depending on selected range).

5. A017_?R
Returns range selected on A/D in single digit string. For example, responds 4.095 CAL if range calibrated or 4.095 UNC if range is uncalibrated (may be +/-3% error). Range is selected by dip switches on DTL-IFB-485 board (4.095V or 8.190V or 40.95V)

6. A123_?D
Returns the value of the last data sent to the DTL2A Series load (e.g. 2037). Note that if you send A123_1234 without an L at the end to load the data, the ?D command will return the last data actually transferred to the load, not 1234.

7. L
Simultaneously loads all modules with previously stored data. There is no return response to this global command.

8. C
Simultaneously zeroes output of all modules. There is no return response to this global command.

9. G_0000
Loads all units with same data, regardless of address. There is no response to this command. Command will not execute if data > 4095.

10. A123_CAL
Performs calibration (Test/setup purposes only). Select desired range on dipswitches on DTL-IFB-485 board. Connect a voltage calibrator to load outputs and using a dumb terminal or terminal emulation program enter the command and follow prompts. On completion, the range will be calibrated and the calibration constants (zero & gain) will be permanently stored in internal EEPROM. This command must
be performed for each range separately to calibrate all ranges. All command strings must end with a Carriage Return (CR = 13 decimal) to activate the command.

Command Syntax (Con'td)

When operating from terminal emulation software, you can type in BACKSPACE to edit out any unwanted or erroneous characters before pressing ENTER.

There is a typically a delay of a few tens of milliseconds after issuing a command and receiving the response. When operating from a program, allowance should be made for the fact that responses are not instantaneous.

Typical Program Sequences

In typical burn in or test set up configurations, there will be multiple loads and units under test connected to the RS-485 cable. The C (clear) command provides a way of setting every load to zero either initially or for shutdown or an emergency.

In some systems, it will be desirable to set all the loads to the same value. The G_xxxx command is the easiest way to do this. If different load settings are required for each U.U.T. then either the Axxx_yyyyL or Axxx_yyyy commands will do the job. Sending an Axxx_yyyyL command will immediately set the addressed load to the data value yyyy on execution of the command. Omitting the L (load) character at the end of the command and just sending Axxx_yyyy has the effect of pre-loading the data yyyy in the addressed load but not changing the output current value from its existing state. Sending the L (load) command will then subsequently update the load. Typically you can pre-load all the loads with desired data and with a single L command update all the units together.

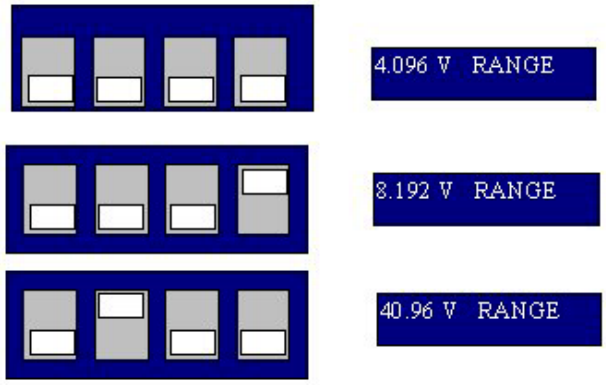
The Axxx_?S command will return the status of a load. If it is operating normally, it will return OK but if the voltage applied to the load is below its compliance limit (typically 0.6V - 2.5V depending on DTLXA Series load type) then FAULT will be returned. This is a simple way to determine if the U.U.T. has failed. Connecting the A/D channel up to the load output allows direct reading of the voltage to better than 0.1% precision using the Axxx_?V command. This way the Unit Under Test (U.U.T.) output voltages can be monitored precisely and if desired, controlling software can be configured to perform data acquisition functions.

Simply sending an address e.g. Axxx followed by a carriage return will return an OK response from a unit set to that address. If the address is not present on the system, no response will be received. This provides a way of initially checking the system by polling for active units. Similarly the Axxx_?R (range) command returns data on the range setting of the A/D channel and its calibration status.

Setting and calibrating the A/D Channel

As supplied, the DTL-IFB-485 will have been calibrated and will not normally require recalibration for 3 months - 1 year. The DTL-IFB-485 board does not include any adjustment potentiometers, instead calibration constants for each range are stored in non-volatile EEPROM in the microcontroller. Updating these constants can be performed using the Axxx_CAL command which is most easily performed using a PC with terminal emulation software or a dumb terminal.

First select the desired range on the A/D channel dipswitch as shown below:



Note that if any any other combination of switch settings is selected, you will receive a BAD RANGE response after the Axxx_CAL or Axxx_?R commands. Connect a voltage calibrator to the A/D channel inputs (note if these happen to be connected to the load output, the load is automatically set to zero during this procedure and will not load the calibrator). The calibrator should be capable of 0 - 20v output in millivolt steps. Type in the Axxx_CAL command. You should receive a response:

Input +0.010V and press ENTER or ESC to exit

The input voltage requested will vary with the range (4.096V range shown). Pressing ESC will abandon calibration and you will receive an ERROR response. Also when you run the Axxx_?R command afterwards, you will receive a response 4.096 UNC - the UNC denoting that this range is no longer calibrated. If the range is uncalibrated, the raw A/D data is returned and because of zero offset and gain errors, is likely to be only 1% accurate.

Pressing the ENTER key will load the zero offset constant into EEPROM and proceed to the next step unless you have applied the wrong voltage. A check is made that the input voltage is within ±100% of the requested value, if not a BAD INPUT response is returned and calibration is abandoned. Assuming all is correct you will receive the next response:

Input +4.000V and press ENTER or ESC to exit

If the calibrator input is more than ±2.5% away from the requested value, you will receive the BAD INPUT message and calibration will be abandoned. Otherwise, you should receive the OK response and this range will then be calibrated. This can be checked with the Axxx_?R command which should respond 4.096 CAL (in this case). Using the Axxx_?V command and different calibrator settings, you can check other input voltages to your satisfaction. Note that calibration must be repeated on each range to calibrate all ranges, although this is unnecessary if you operate the DTL-IFB-485 on one fixed range.

Programmed Operation

Although most of this note describes operation with a dumb terminal, you can obviously issue commands from a program written in any language of your choice. All that is important to observe is that the command strings are correct in syntax and of the right length. For instance in BASIC:

```
OPEN "COM1:9600" AS #1:    opens COM1 port for I/O
A$ = "A012_?V":            setup command string to read A/D
PRINT #1, A$:              send command (note PRINT adds a CR)
PAUSE 20:                   wait 20 milliseconds
INPUT #1, X$:              read result into X$
```

Key points to note are that you must send an ASCII carriage return after the command string (character 13). The DTL-IFB-485 will not start parsing the command until it receives a carriage return or the command string is greater than 10 characters long (which should not happen in normal operation).

The input buffer length of is limited to 10 characters. Commands longer than this may be executed or ignored (e.g. A123_1234LXXXXXXX will be treated as A123_1234L, but some of the additional characters may end up in the leading positions of the input buffer and mess up the next command, requiring resending of the command.

After sending only a carriage return with line feed, the next command will be skipped and may need to be resent. Remember that any addressed (unit specific) command will produce a response.

After the command has been sent, it takes a few tens of milliseconds for the DTL-IFB-485 to process it and respond. Don't try and read back data immediately. Also in the event that there is no response, you need some form of timeout so that your program does not hang waiting for input.

Coding Table

Unipolar	Input Ranges			ASCII Code	Binary Equivalent
Scale	0 to +4.096V	0 to +8.192V	0 to 40.96V		
+FS - 1 LSB	+4.095V	+8.190V	+40.86V	FFF	1111 1111 1111
7/8 FS	+3.584V	+7.168	+35.84V	E00	1110 0000 0000
3/4 FS	+3.072V	+6.144V	+30.72V	C00	1100 0000 0000
1/2 FS	+2.048V	+4.096V	+20.48V	800	1000 0000 0000
1/4 FS	+1.024V	+2.048V	+10.24V	400	0100 0000 0000
1/8 FS	+0.512V	+1.024V	+5.12V	200	0010 0000 0000
1 LSB	+0.001V	+0.002V	+10.00V	001	0000 0000 0001
0	+0.000V	+0.000V	+0.000V	000	0100 0000 0000

Utilizing Windows™ "HyperTerminal" to Control the DTL-IFB-485 (RS-485 Interface Board for 100 Watt Serial Input Loads)

Many users will take the WinBurn-485X, Active X software component, and design their own installable software program to use in controlling the DTL-IFB-485 and DTLXA Series of Electronic Loads. An alternative approach outlined herein, is to use a "dumb terminal" or a "terminal emulation program" to send and receive commands to and from the DTL-IFB-485.

If you don't have a dumb RS232 terminal handy (who does these days) a terminal emulation program running on a PC, will do the same job just as well. Windows 95/98 provide a standard terminal emulation program called HyperTerminal (see Figure 3), which is usually located in the Accessories folder after Windows installation.

Figure 3. Windows 95/98 HyperTerminal

In the days of DOS, there were many terminal emulation programs available (example: Bitcom). If you want to operate under DOS, these will work just fine.

Figure 4. DTL-RS232-485 Adapter Card

This technical description explains how to configure HyperTerminal to operate with the DTL-IFB-485. An RS232 to 485 adapter (See Figure 4, DTL-RS232-485 product, or Figure 6, for schematic) can make it easy to connect to an RS-232-standard serial communication port (see Figure 5) of a personal computer (PC).

COM Port >>>>>>

Figure 5. Typical PC "Com" Port

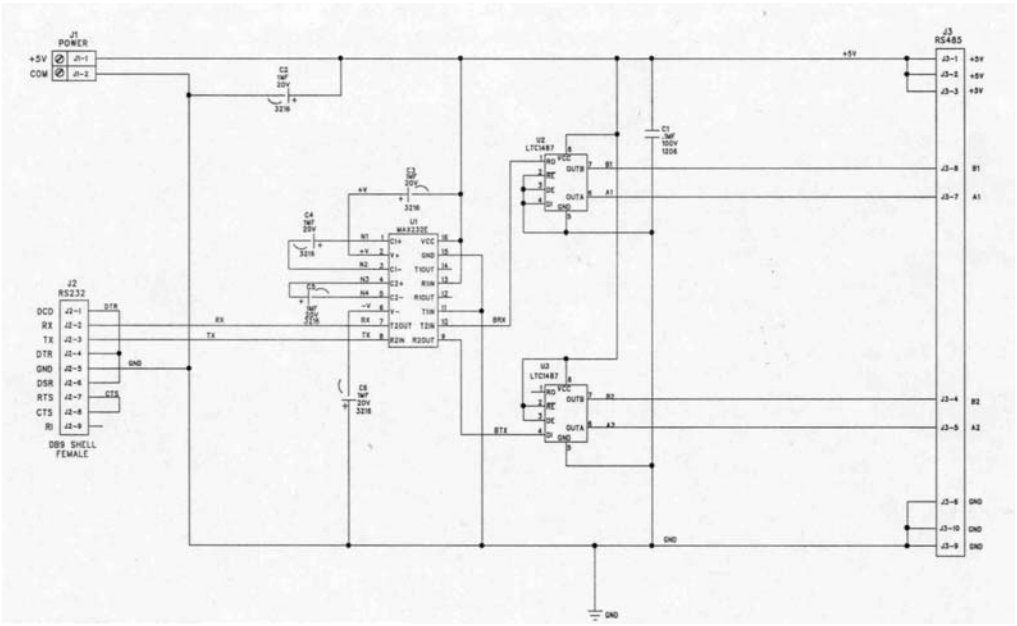


Figure 5. DTL-RS232-485 Adapter Card Schematic

1. From the Windows START button select PROGRAMS, then ACCESSORIES, then HYPERTERMINAL. A window called HyperTerminal will open up. Look for the ICON with Hypertrm.exe beneath. Double click on it to start HyperTerminal.

2. HyperTerminal will first open up a window called Connection Description (see Figure 7). HyperTerminal is a full featured terminal emulation program that can work with dial up connections and modems, and is designed to store all this connection-specific data. We don't need a lot of these features, for what we want to do which is to create a dumb terminal connecting directly to one of the COM ports, but we have to go along with HyperTerminal's requirements to get there. In the Connection Description window, enter a name that you like for the configuration that you are about to set up e.g. "9600 Dumb" and also choose an icon that you like the look of. Click the O.K. tab.

Figure 7. Connection Description Screen

3. A window titled Phone Number will now open up. Well, we don't have a phone number for a dumb terminal, so go down to the Connect Using: box and select Direct to ComX where X can be 1,2,3 or 4 (see Figure 6).

Figure 8. Direct to Com Port Setting

If you are using a RS232-485 converter on one of the computer internal COM ports then enter the appropriate COM port. Most desktop computers provide 2 internal COM ports, COM1 & COM2 (laptops may have only one) that are brought out on the rear as 25 or 9-pin connectors . If you are using a plug-in RS-485 card or an RS-232 serial port expansion card, these ports will probably end up being set to COM3 and/or COM4, whatever Window's device manager has permitted as satisfactory for installation without resource conflict. Enter the appropriate COM port and click the O.K. tab.

4. The next window that opens up is titled Comx Properties. The first box is Bits per Second. Select the speed (9600, 2400, 1200 or 300) to correspond with the speed selected on the 2 position baud rate dipswitch on the DTL-IFB-485 (usually 9600 is the best choice). Set Data Bits to 8, Parity to None, Stop Bits to 1 and Flow Control to None.

Figure 6. Direct to Com Port Screen

The Advanced button controls FIFO buffer settings for the COM port UART, the speeds that we are using are so slow that these settings are fairly irrelevant, choosing default will work fine.

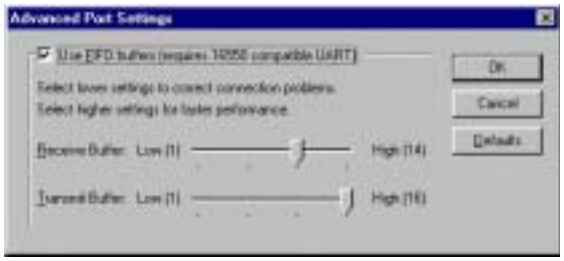


Figure 7. Advance Port Setting Screen

After selecting O.K. you will finally reach the terminal window (see Figure 8), but there are still more configuration details to set.

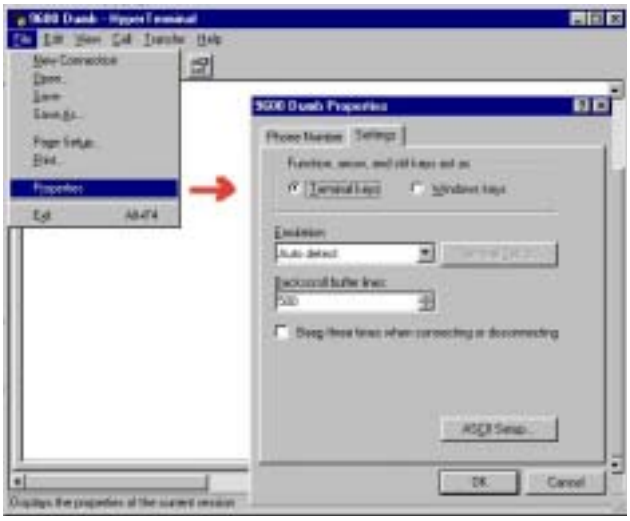


Figure 8. Terminal: Properties Settings

5. From the terminal window, select File and then Properties (see Figure 8). Select the Settings tab. Set Terminal Keys on, Emulation to Auto Detect, Backscroll Buffer lines to 500 (all these may be default values).

Under ASCII settings (see Figure 9), set Send line ends with line feeds on, and echo typed character locally on, line delay 0 and character delay 0. Under ASCII receiving set Append Line Feeds to Incoming Line Ends on, force 7 bit ASCII off and line wrap on. Select the O.K. button and from the terminal window select exit (File/exit) or the top right corner X button (i.e. exit the application).

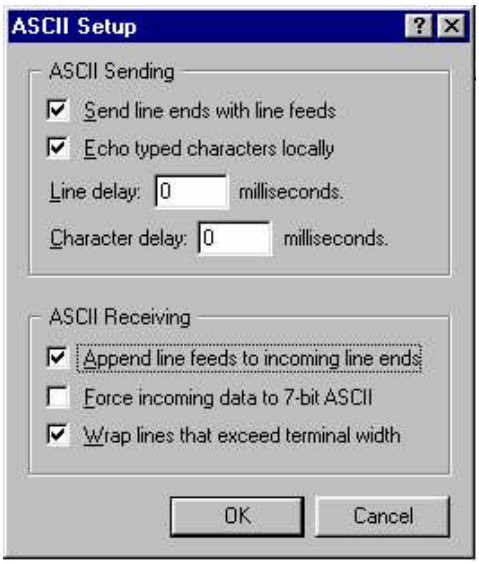


Figure 9. Terminal: Properties Settings-ASCII

You will receive a "You are currently connected" prompt “Do you want to save session 9600 Dumb?” (or whatever name you gave it). Respond by clicking yes. You will exit to the HyperTerminal window and should see an icon with “9600 Dumb.ht” under it (see Figure 3). Clicking on it will start HyperTerminal up with all the settings that we have entered. The reason for exiting and re-starting HyperTerminal is that it does not seem to pick up all the configuration settings unless you do it this way.

6. At the terminal screen, you can type in any of the DTL-IFB-485 commands e.g. A000_?V and the response should return from the board. Next time you run HyperTerminal, you can just click on the “9600 Dumb.ht” icon to run the terminal emulation program with all your pre-selected settings.

7. HyperTerminal has an anomality in its' set up feature. If you cannot establish communication with the DTL-IFB-485 after you have followed all the preceding steps, proceed as follows:

Enable HyperTerminal’s status bar by clicking on the View menu. If you see “Connected 0:03:33” followed by “Auto Detect” and a second box also displaying “Auto Detect”, you will have a problem. The second box should display the connection speed (assume 9600 Baud) as 9600 8-N-1 (see Figure 10), but despite entering all this data in the HyperTerminal setup, the program is so hyper-intelligent, that it prefers to ignore it and auto detect the baud rate from the responding device.

Figure 10. Status Bar: 9600 8-N-1

This "auto-detection of HyperTerminal", requires the responding device to send a carriage return. This might work well with modem handshakes, but it doesn’t work with the DTL-IFB-485, which doesn’t emit any carriage returns until it is spoken to at the right speed. Surprisingly, there is no way to suppress this auto detecting of the baud rate from the configuration menus. To get round this problem, remove the connector from the DTL-IFB-485 and jumper pin 4 to 8 and 5 to 7 on the connector.

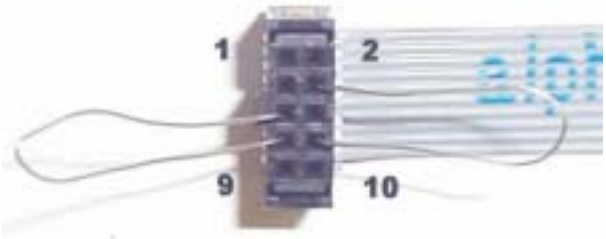


Figure 10. Status Bar: 9600 8-N-1

DTL-IFB-485 & “HyperTerminal”, cont'd

Hit Enter on the keyboard and the second “Auto Detect” should immediately change to “9600 8-N-1” (see Figure 9). Remove the jumpers and plug the connector back into the DTL-IFB-485. It should now be operational. On exiting HyperTerminal, it will prompt you to save this session and you should respond - yes. In future when you select the icon for this HyperTerminal configuration, it will automatically start with the correct baud rate configured and you will not have any further problem with auto detect. Contact DATEL if any problems...a file 9600 Dumb.ht can be provided that will contain the proper configuration setup.

8. The command syntax of the DTL-IFB-485 is very specific and must be adhered to for satisfactory operation. The microcontroller on board the DTL-IFB-485 has limited resources available for command parsing, so it is up to the user to provide properly structured command strings. All commands are case insensitive i.e. a000_?r is the same as A000_?r or A000_?R. All commands must be followed by a carriage return (ASCII code 13) to execute. Refer to page 5 herein for the DTL-IFB-485 command syntax to use when using the HyperTerminal screen.

Utilizing WinBurn™ Active X Control (DTLx485) with the DTL-IFB-485

Figure 11. DTLx485 ActiveX Control

Each shipment/lot of DTL-IFB-485 product receives an installable WinBurn™ Active X component (DTLx485) on 3 1/2" diskettes. The control lets users provide simple lines of code to send "behind the scenes" ASCII commands to the DTL-IFB-485. The DTLx485 ActiveX component speeds test and burn-in application development, by letting users customize the user graphical interface as needed. Utilizing higher-level, graphical programming languages to control the ActiveX component, allows engineers to then construct virtual test and burn-in instrument programs with objects, instead of lines of code.

Installing WinBurn™ DTLx485

Insert the WinBurn™ DTLx485 disk, and click on the setup.exe file in /Windows Explorer...or go to Start, then Run and type A:\setup.exe (or find Setup.exe with the Browse command).

Figure 12. Setup Screen Options

Most users will choose the "Typical" setup option. This installs all the required system files with the actual application. The Compact Setup option doesn't install certain system files with the application, on the assumption that these files probably pre-exist in the users Windows\system directory already (example: an application running on a desktop, that has Visual Basic installed). This provides avoids duplicating certain files, keeping the overall application file as small as possible. Users who experience problems in using a Compact Setup Install, should reinstall using the Typical Setup option. The Custom option essentially allows the user to pick from either the Typical (default) or Compact options.

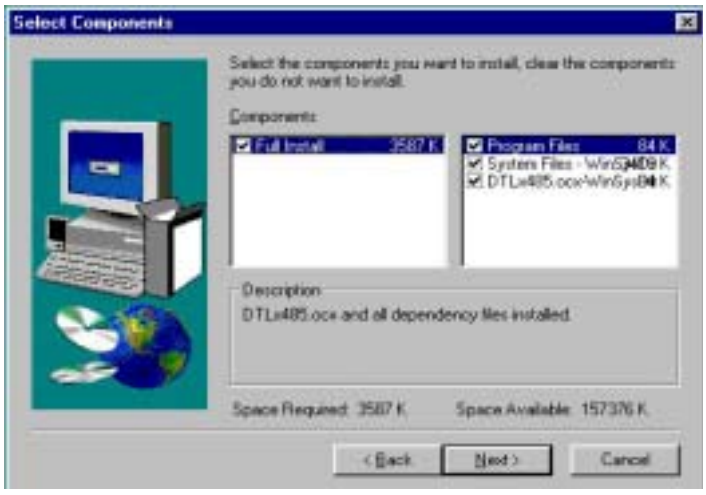


Figure 13. DTLx485 ActiveX Control Custom Setup Screen

The install program will request the second disk, and then prompt the user to Finish the install and exit, or to Launch the Application (not setup to do this presently in Beta version 1.01).

Adding DTLx485 Active X Control to Your Project's Toolbox

After installing the WinBurn™ DTLx485, you can add the control to your Visual Basic-based application as follows:

On the menu tab, select Project, and then click on Components, which will display a Components Dialog Box (see Figure _ _). Alternatively, right-clicking on the Toolbox also brings up the Component Dialog Box. To add the DTLx485 ActiveX control to your application project, click on the Browse button, and locate the DTLx485.ocx file (WinBurn™ default directory of C:\Program Files\DATEL\DTLx\DTLx485. Click on the DTLx485.ocx file to add it to the Components list. Now, either double-click on the DTLx485, or select the check box to the left of the DTLx485 name to add it to the Toolbox. Click OK to close the Components dialog box.

DTLx485ctl Class/Members

Under Visual Basic, the WinBurn™ DTLx485 ActiveX Control has the "Class" DTLx485ctl, with the following "Members" to point out:

CheckFaultStatus	Function: Returns the status of the DTLXA Series Fault Line 0 = No Fault 1 = Fault 2 = Time Out 3 = Port Not Open
ClosePort	Function: Closes the Serial Communication Port 8 = Run time error
Col	Property: Sets the DTLXA Series load position (Column = X) 0 -15
ELType (DtlType?)	Property: Sets the DTLXA Series Model Used (Current/ Voltage) 2 =DTL2A/DTL2A-LC 3 = DTL3A 4 = DTL4A
LastReadVoltage	Property: Reports the last A/D reading -1 = Error
LoadStoredCurrent	SubRoutine: Updates load to value set for load
OpenPort	Function: Opens the Serial Communication Port
PortNumber	Property: Returns/sets the Communication Port
ReadVoltage	Function: Performs A/D conversion to read voltage (presumably at load)
RequestAmps	Property: Prepares the load for the current value to be loaded with the StoreLoadCurrent and SetLoadCurrent commands
ResetAllLoads	SubRoutine: Resets all loads to zero current
ResetLoads individual load to zero current	SubRoutine: Resets
Row	Property: Sets the DTLXA Series load position (Row = Y) 0 - 15
SetAllLoadCurrents	Function: Sets all DTLXA Series Load Input Latches to the same load current value. This command requires all loads to be of the same type voltage/current (i.e., all 20A/ 50V models)
SetLoadCurrent	Function: Sets DTLXA Series Load with load current value to be set with next "StoreLoadCurrent", after RequestAmps sets value
StoreLoadCurrent	Function: Updates load to value set for load

In general: Default properties are:

Com Port = 5 (typical for 485 board...reset to 1 or 2 if using PC's com Port.

Load = 2 for a DTL2A, change if DTL3A or DTL4A

Row/Column set to 1st unit (0,0), change for multiple unit systems.

To set a particular load's current, you'll first set properties if the Default settings don't apply. Then you'll do a RequestAmps, to set the desired current value. Then, a StoreLoad Current will enter this into the 1st latch. Finally, a SetLoadCurrent (or SetAll Load Currents) command, will update the loads with a new value.

WinBurn™ DTLx485, Port Control

Developing an application that controls "instruments", whether they be test systems or burn-in racks, requires a user to first check the communication between the PC and the "instruments". The DTL-IFB can be connected to a standard PC RS232 COM port with an appropriate RS232/RS485 adapter (see Figure 4). Alternatively, a plug-in RS485 board can be plugged into the PC. COM1 and COM2 ports are typically brought out on the rear of a PC, and plug-in cards typically use COM3 or COM4.

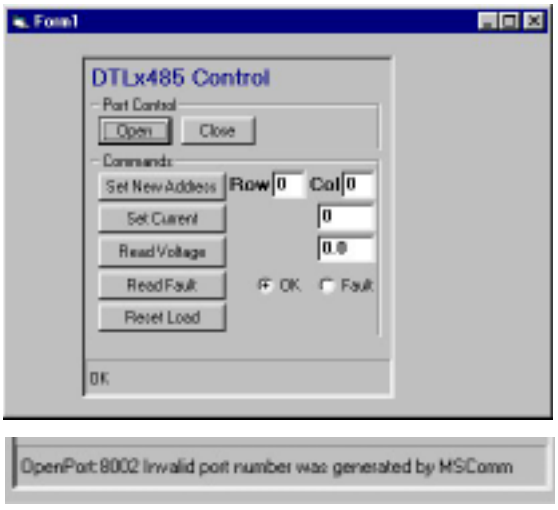


Figure 14. OK or Open Port Invalid Message

WinBurn™ DTLx485 Commands: Set Address

The Active X Control for a particular DTL-IFB-485 unit, will then have its Address set to match the Rotary Switches on the DTL-IFB-485. Physically, the rotary switches set the Hex Address for a particular DTL-IFB-485. Example: A123 is equal to 7B in Hex so rotary switch SW2 is set to "7" and SW3 is set to "B", while the DTLx485 Control is has its Row set to "7" and its Column set to "B". Refer to Figure15 for examples of dimensioning the Rows and Columns in code. Whereas each rotary switch has 16 settings, in the coding example of Figure 15, arrays have been setup for the rows and columns.

The desired load current of any DTL-IFB-485 and DTLXA Series of 100 Watt loads is then entered in the "Set Current" text box. Figure 15 coding shows how a low and high value setting can be used, should dynamic loading and/or power-cycle burn-in applications be desired.

Users must be careful to have provided appropriate heatsink/cooling to the DTLXA Series 100W electronic load to avoid catastrophic failure. The DTLXA Series is rated for only 10 Watts of power dissipation without external heatsink/cooling. Clicking on the "Set Current" button, will now update the load with the value entered in the text box.

The DTL-IFB-485 has an isolated on-board 12-bit A/D, typically used to read the voltage of a power source (DUT) under test or burn-in. Clicking on the "Read Voltage" button of the DTLx485 Control, will return the Voltage reading of the DUT. User's can also poll the DTL-IFB-485's Fault line. The Fault line goes active whenever the DTLXA Series falls below its compliance voltage (typically 2.5V, but 0.6 Volts for model DTL2A-LC). Users desiring to reset the load to no current, can click on the DTLx485 Control button "Reset Load".

A "timer" function can be used, should users desire to perform a dynamic burn-in, switching between high and low-levels of current. Review the General Declarations coding for a Form Load on the next page for examples of this usage.

The nature of the protocol of interfacing through an RS-485 port to communicate to the DTL-IFB-485 boards does not lend itself to fast updates (although all loads can be updated at the same time). Although users can control up to 256 DTL-IFB-485s from a single PC port, they may find that 128 units on a single RS-485 port, provides reasonable update times for burn-in applications. Also, multiple ports on a single PC approach can be used to extend the # of units beyond the 256 limit of a single COM port.

DATEL® INNOVATION and EXCELLENCE

ISO 9001

DS-04XX 7/03

DATEL, Inc. 11 Cabot Boulevard, Mansfield, MA 02048-1151
Tel: (508) 339-3000 (800) 233-2765 Fax: (508) 339-6356
Internet: www.datel.com Email: sales@datel.com
Data Sheet Fax Back: (508) 261-2857

DATEL (UK) LTD. Tadley, England Tel: (01256)-880444
DATEL S.A.R.L. Montigny Le Bretonneux, France Tel: 01-34-60-01-01
DATEL GmbH München, Germany Tel: 89-544334-0
DATEL KK Tokyo, Japan Tel: 3-3779-1031, Osaka Tel: 6-354-2025