DESTINY NETWORKING MANUAL

YieldPoint Inc.

May 2006

Warranty and Service Policy

Product Warranty

YieldPoint Inc warrants the instruments described in this manual to be free from defects in materials and factory workmanship to the original buyer. This warranty is contingent upon proper use of the equipment, and does not cover equipment that has been modified or has been subjected to abusive physical or electrical stresses. YieldPoint Inc., agrees to repair or replace, at its sole discretion, any instrument that fails to perform as specified within 6 months after date of the original shipment from the factory, or 3 months after the date of installation, whichever date comes first.

Yieldpoint Inc., reserves the right to make substitutions and modifications in the specifications of equipment that do not materially or adversely affect the performance of the equipment.

New equipment may be returned within 30 days of shipment with prior approval. New items which are less than thirty days old after shipment may be returned for credit, less a minimum restocking and testing charge of twenty percent of the list price upon factory approval only, provided the customer pays all shipping and handling charges. Specially ordered, or modified goods, or goods which have been used or have been unpacked, or goods which have been shipped more than thirty days prior are not returnable.

The information contained in this manual is subject to change without notice. YieldPoint Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Further, YieldPoint Inc., assumes no responsibility for any errors that may appear in this manual and makes no commitment to update, or keep current, the information contained in this manual.

Service Policy

Units within the warranty period returned for repair, test, and recalibration are serviced at no charge in accordance with the terms of the warranty policy. The Customer pays all transportation and other charges to the factory.

Units out of warranty returned for repair, test, and/or recalibration are handled on a time and material basis. If requested, or if costs exceed 50% of current list price, YieldPoint Inc., advises the customer prior to making the repairs. Such repairs are performed at the customer's expense. Typical test, recalibration, and repairs are 25% of the instrument's current list price. Transportation charges both ways are at the customer's expense.

Please be sure all returns are shipped with the following information included:

1. Your company Name with Billing and Shipping Addresses.
2. A complete description of your problem, or re-calibration data.
3. The contact person at your company, with their telephone and facsimile numbers.
4. Non-Warranty returns additionally need your Purchase Order Number.

Please pack your returned instruments in their original shipping cartons, or in equivalent strong protective shipping cartons.

Table of Contents

1. DESTINY Networking: Hardware

DESTINY is an RS485 networking system that enables data from all types of digital sensors to be transmitted over long distances and interfaced with existing mine-wide communications infrastructure such as: Ethernet running TCP/IP, leaky feeder wireless and other RS485 industrial networks running a varirty of serial protocols. It also provides solutions for local readout of the network by BlueTooth® enabled PDA type devices and data-loggers.

Each DESTINY network comprises a DESTINY Master located at one end of the bus and up to 15 DESTINY Slaves configured in a daisy chain configuration along a 2tp cable (1tp for 20-28Vdc,GND and 1tp for differential (A+B) signals).

1.1 What is RS485?

Recommended Standard 485 (RS-485) has become the industry's workhorse interface for multipoint, differential data transmission. RS-485 is unique in allowing multiple nodes to communicate bi-directionally over a single twisted pair. No other standard combines this capability with equivalent noise rejection, data rate, cable length, and general robustness. For these reasons, a variety of applications use RS-485 for data transmission. The list includes automotive radios, hard-disk drives, LANs, cellular base stations, industrial programmable logic controllers (PLCs), and even slot machines. The standard's widespread acceptance also results from its generic approach, which deals only with the interface's electrical parameters. RS-485 does not specify a connector, cable, or protocol.

In its simplest form, DESTINY is configured as an RS-485 bidirectional half-duplex bus comprising a transceiver (driver and receiver) located at each end of a twisted-pair cable. Data can flow in either direction but can flow only in one direction at a time. DESTINY allows for connection of up to 16 unit loads (ULs) to the bus. The 16 ULs commonly comprise 1 DESTINY Master and up to 15 DESTINY Slaves. *Figure 1* illustrates the configuration of a multi-point bus. In this application, three transceivers—one Master and two Slaves—connect to the twisted pair.

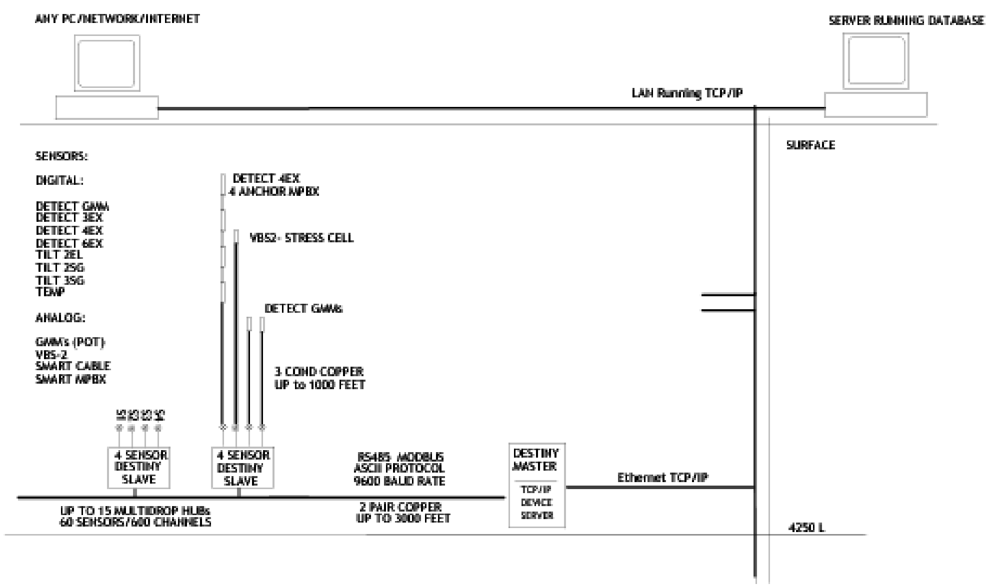


Figure 1: The conceptual topology of a DESTINY network. The Master forms a bridge between the mine communications system (TCP/IP in this case) and the RS485 DESTINY network. The network can accommodate up to 15 Slaves. The RS485 requires two twisted pairs (usually 18-24AWG) one to provide a power bus (+24Vdc and GND), the other for differential RS485 signals.

1.2 DESTINY Master

The DESTINY Master is both (i) an RS485 bus controller and (ii) a bridge to an existing mine-wide communication infrastructure. Different versions of the DESTINY Master are available depending on the type of communication interface required– RS232 (for leaky feeder radio-modems), TCP/IP for Ethernet, RS485 for existing industrial networks, or frequency modulation for ESG seismic systems.

Currently four types of DESTINY master are available:

DESTINY/RS232 provides an interface to an RS232 serial port such as a leaky-feeder radio modem, a PC or a serial to Ethernet device server (bought separately).

DESTINY/IP provides an interface to an existing Ethernet network running TCP/IP. The MASTER TCP/IP can operate in either TCP or UDP mode. TCP/IP connectivity is provided by a piece of Hardware called a Device Server (DS).

DESTINY/RS485 provides a bridge to an existing wide-wide RS485 network. The DESTINY network is optically isolated from the host network, and resides as a single unit load on the host (mine-wide) RS485 network. The MASTER RS485 can be powered from either the existing RS485 Network or a separate 20-28V supply.

DESTINY/ESG uses a Frequency Modulation technique to send information from an entire DESTINY network over a single channel of an ESG seismic system. The DESTINY network can be powered from the ESG seismic systems 28V supply. DESTINY/ESG sends data from each instrument on the DESTINY network at regular intervals.

All destiny Masters require a stabilized 20-28V power supply, and draw between 40mA and 180mA. The Master is provided with a medical grade 24Vdc power supply which should be plugged into a grounded 110VAC 60Hz outlet. The network cable shield should be interconnected at each Slave and then is connected to the earth at the 110VAC earth at the Master.

1.3 DESTINY Slave

Each DESTINY Slave provides a 4 PORTs that can accept any type of YieldPoint digital sensor and interface these to an RS485 network that can accommodate up to 15 Slaves (i.e. 60 sensors). Each Slave is powered from the (24Vdc) RS485 power bus, and draws a maximum of 40mA. All network wiring to a DESTINY Slave should be completed before any sensors are attached. Power is only supplied to the Slave when a sensor is plugged in. This is indicated by the LED on the unit which will begin to flash intermittently.

1.4. Network Design - Considerations

Since RS-485 allows connection of multiple transceivers, the bus (i.e. network) configuration is not as straightforward as in a point-to-point bus (RS-232C, for example) in which a single driver connects to one receiver alone. The optimal configuration for the RS-485 bus is the daisy-chain connection from Master (one end of the bus) to Slave 1 to Slave 2 to Slave 3 to Slave 15(max) at the other end of the bus. The bus must form a single continuous path, and the nodes in the middle of the bus must not be at the ends of long branches, spokes, or stubs. Figure 3a, Figure 3c, and Figure 3e illustrate three common but *improper* bus configurations. Figure 3b, 3d and 3f show equivalent *daisy-chained* configurations. Connecting a Slave to the cable creates a stub, and, therefore, every Slave has a stub. Minimizing the stub length minimizes transmission-line problems. A true daisy-chain connection, where for internal nodes both cables terminate at the terminal block, is the recommended approach to eliminate configuration problems.

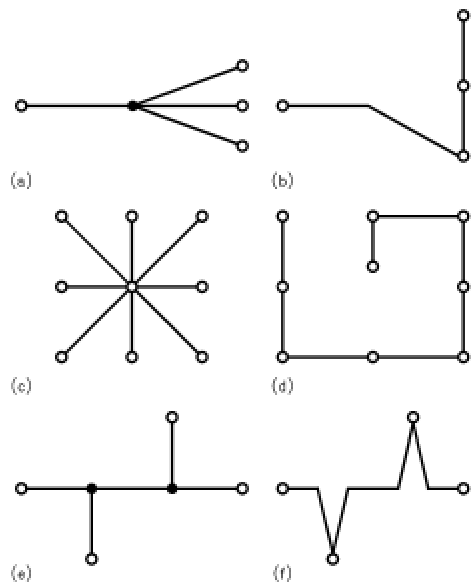


Figure 2: Proper (RHS) and improper (LHS) network designs.

1.5 Network Construction – Physical

Although the potential difference between the data-pair conductors (A and B) determines the signal without officially involving ground, the bus needs a ground wire to provide a return path for induced common-mode noise and currents, such as the receivers' input current. A typical mistake is to connect a Master and Slave(s) with only two data wires. The usual solution is to use a second twisted pair to provide power (Typ. 20-28V dc) and GND to the network. DESTINY components contain two 2-position terminal block to attach a cable with two twisted pairs. The wiring configuration is shown in Figure 3. the orange pair (A+B) is the differential signal pair, and the blue pair is 24Vdc (blue with white) and GND.

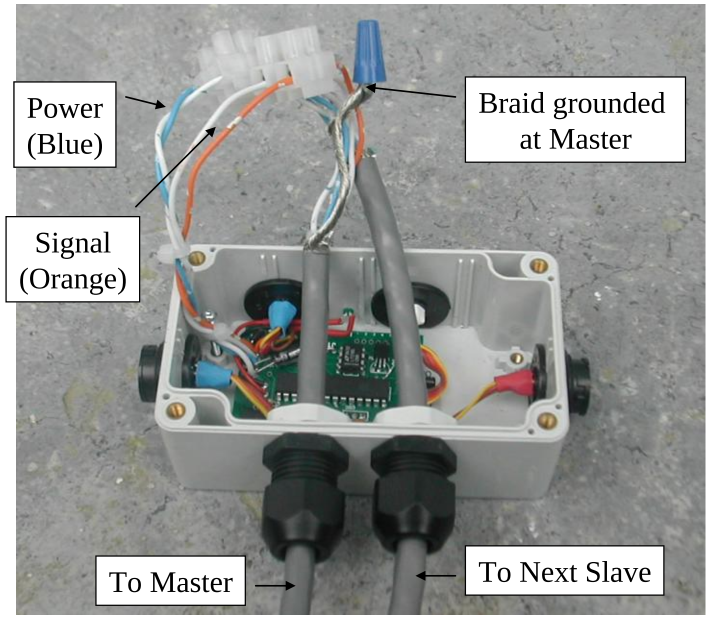
The DESTINY Master typically draws a max. current of 100mA and each Slave draws 16mA . To minimize power consumption each sensor on the bus is powered and reading taken sequentially, so avoiding overloading the network for large sensor arrays – Note: a single sensor draws 10-25mA depending on its type. Therefore a network with 1 Master and 15 Slaves would draw a little less than 350-365mA.

YieldPoint provides a high quality (medical grade) 24Vdc power supply for DESTINY networks that plugs into a 110V 60Hz outlet. The user may choose to power the network from any source that can provide 20Vdc to 28V dc at the required current rating which depends on the number of Slaves on the network, and will vary with network length.

1.5.1 Cabling

The standard EIA RS485 specifies does not specify the interconnection medium. However, use of twisted-pair cable is recommended. You can use a range of wire gauges, but designers most frequently use 18- 22 AWG. The characteristic impedance of the cable should be 100 -120 Ohm.

The recommended cable is Belden 9842 or an equivalent



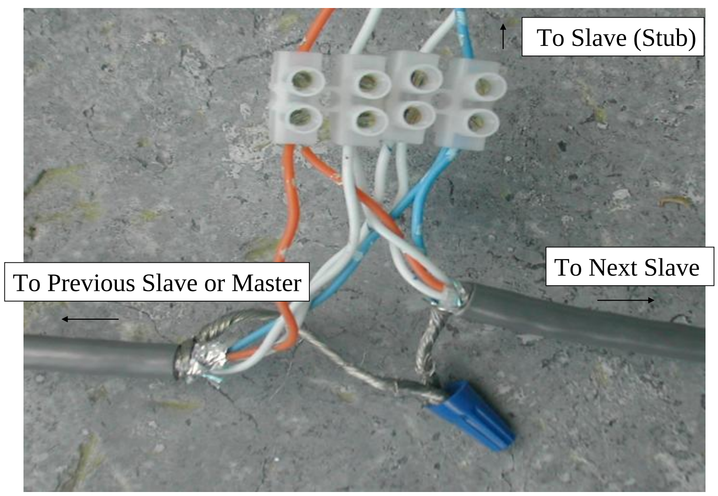


Figure 3 . Making the RS485 network connections at a DESTINY slave. The Blue pair is power (Blue +24Vdc and While GND), the orange pair is signal (Orange-A; white-B). The braid should be coupled together and is grounded at the master. The network cable shield should be interconnected at each Slave and then is connected to the earth at the 110VAC earth at the Master.

1.5.2 Data Rate Vs Cable Length

Data can be transmitted over an RS-485 bus for 4000 ft (1200m), and data can be sent over the bus at 10 Mbps. However, data cannot be sent at data at a rate of 10Mbps over a length of 4000ft: Rather there is a trade-off between data rate and network length: the longer the network, the slower that data rate required. *Figure 4a* shows a conservative curve of data rate versus cable length for RS-422 and RS-485. The key point is that you can't obtain the maximum data rate at the maximum cable length. But, if you operate the bus within the published, conservative curves, you can increase the likelihood of error-free operation.

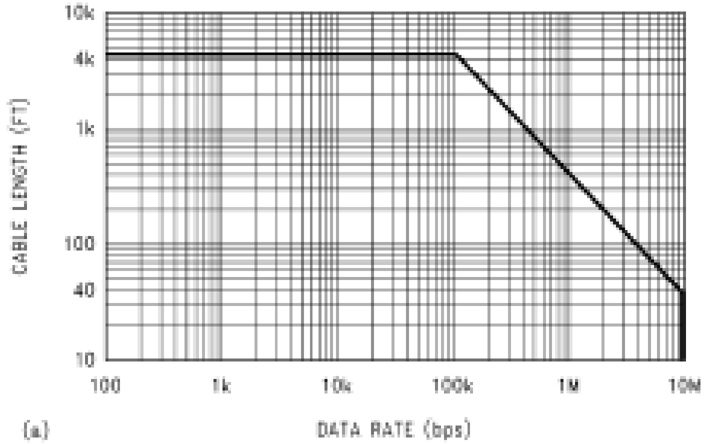


Figure 4: DESTINY sends at a rate of 9600bps and so cable lengths of 4000ft are feasible.

1.5.3 Termination: Important.

DESTINY RS-485 buses should be properly terminated. The purpose of the termination is to prevent adverse transmission-line phenomena, such as reflections. The DESTINY Master which must always reside at one end of the bus is internally terminated, and idle-state bias resistors are also included. The other end of the bus (far-end salve) should be terminated with a 1W 120Ohm resistor.

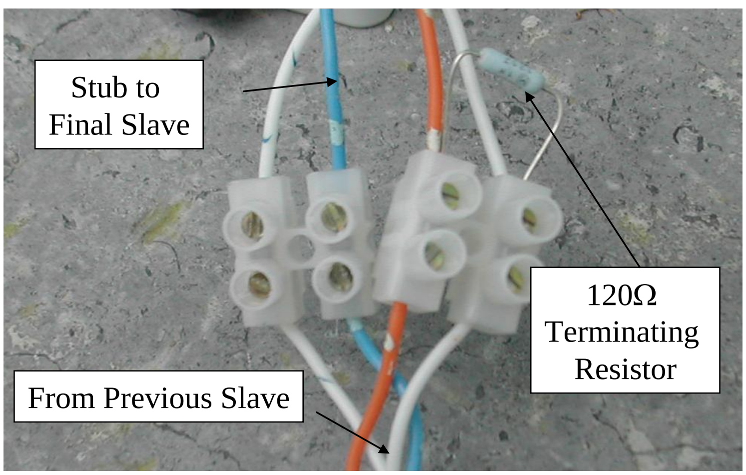


Figure 5. Termination with a 120Ω resistor.

1.5.4 ESD protection Transients

DESTINY provides RS485 transceivers that are rated for IEC-1000-4-4:+/-15kV Air discharge or +/- 4kV contact discharge. This level of ESD protection will guarantee immunity from field failures in all but the most severe ESD environments.

1.5.5 Short Circuit Fault Condition Protection

All YieldPoint sensors are equipped with a thermal shutdown "polyswitch" in the sensor connector which will prevent the DESTINY network from being interrupted due to a fault condition related to a sensor on the network. The network data lines themselves are also protected against a fault or short-circuit condition.

1.6 TCP Connectivity with a Device Server (DS)

Each Master/IP contains an internet enabled device server (Provided by Tibbo. Technology Inc. http://www.tibbo.com), that provides connectivity to an Ethernet network running TCP/IP using an RJ45 plug . In the current version of DESTINY/IP all Master **Configuration** commands are transmitted using UDP (User Datagram Protocol), whereas all network **Messaging** uses TCP (Transmission Control Protocol) connection to a specified Master Device. The application has the capability to run in *AutoPoll* mode and take automated readings from the entire network once every 30 minutes, by automatically creating a new TCP connection each time a different Master is polled.

Each data download cycle requires a **Global Poll** of each Master in sequence (note: the Masters are polled in the order they were added to the database). For each master this involves (i) issuing an instruction to the Master to reboot itself, (ii) creating a TCP connection between the PC (or Server) and Master using a Port which increments between 2000-2100 (iii) transferring data from all Slaves on the RS485 network.

In between data download cycles a TCP connection is established to display the Modbus ASCII messages (ModLog) being handled by the RS485 network controlled by the specified Master.

1.7 Addressing

3 levels of hierarchical addressing are required to define where a given sensor resides within the DESTINY network.

(i) Each DESTINY master has a pre-programmed MASTER_ID (01-32).

(ii) Each Master is in control of a DESTINY network which can accommodate up to 15 Slaves which each have a SLAVE_ID (01-15). **Important: Slaves on the same DESTINY network must have different SLAVE_IDs**. Failure to follow this rule may result in damage to components.

(iii) Each Slave can accommodate sensors in 4 different Ports which are addressed by the PORT_ID(01-04).

Therefore a sensor is addressed as follows: MASTER_ID=01, SLAVE_ID=03, PORT_ID=01 or 01-03-01 (Figure 5). In addition each sensor will have its own Sensor_ID(0-255), Sensor_Type (Code 21=GMM etc) which can be used to confirm that the sensor has been plugged into the correct Port.

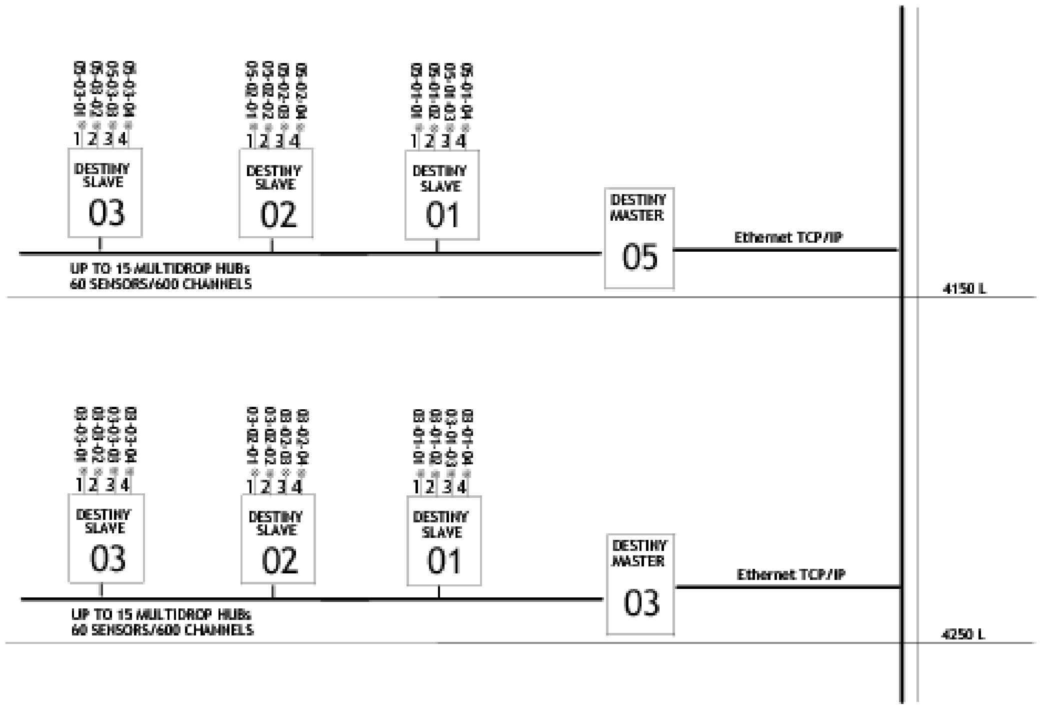


Figure 6: The Addressing hierarchy.

2. DESTINY/IP: PC/Server Software

2.1. Overview

DESTINY/IP is a PC/sever application to configure, monitor and manage the DESTINY network. The user interface has three "environments" each represented by a different Tab on the main form.

1. The Messaging Environment
2. The Configuration Environment
3. The Datalogging environment

2.2. Messaging Environment (The "Messaging" Tab)

The " Messaging " environment presents the user with the real-time messages being transmitted over either the Ethernet (TCP/IP) or the RS485 (Modbus) networks. A rudimentary ability to interpret these messages will enable to user to provide better maintenance and troubleshooting.

2.2.1. Device Server (DS) Messaging (Ethernet)

DS messages are sent over an Ethernet connection between the PC/server and the DESTINY Master. Commands exist to: (i) create a TCP connection, (ii) retrieve information regarding the network configuration, (iii) retrieve data from specified network slaves and (iv) synchronize the time and date for the Real Time Clock (RTC) in the Master with the PC/server (usually daily at 3:00am)

2.2.1.1 Port change, and TCP Connection

The message sequence shown in Figure 6, indicates that a successful TCP connection has been established with the current MasterID (01 in this case). During the sequence the green LEDs next to the Winsock state indicate progress. The final :OK command is a response from the DESTINY Master, indicating that an operational TCP connection has been established. At the end of the sequence the Winsock state should be "**Connected**". This sequence is followed every time the software needs to connect to a different Master, and the Port No. increments (between 2000 and 2100) each time a new connection in required. Note: A TCP connection must be established before commands (defined by function codes discussed in the next section) can be sent to the Master

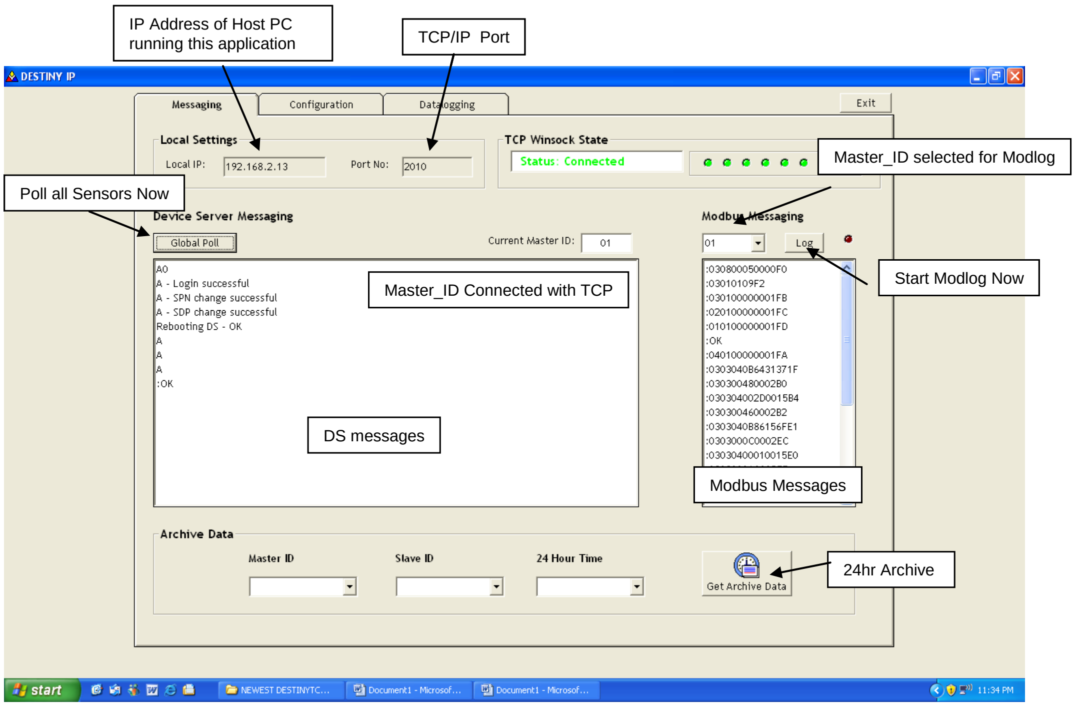


Figure 7. The Messaging Tab.

2.2.1.2 Device Server Messaging: Function Codes:

4 message functions exist:

M –Retrieve the SlaveMap
N – Retrieve Data from a specified Slave.
D – Set Date on the Master's clock
T - Set Time on the Masters clock.

"M" –Retrieve the SlaveMap indicating the population of Slaves

Example Query :

:M070099CRLF

The MasterID (chars3-4) is 07 and the 99 (chars 7-8) indicate that the current (i.e. latest) SlaveMap is retrieved.

24hr Archived Maps

Archived SlaveMaps from the Master are retrieved using the M command with 00-23 for chars 7-8, specifying the hour(00=midnight, 1800=6pm) for which the archived map should be returned. These archived maps take account of whether Slaves have been added or deleted from a network over a 24hr period.

Example Response:

:M070099CRLF   (echo original command)
:1100011100000000CRLF

The response consists of two parts (i) The message is echoed back (case for all DSmessages) and (ii) a 16 character string comprising 1's and 0's represents the SLAVE_ID's that the master has detected on it's network. In the example above Slaves with ID's 1,2,6,7,and 8 were detected by the Master on its network.

"N" – Obtain the Sensor reading from all 4 PORTs.

Example Query :

:N070399CRLF

The Master_ID(chars3-4) is 07, the Slave_ID(chars 5 and 6) is 03 and the and the 99 (chars 7-8) indicates that the latest set of readings is to be retrieved.

24hr Archived Data

Archived Data from the Master are retrieved using the M command with values of 00-23 for chars 7-8, specifying the hour for which the archived data should be returned.

Example Response:

:N070399CRLF   (echo original command)
:2006,5,12,8,0,23,7,3,1,1,21,2950,5374,0,0,0,0,0,0,0,0, CRLF
:2006,5,12,8,0,24,7,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0, CRLF
:2006,5,12,8,0,25,7,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0, CRLF
:2006,5,12,8,0,27,7,3,1,1,21,2950,12244,0,0,0,0,0,0,0,0, CRLF

Each line corresponds to a Port of the Slave (4 in all). In the example response, Slave 03 on Master 07 has two DETECT GMMs plugged into Ports 01 and 04.

The format of the data strings is as follows.

:Year, Month, day, hour,minute, second, Master_ID, Slave_ID,Port_ID, Sensor_ID, SensorType, Channel_1, channel_2,……Channel_10,CRLF

The value 2950 for the first channel of Port 1 is divided by 10 to obtain the temperature in K. The second channel (5374) is displacement and in mm according to

(Disp-100)/100 =(5374-100)/100 = 52.74mm

The transformations are necessary to avoid the overhead of carrying float-type variable around on the network, and are automatically applied by the DESTINY/IP software.

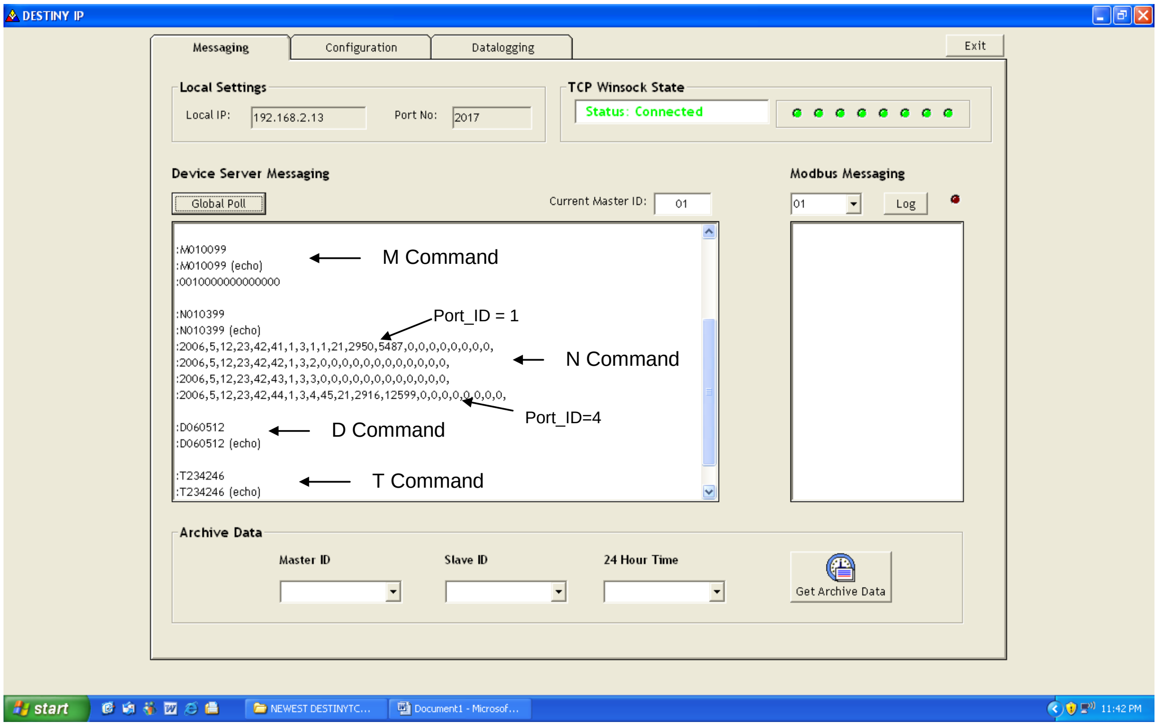


Figure 8 A typical sequence of DS messages. The M command is used first to determine the population of Slaves existing on the network. Then the N command is used to retrieve data from existing Slaves. Finally commands (D and T) are issued to synchronize the Master's clock.

"D" Command – Set Master Date

Example Query
:D060511CRLF

Set the Date of the master's clock to 11th May 2006.

Example Response:

:D060511CRLF (echo)

The response is based on the reading the date back from the Master's clock, and indicates whether the operation was successful.

"T" Command – Set Master Time

Example Query
:T140711CRLF

Set the Time of the master's clock to 2:07:11PM

Example Response:

:T140711CRLF (echo)

The response is based on the reading the time back from the Master's clock, and indicates whether the operation was successful.

2.2.2 Modbus Messaging (RS485).

This Section is not intended to provide a detailed introduction to Modbus protocol. That is provided by the reference *Modicon Modbus Protocol Reference Guide PI–MBUS–300 Rev. J* provided on the CD.

Modbus ASCII messages travel between Master and Slaves on each RS485 network.

2.2.2.1 Modbus ASCII protocol

On the DESTINY RS485 network all DESTINY components operate using a 9600,8,N,1 serial interface. Each 8–bit byte in a message is sent as two ASCII characters. The format for each byte in ASCII mode is:

Coding System: Hexadecimal, ASCII characters 0–9, A–F
One hexadecimal character contained in each ASCII character of the message
Bits per Byte: 1 start bit
8data bits, least significant bit sent first
no bit for parity
1 stop is used;
Error Check Field: Longitudinal Redundancy Check (LRC)

The protocol is Modbus ASCII. All messages start with a 'colon' (:) character (ASCII 3A hex), and end with a 'carriage return – line feed' (CRLF) pair (ASCII 0D hex and 0A hex). The allowable characters transmitted for all other fields are hexadecimal 0–9, A–F. Networked devices monitor the network bus continuously for the 'colon' character. When a colon is received, each slave decodes the address field (the next 2 chars) to find out if it the address corresponds. Intervals of up to one second can elapse between characters within the message. If a greater interval occurs, the Master assumes that no slave with that address exists on its network. .A typical message frame is shown below.

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	*n* CHARS	2 CHARS	2 CHARS CRLF

Figure 9: The framing of a Modbus ASCII Message

2.2.2.2 The Query–Response Cycle

The Query: The function code(next 2 chars) in the query tells the addressed DESTINY Slave device what kind of action to perform. The data bytes contain any additional information that the slave will need to perform the function. For example, a function code 03 will query the slave to read holding registers and respond with the contents of those registers which correspond to a sensor reading. The data field must contain the information telling the Slave which data to read. The error check field provides a method for the Slave to validate the integrity of the message contents.

The Response: If the Slave makes a normal response, the function code in the response is an echo of the function code in the query. The data bytes contain the data collected by the Slave. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error. The error check field (LRC check) allows the master to confirm that the message contents are valid.

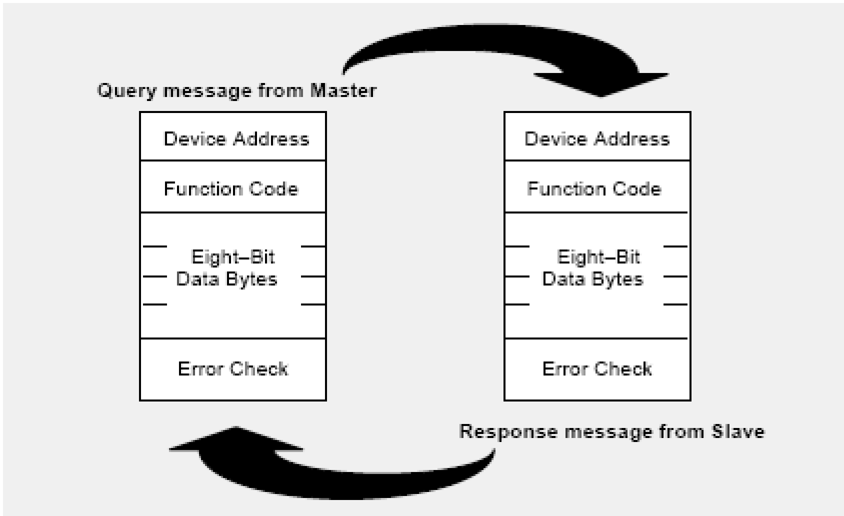


Figure 10: The Query=Response cycle for Modbus.

2.2.2.3 Modbus Device Address codes

Slaves are addressed by SlaveIDs **01-0F** (1-15) corresponding to the addresses for each of the DESTINY Slaves on the RS485 network

2.2.2.4 Modbus Function Codes

"01" (Read Coil Status) Returns the PortMap(Slave memory location 0000) corresponding to which Ports have sensors plugged in.:

Example:

Query
: 0101000000001CRLF
(Read the PortMap for Slave01)

Response
: 01010102 CRLF
(Slave01 has a sensor plugged into Port2 (0002H=0010 in binary). Other codes 0003H (=0011b) indicates sensors in Ports 1 and 2.etc.

"03" Read a specified number of bytes from a specified memory location of a specified Slave.

Example:

Query:
: 010300320002C2 CRLF
(Starting at memory location 50 (Hex 0032) read two(**0002**) bytes. i.e. Bytes at locations 50 and 51)

Response:
: 010302000100155DCRLF Contents of Slave memory locations (or holding registers) 50 and 51

“08” Turn on Sensors and Read Command.

Example:

Query
: 0108000050000F2 CRLF

Reponse
: 0108000050000F2 CRLF

Response is simply an echo of the query if the sensor is read correctly. This process may take several minutes and the Red LED above the Modbus Messaging Text-box will flash every 10s to indicate the application isrunning..

2.2.2.5 The Function sequence

A series of Query-response cycles are used by a Master to obtain data from all the networked Slaves. The sequence consists of two loops:

(i) **LOOP 1** The Master sends a **01** command to each Slave (1-15) and if it does not receive a reply within 1s then it assumes that no Slave with that SlaveID exists on the network, so it proceeds to the next SlaveID. If a response is obtained, then a **08** command (turn on and read all sensors) is issued to the Slave that triggers a reading from each of its 4 Ports.

(ii) **LOOP 2** The Master sends a **01** command to each Slave and if it does not receive a reply within 1s then it assumes that no Slave with that SlaveID exists on the network, so it proceeds to the next SlaveID. If a response is obtained, then a **03** command is issued to the Slave which sends data from holding registers corresponding to the reading taken in LOOP 1.

Since data from the DESTINY Master is requested at specified intervals this data is stored in on-board memory.

This sequence repeats continually until, at 30min intervals (on the hr and ½ hr) the DESTINY/IP application running on the PC/server sends a request for the data to be downloaded from the Master (see DS messaging above).

2.3 The Configuration Environment (the “Configuration” Tab”)

YieldPoint uses a 3rd party (Tibbo Technology Inc.) IP device server in each DESTINY Master IP. Full information and specificationss on this device can be found at the URLs :
http://www.tibbo.com/ds202.php
http://www.tibbo.com/tdst_dsman.php

The configuration environment provides the functionality to (i) Add/Remove Masters (ii) Configure a device server within a Master (iii) connect to a specific Master and send commands from the Command-line.

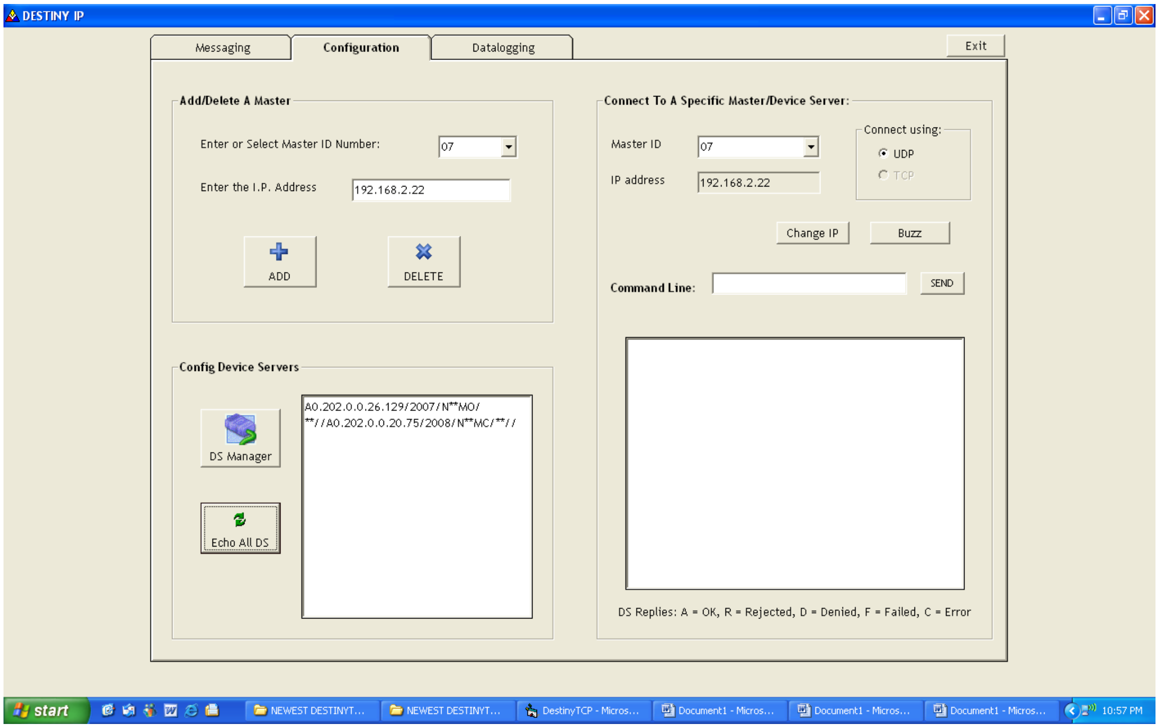


Figure 11: The Configuration Environment

2.3.1 .Add/Delete a Master

Before DESTINY/IP can communicate with a Master the MasterID and IP address of its Device Server must be added as shown in Figure 11.

2.3.2 Configuring a Master's Device Server: DS Manager

DS Manager is used to configure the DS. The DS Manager will autodiscover all the Device Servers connected to the network as it is opened (Figure 11).

The additional Tabs on the settings form (Figures 14 and 15) control the Serial Settings for communications between the Master and Device server, and how the data sent using TCP is packetized. The user should not need to change these settings.

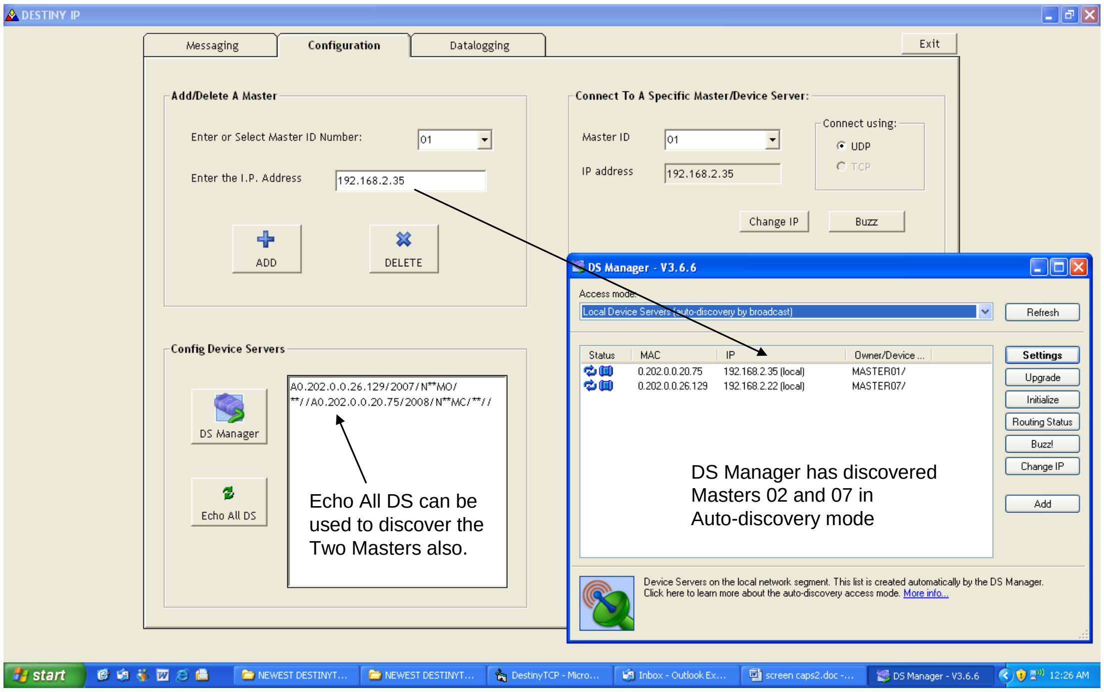


Figure 12: DS manager has auto-discovered 2 Device servers (i.e. DESTINY masters) connected to the Ethernet network. The MAC address are provided from the factory and are indicated on the Master.

By Selecting a Device Server and clicking on the Settings button the form in Figure 13 allows the setting to be changed (Figure 12).

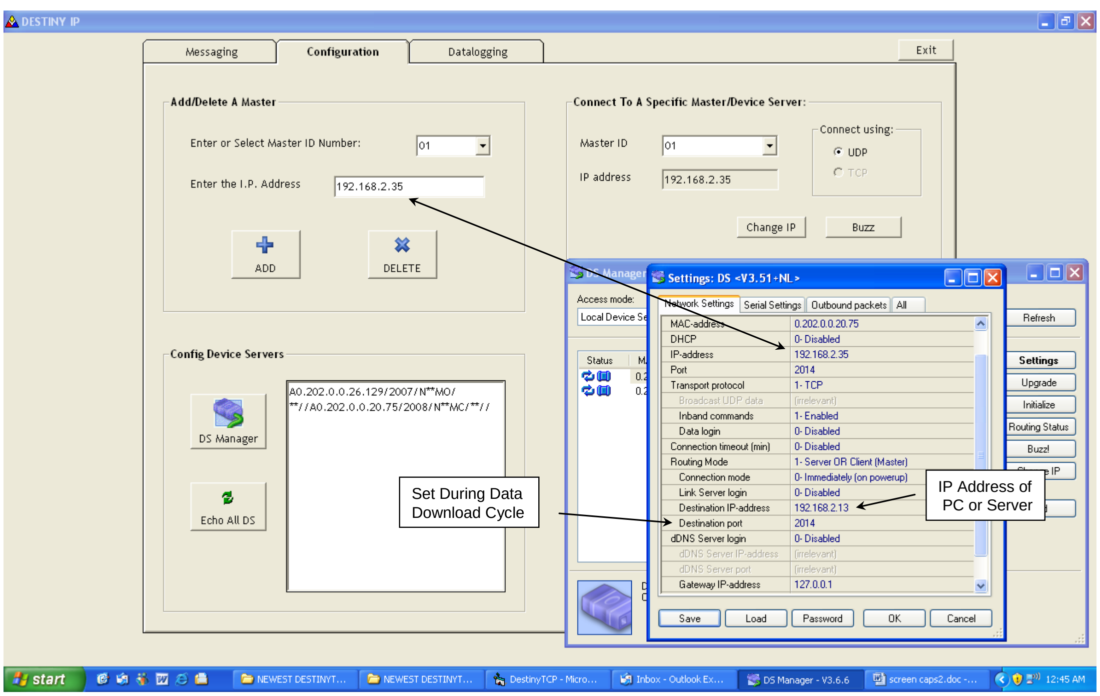


Figure 13: The Settings form.

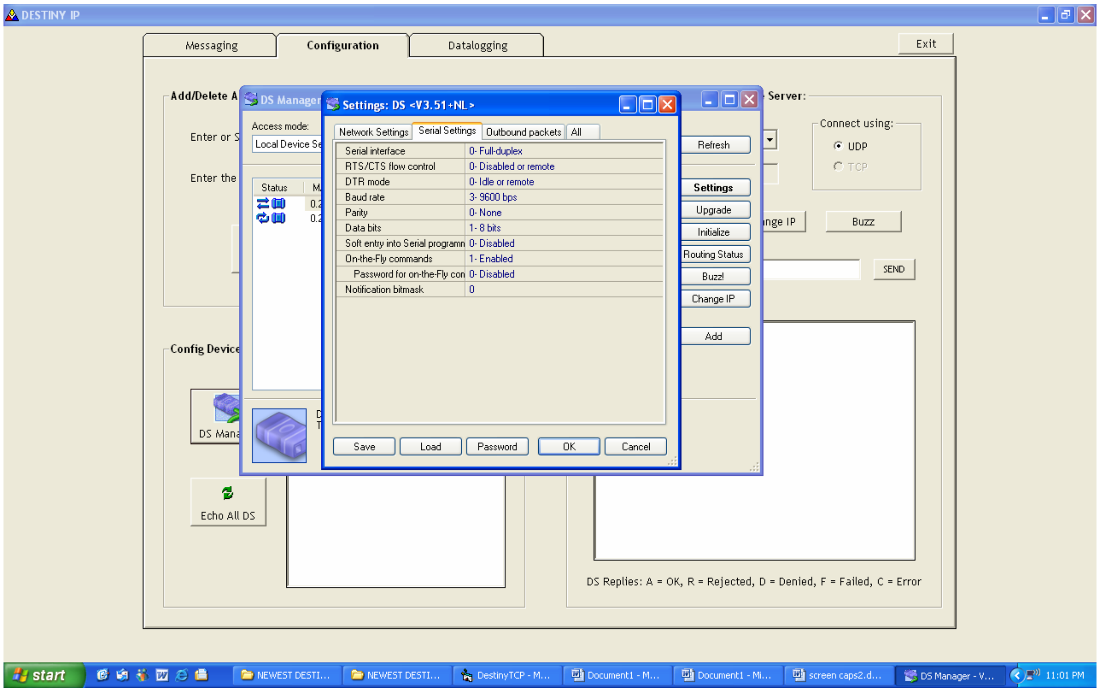


Figure 14: Serial Settings for communications between the device server and the Master. These setting should not be changed.

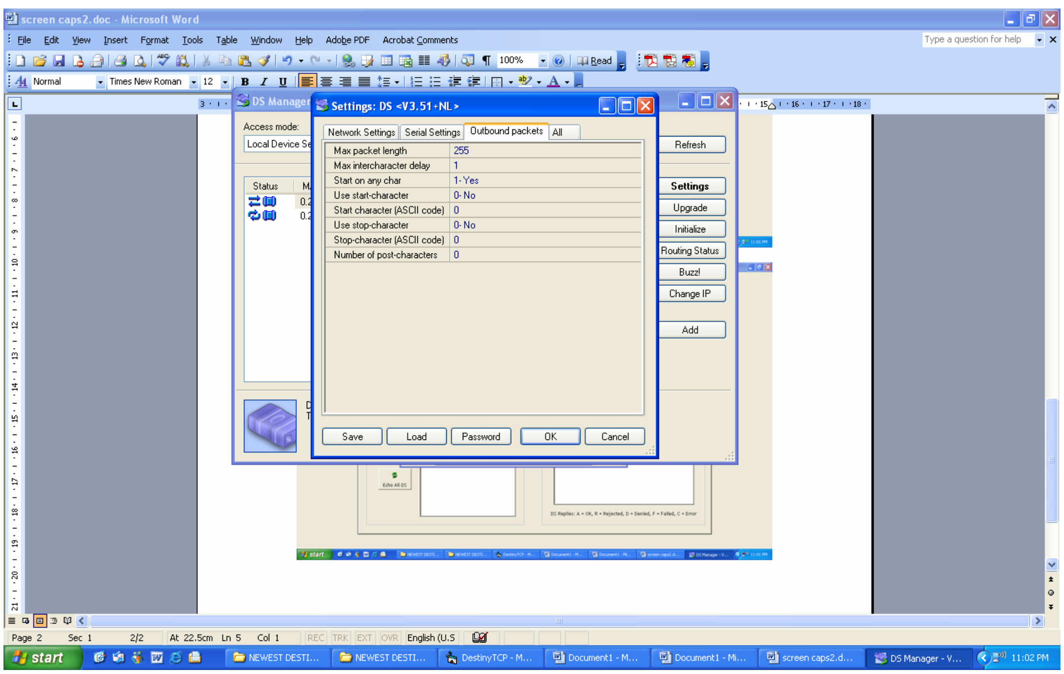


Figure 15: Packetization Settings. These should not be changed.

2.3.3 Connect to a specific Master/Device Manager

This functionality allows UDP commands to be sent directly to the DS in the selected Master. For example, clicking on "Buzz" (see Figure 11) will cause the DS status LEDs to flash rapidly 3 times. A wide range of other commands can be enetered at the command line, and full deails are provided on the DS manual included with the CD.

2.4 The Datalogging environment (The "Datalogging" Tab)

Data from the DESTINY network is automatically written to a file when readings are retrieved every 30 minutes. The naming of these files is as follows

Slave_ID Sensor_ID

yp – 01 – 03 – 01 – 45 - 21.txt

Master_ID Port_ID Sensor_Type

If the file does not exist then DESTINY-IP will create a new file.

The Datalogging Tab (Figure 16) shows a table of all the files existing and the number of records in each file. Records from the table and the associated text file (Figure 17) can be deleted at any time

2.4.1 Troubleshooting: The Error Logfile

The Logfile is a text document that can be opened from the data-logging environment. The Logfile is updated every time a new connection is made to a Master. In addition, all instances where the DESTINY-IP encounters an error due to an unexpected response from the Master are recorded in the LogFile. These errors may be caused by (i) a non-responsive or damaged Master (power loss to master), (ii) TCP/IP transmission failures. As a form of preventative maintenance the LogFile should be viewed on a regular basis to ensure that all component of the system are performing reliably. The file can be deleted at any time.

2.5. TestMode

If a master has no slaves on its network then it will (i) return a Slave_Map of 0000000000000000 in response to any M command and (ii) return a test data sequence as shown below.

:N070399CRLF (echo original command)
:2006,5,12,8,0,23,7,1,1,1,21,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000, CRLF
:2006,5,12,8,0,23,7,1,2,1,21,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000, CRLF
:2006,5,12,8,0,23,7,1,3,1,21,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000, CRLF
:2006,5,12,8,0,23,7,1,4,1,21,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000, CRLF

This is so-called TestMode output and can be used to determine whether the Ethernet network is operating correctly and without errors. To do this The Master should simply be attached using the RJ45 plug. Note: It may be necessary to complete a full Modbus cycle (LOOPs 1 and 2 above) after unplugging the Master from its network, for the Master to enter TestMode.

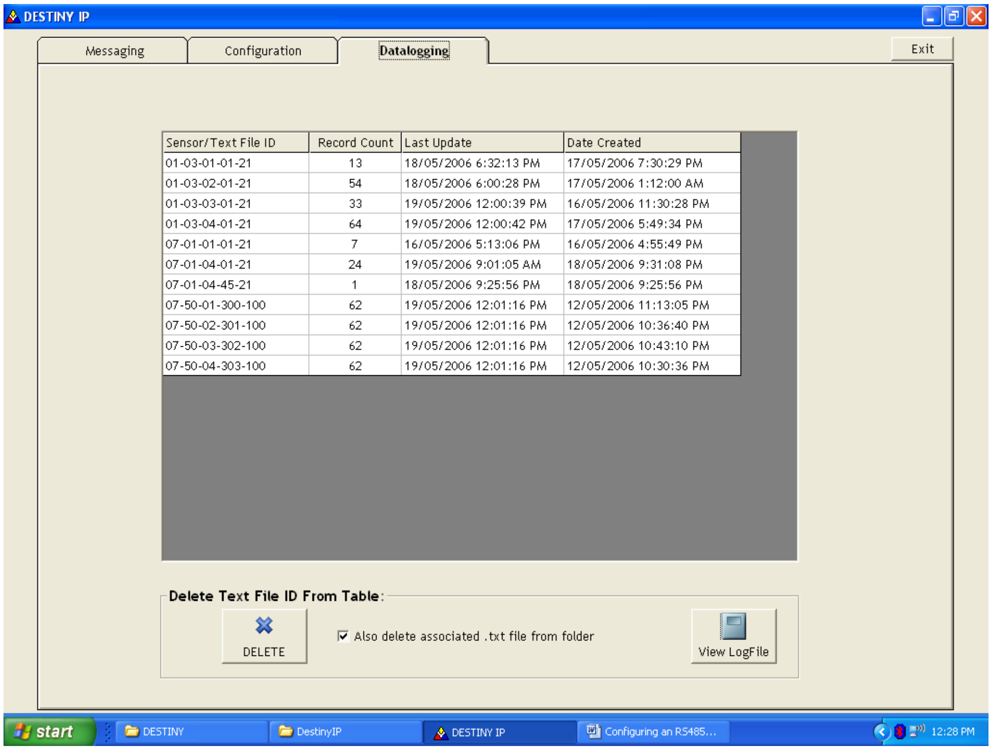


Figure 16: The Datalogging environment lists the files created and the number of records in each. The final four files are created in TestMode.

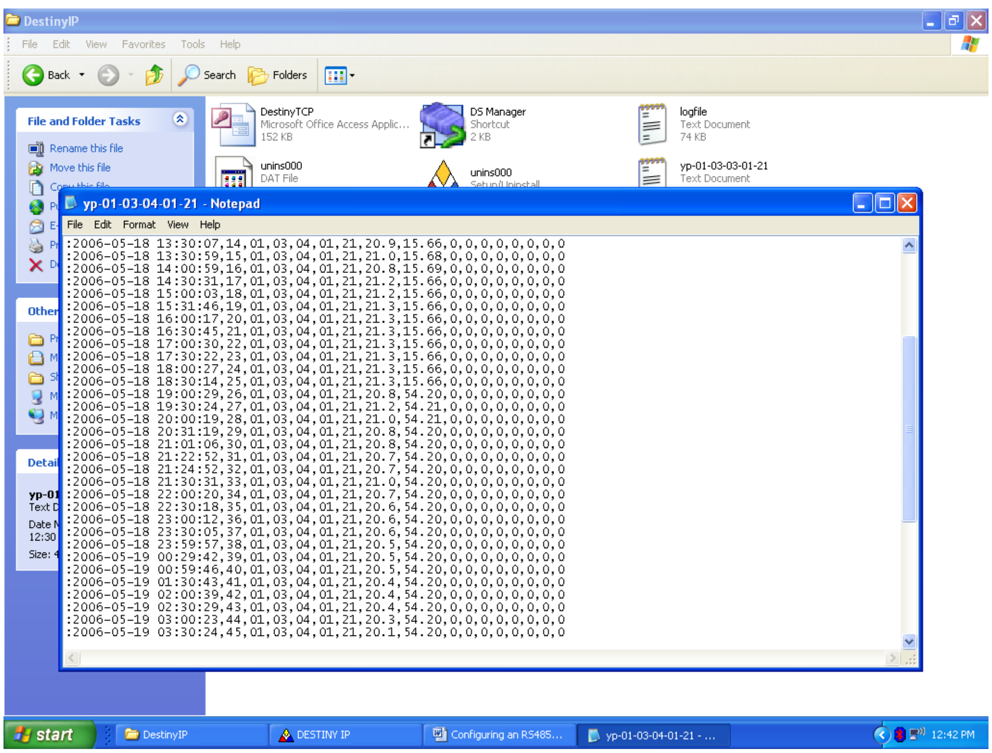


Figure 17: Output for file yp-01-03-04-01-21 (MasterID 01, SlaveID 03, PortID 04, SensorID 01, SensorType 21 (DETECT GMM). The output shows a steady temperature (channel 1) between 20 and 21.5C and a displacement that jumps from 16.66 to 54.20 between 6:30pm and 7:00pm on May 18th 2006

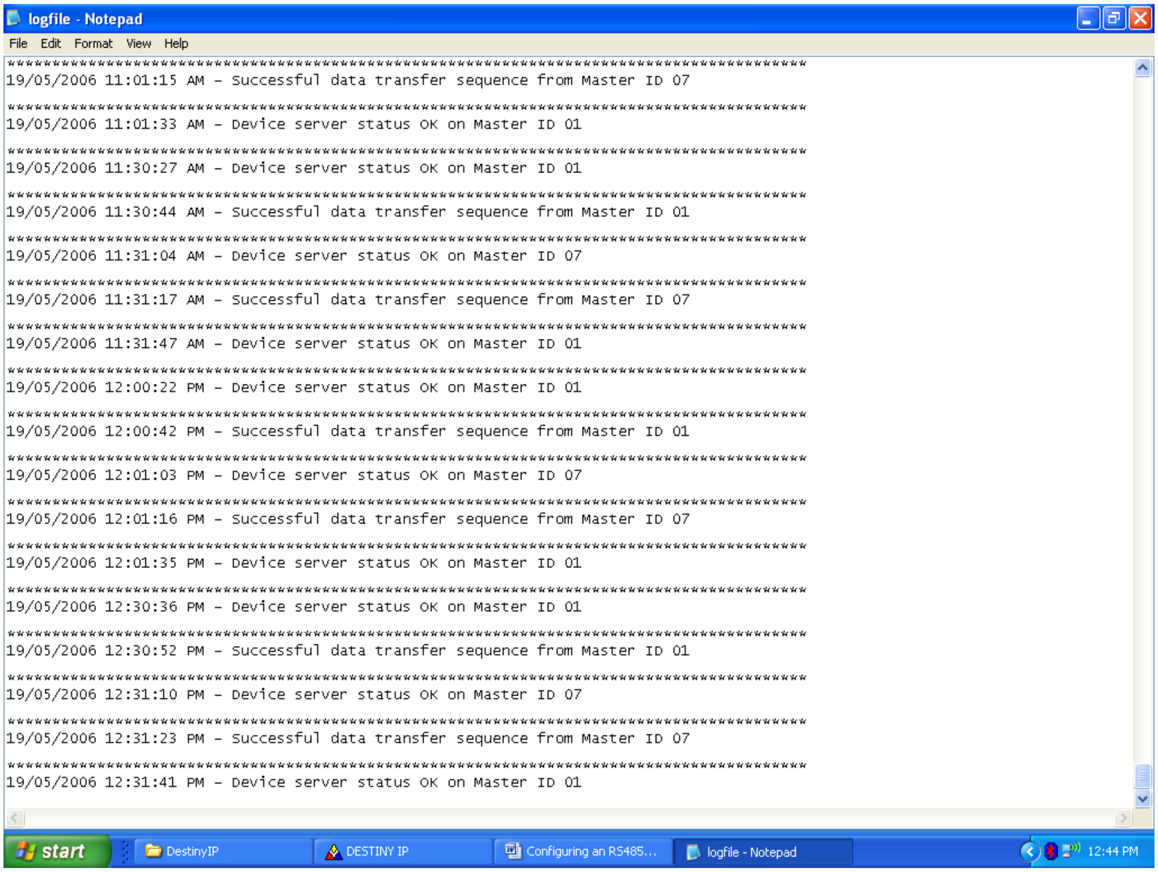


Figure 18. An example Logfile.

3. Getting Started–An *in-office* network.

It is strongly advised that a new user spend 20 minutes and build a small “in-office” network with a Master and a Slave, in order to understand how the software and hardware work in a controlled environment. This will prove far easier to troubleshoot than in the underground environment. Call YieldPoint at 613-531-4722 if you need help getting started with an office network.

To run DESTINY/IP the user requires a PC or server of the following specification

- Intel Pentium processor
- Microsoft Windows XP Professional or Home Edition (Service Pack 1 or 2),
- 128MB of RAM (256MB recommended)
- Up to 10MB of available hard-disk space

The installer provided on the CD will create two folders /Program Files/DESTINY/IP and /Program Files/Tibbo. The latter contain the files needed to run the Device Server (DS) manager. The installer will also create a shortcut to DESTINY/IP on the desktop.

3.1 Connecting to a single Master.

STEP 1

Simply plug a network cable into the RJ45 plug, and connect the master to its power supply unit (PSU).

3.2. Configuring the Master’s IP address.

STEP 2

Following the instructions outlined in Section 2.3.2 (Configuring a Master’s Device Server) input the PC/Server’s IP address,Gateway IP and Subnet Mask.

Make sure all other settings correspond.

STEP 3

On the Configuration Tab add the Master (MasterID=01 in this case) with the correct IP address of the **Desination** PC/server on which the application is running. This information is presented on the messaging tab.

3.3. Running in TestMode with a single Master attached.

STEP 4:

Return to the messaging environment and click on the “Global Poll” button. A connection will be established with Master 01 and the Device Server messaging will output the test TestMode data shown in Figure 20. If this process in unsuccessful (i) check all network connections, and (ii) check all IP setting in the DS manager. If the problem persists it may be necessary to disable the windows firewall in the **Control Panel**. When this sequence is complete the Master will automatically begin its Modbus messaging cycle, and since there are no Slaves attached to the network, it will simply send a **01** function code to Slaves 1-15, but will not receive any replies. Thereafter every 30mins the Master will interrupt its Modbus cycle and transmit the TestMode datastream.

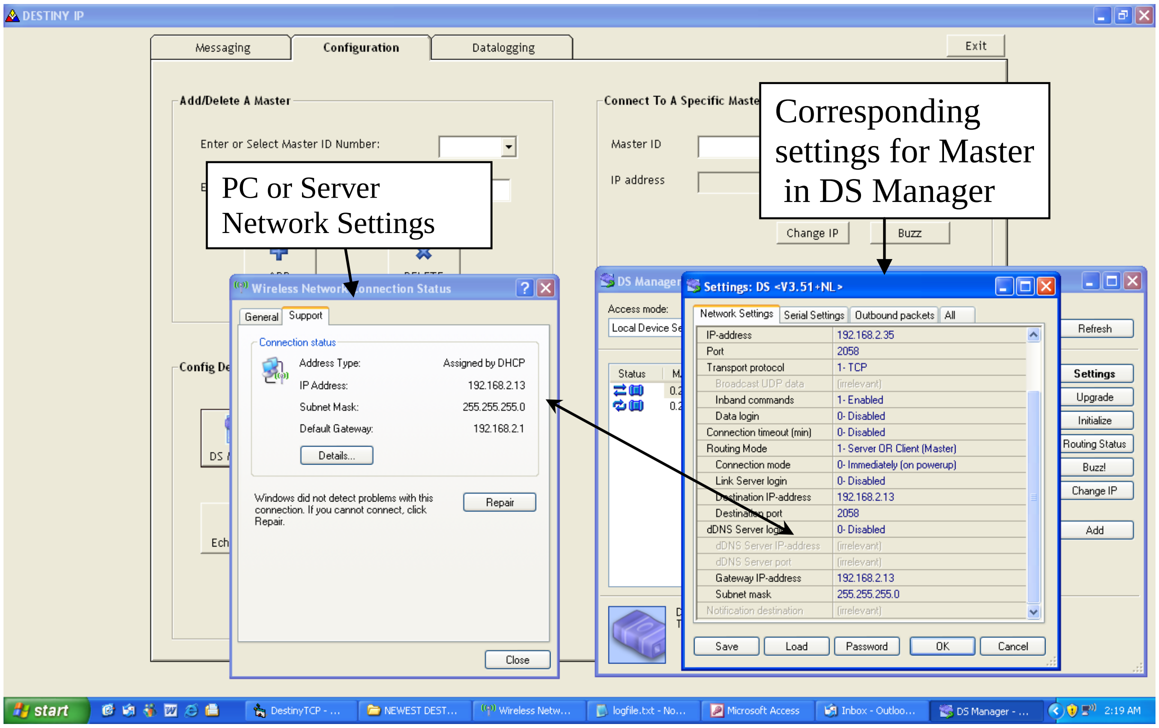


Figure 19: Using DS Manager to set the IP settings. The box on the left is opened from the networking icon as the LHS of the taskbar.

3.4. Building a simple network

If already plugged in, unplug the power Supply unit from the 110VAC power supply.

STEP 5

Connect a Slave (in this example SlaveID is 01) to the Master using a short length of network cable (see Section 1.5). After the network connection has been made, plug a sensor (DETECT GMM is this case) into Port 01 of the Slave. Plug into the 110V power supply.The Slave's LED will flash when contact is made indicating it is operational and intermittently when it receives Modbus Commands from the Master. Note: The Modbus Slave is only powered when a sensor is plugged in.

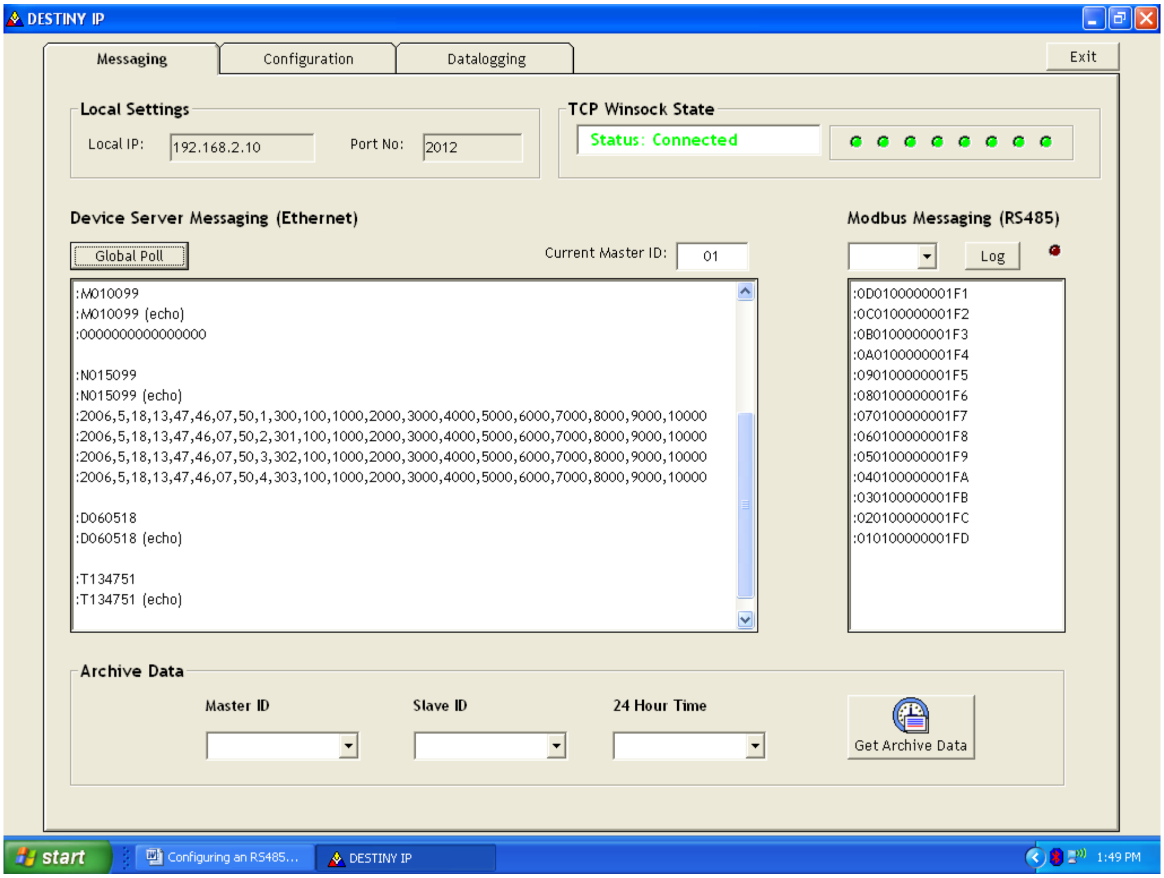


Figure 20 The TestMode data for Master 01 connected without any sensors attached.

STEP 6.

Return to the **Messaging Tab**. Select 01 from the pull-down list under Modbus Messaging and click the Log button. The DS messaging window will indicate progress as a TCP connection is established to Master 01. When the connection is complete, and after a 10s delay the Modbus messaging window will start to display Modbus messages being sent and received by the Master over the RS485 bus. Wait for a complete Modbus cycle (1-2mins) which is indicated by an :OK appearing in the Modbus Messaging textbox. Click **Global Poll**. This will retrieve the data read from the sensor plugged into Port 01 of the Slave (Figure 21). This data for Port 01 will be appended to the File yp-01-03-01-01-21.txt which will have been created (if it didn't exist) in the /Program Files/ DESTINY/IP folder (Figure 22). The other 3 Ports return zeros since no sensors areattached, and for these ports nothing will be written to file.

The corresponding logfile is shown in Figure 23. "*" symbols indicate successful operations and "#" symbols unsuccessful or failed operations.

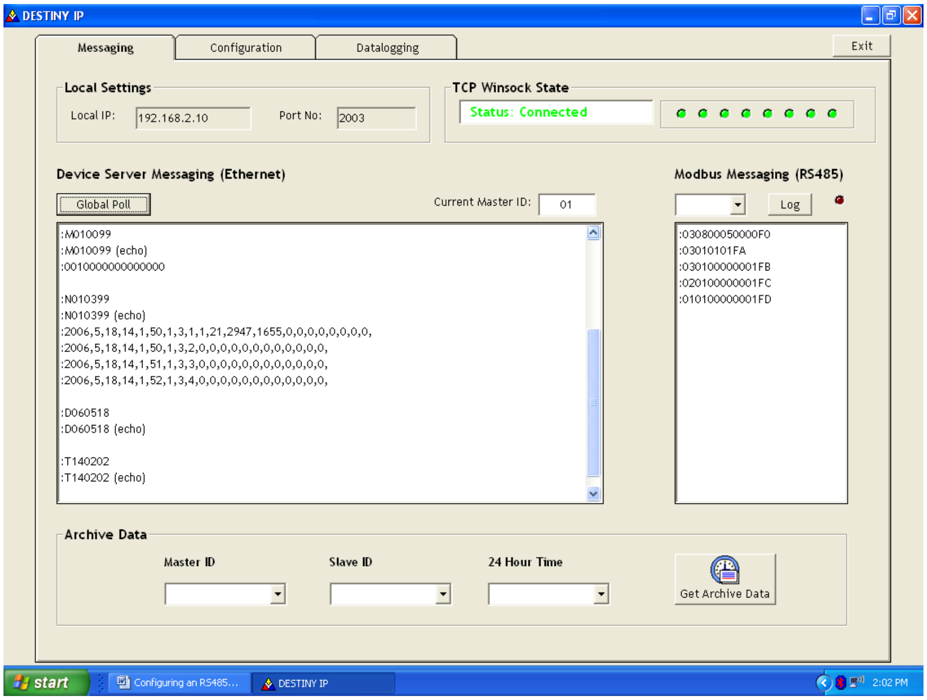


Figure 21 The output screen following a Global Poll with a sensor (SensorID=01 and SensorType=21 GMM) plugged into Port 01 of Slave 03 on Master 01. The Reading is for 18^{th} May 2006 at 2:01:50. Channel 1 indicates a temperature of 21.3C ([2947-2734]/10) and channel2, 1655, a displacement of 15.55mm ([1655-100]/100). These conversions are automatically completed by DESTINY/IP before writing to file (Figure 20).

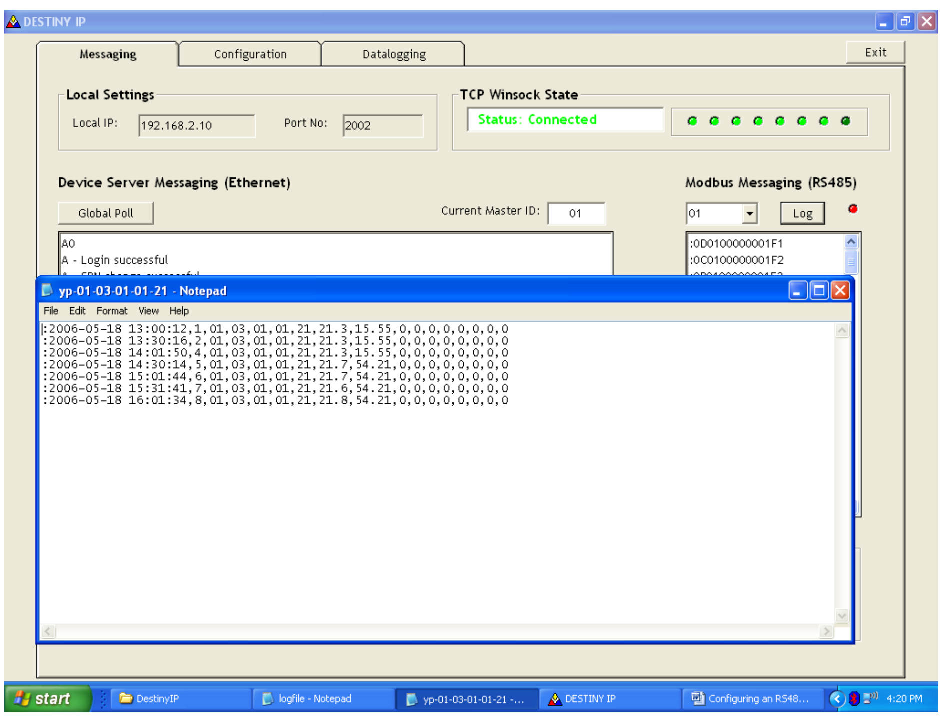


Figure 22: The data written to the file over a 3 hour period. Channel 1 indicates temperature (^{o}C) of 21.3-21.8 and Channel 2 (displacement in mm) indicates a significant jump in movement from 14.55mm to 53.21mm between 2:00pm and 2:30pm.

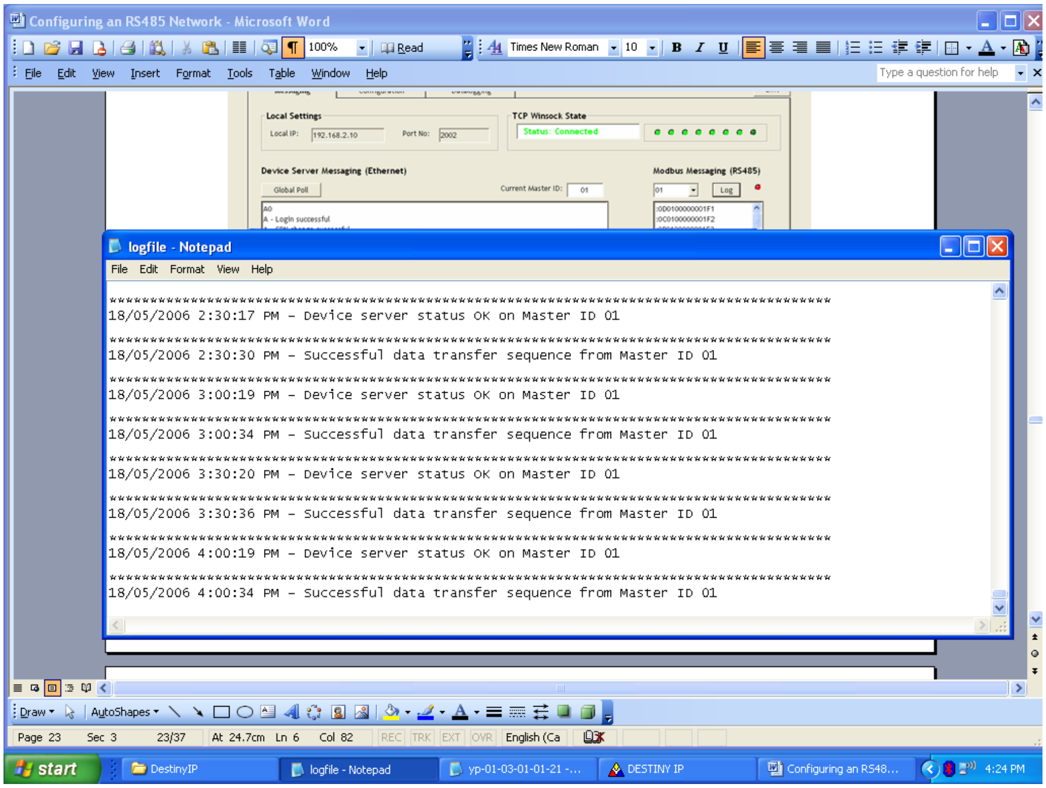


Figure 23: The Logfile between 2:30pm and 4:00pm.

3.5. Help setting up an *in-office* network.

YieldPoint will provide telephone assistance and Step-by-Step instruction for customers who want to set up an office network: Call 613-531-4722.

4. 0 Esssential Reading.

Goldie J. Ten Ways to Bulletproof RS-485 Interfaces Copyright 1996 Reed Elsevier Inc. National Semiconductor Application Note 1057 October 1996. *(Included on CD)*

4.1 Useful References

1. Murdock, G and J Goldie, "AN-702: Build a direction-sensing bidirectional repeater," *Interface Databook*, National Semiconductor Corp, 1996.
2. True, K, "AN-808: Long transmission lines and data-signal quality." *Interface Databook*, National Semiconductor Corp,1996.
3. Vo, J. "AN-903: A comparison of differential-termination techniques," *Interface Databook*, National Semiconductor
Corp, 1996.
4. Goldie, J, "AN-847: Fail-safe biasing of differential buses," *Interface Databook*, National Semiconductor Corp, 1996.
5. ANSI/TIA/EIA-422-B-1995, *Electrical characteristics of balanced-voltage digital-interface circuits*.
7. Sivasothy, S, "AN-409: Transceivers and repeaters meeting the EIA RS-485 interface standard," *Interface Databook*,

National Semiconductor Corp, 1996.

Appendix 1:

http://pdfserv.maxim-ic.com/en/an/AN763.pdf

Maxim Semicononductor Application Note 763: Jul 12, 2001

Guidelines for Proper Wiring of an RS-485 (TIA/EIA-485-A) Network

The proper method of wiring an RS-485 network is described, with recommendations for twisted-pair cabling and correct location of termination resistors. Received waveforms are shown for examples of proper and improper cable termination. Network configurations are shown for simple single-transmitter/multiple receiver through multiple transceiver to multi-branched circuits.

This application note is intended to provide basic guidelines for wiring an RS-485 network. The RS-485 specification (officially called TIA/EIA-485-A) does not specifically spell out how an RS-485 network should be wired. But it does give some guidelines. These guidelines and sound engineering practices are the basis of this note. The suggestions here, however, are by no means inclusive of all the different ways a network can be designed.

RS-485 transmits digital information between multiple locations. Data rates can be up to, and sometimes greater than, 10Mbps. RS-485 is designed to transmit this information over significant lengths, and 1000 meters are well within its capability. The distance and the data rate with which RS-485 can be successfully used depend a great deal on the wiring of the system.

Wire

RS-485 is designed to be a balanced system. Simply put, this means there are 2 wires, other than ground, that are used to transmit the signal.

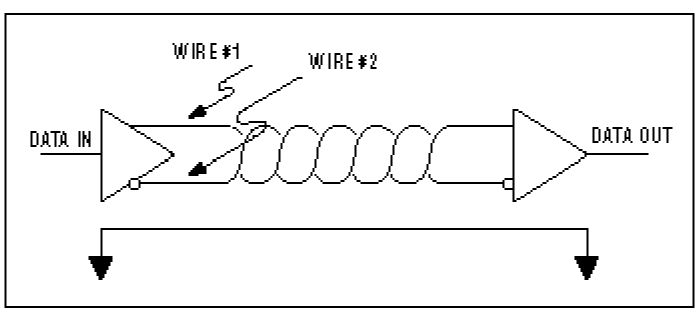


Figure 1. A balanced system uses 2 wires, other than ground, to transmit data.

The system is called balanced, because the signal on one wire is ideally the exact opposite of the signal on the second wire. In other words, if one wire is transmitting a high, the other wire will be transmitting a low, and vice versa. See Figure 2.

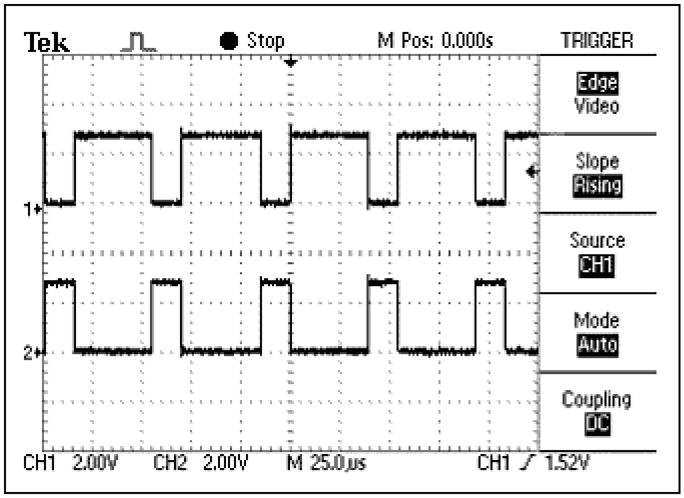


Figure 2. The signals on the 2 wires of a balanced system are ideally opposite.

Although RS-485 can be successfully transmitted using multiple types of media, it should be used with wiring commonly called "twisted pair."

What Is Twisted Pair, and Why Is It Used?

As its name implies, a twisted pair is simply a pair of wires that are of equal length and are twisted together. Using an RS-485-compliant transmitter with twisted-pair wire reduces two major sources of problems for designers of high-speed long-distance networks: radiated EMI and received EMI.

Radiated EMI

As shown in Figure 3, high-frequency components are present whenever fast edges are used in transmitting information. These fast edges are necessary at the higher data rates that RS-485 is capable of transmitting.

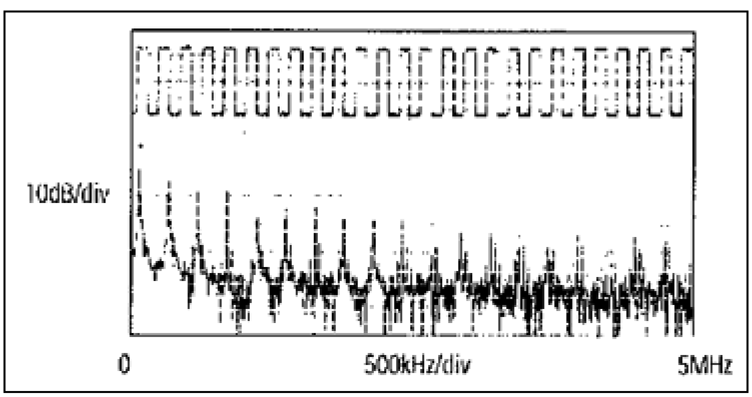


Figure 3. Waveform of a 125kHz square wave and its FFT plot

The resultant high-frequency components of these fast edges coupled with long wires can have the effect of radiating EMI. A balanced system used with twisted-pair wire reduces this effect by trying to make the system an inefficient radiator. It works on a very simple principle. As the signals on the wires are equal but opposite, the radiated signals from each wire will also tend to be equal but opposite. This has the effect of canceling each other out, meaning no net radiated EMI. However, this is based on the assumption that the wires are exactly the same length and in exactly the same location. Because it is impossible to have two wires in the same location at the same time, the wires should be as close to each other as possible. Twisting the wires helps counteract any remaining EMI due to the finite distance between the two wires.

Received EMI

Received EMI is basically the same problem as radiated EMI but in reverse. The wiring used in an RS-485 system will also act as an antenna that receives unwanted signals. These unwanted signals could distort the desired signals, which, if bad enough, can cause data errors. For the same reason that twisted-pair wire helps prevent radiated EMI, it will also help reduce the effects of received EMI. Because the two wires are close together and twisted, the noise received on one wire will tend to be the same as that received on the second wire. This type of noise is referred to as "common-mode noise." As RS-485 receivers are designed to look for signals that are the opposite of each other, they can easily reject noise that is common to both.

Characteristic Impedance of Twisted-Pair Wire

Depending on the geometry of the cable and the materials used in the insulation, twisted-pair wire will have a "characteristic impedance" associated with it that is usually specified by its manufacturer. The RS-485 specification recommends, but does not specifically dictate, that this characteristic impedance be 120 ohms. Recommending this impedance is necessary to calculate worst-case loading and common-mode voltage ranges given in the RS-485 specification. The specification probably does not dictate this impedance in the interest of flexibility. If for some reason 120-ohm cable cannot be used, it is recommended that the worst-case loading (the number of transmitters and receivers that can be used) and worst-case common-mode voltage ranges be recalculated to make sure the system under design will work. Publication TSB89 has a section specifically devoted to such calculations.

Number of Twisted Pairs per Transmitter

Now that we have a feel for the type of wire needed, the question arises as to how many twisted pairs a transmitter can drive. The short answer is exactly one. Although it is possible for a transmitter to drive more than one twisted pair under certain circumstances, this is not the intent of the specification.

Termination Resistors

Because of the high frequencies and the distances involved, proper attention must be paid to transmission-line effects. However, a thorough discussion of transmission-line effects and proper termination techniques are well beyond the scope of this application note. With this in mind, terminations will be briefly discussed in their simplest form as they relate to RS-485.

A terminating resistor is simply a resistor that is placed at the extreme end or ends of a cable (Figure 4). The value of the terminating resistor is ideally the same value as the characteristic impedance of the cable.

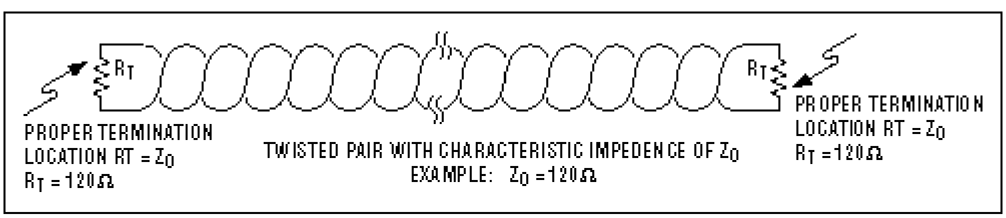


Figure 4. Termination resistors should be the same value of the characteristic impedance of the twisted pair and should be placed at the far ends of the cable.

When the termination resistance isn't the same value as the characteristic impedance of the wiring, reflections will occur as the signal is traveling down the cable. This is governed by the equation (Rt-Zo)/(Zo+Rt), where Zo is the impedance of the cable and Rt is the value of the terminating resistor. Although some reflections are inevitable due to cable and resistor tolerances, large enough mismatches can cause reflections big enough to bring about errors in the data. See Figure 5.

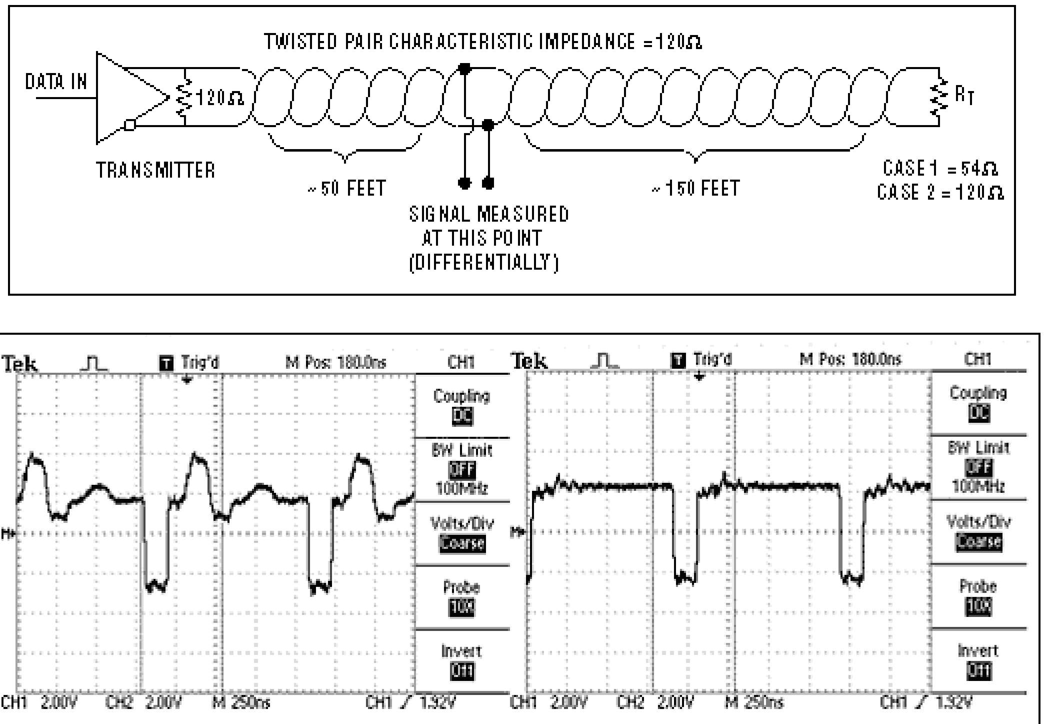


Figure 5. Using the circuit shown at the top, the waveform on the left was obtained with a MAX3485 driving a 120-ohm twisted pair terminated with 54 ohms. The waveform on the right was obtained with the cable terminated properly with 120 ohms.

With this in mind, it is important to match the terminating resistance and the characteristic impedance as closely as possible. The position of the terminating resistors is also very important. Termination resistors should always be placed at the far ends of the cable.

As a general rule, termination resistors should be placed at *both* far ends of the cable. Although properly terminating both ends is absolutely critical for most system designs, it can be argued that in one special case only one termination resistor is needed. This case occurs in a system when there is a single transmitter and that single transmitter is located at the far end of the cable. In this case it is unnecessary to place a termination resistor at the end of the cable with the transmitter, because the signal is intended to always travel *away* from this end of the cable.

Maximum Number of Transmitters and Receivers on a Network

The simplest RS-485 network is comprised of a single transmitter and a single receiver. Although useful in a number of applications, RS-485 allows for greater flexibility by permitting multiple receivers and transmitters on a single twisted pair. The maximum allowed depends on how much each device loads down the system.

In an ideal world, all receivers and inactive transmitters will have infinite impedance and will not load the system down in any way. In the real world, however, this isn't the case. Every receiver attached to the network and all inactive transmitters will add an incremental load. To help the designer of an RS-485 network figure out just how many devices can be added to a network, a hypothetical unit called a "unit load" was created. All devices that are connected to an RS-485 network should be characterized in regard to multiples or fractions of unit loads. Two examples are the MAX3485, which is specified at 1 unit load, and the MAX 487, which is specified at 1/4 of a unit load. The maximum number of unit loads allowed on a twisted pair, assuming a properly terminated cable with a characteristic impedance of 120 ohms or more, is 32. Using the examples given above, this means that up to 32 MAX3485s or up to 128 MAX487s can be placed on a single network.

Examples of Proper Networks

With the above information, we are ready to design some RS-485 networks. Here are a few examples.

One Transmitter, One Receiver

The simplest network is one transmitter and one receiver (Figure 6). In this example, a termination resistor is shown at the transmitter end of the cable. Although unnecessary here, it is probably a good habit to design in both termination resistors. This allows the transmitter to be moved to locations other than the far end and permits additional transmitters to be added to the network should that become necessary.

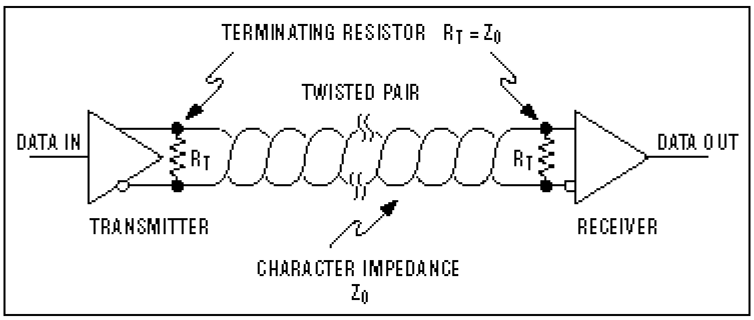


Figure 6. A one-transmitter one-receiver RS-485 network

One Transmitter, Multiple Receivers

Figure 7 shows a one-transmitter multiple-receivers network. Here, it is important to keep the distances from the twisted pair to the receivers as short as possible.

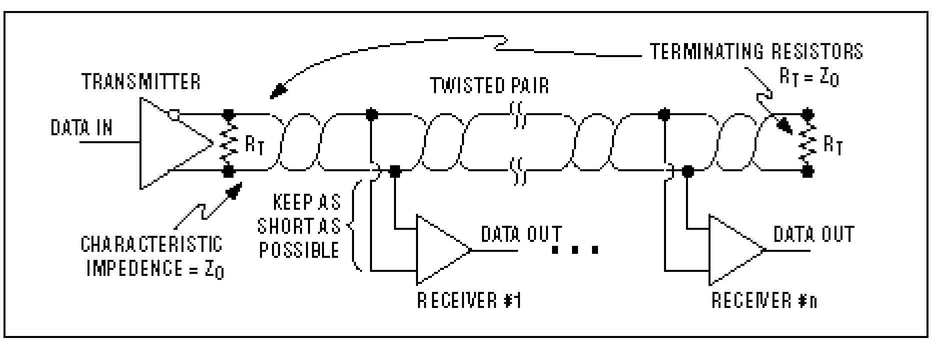


Figure 7. A one-transmitter multiple-receivers RS-485 network

Two Transceivers

Figure 8 shows a two-transceivers network.

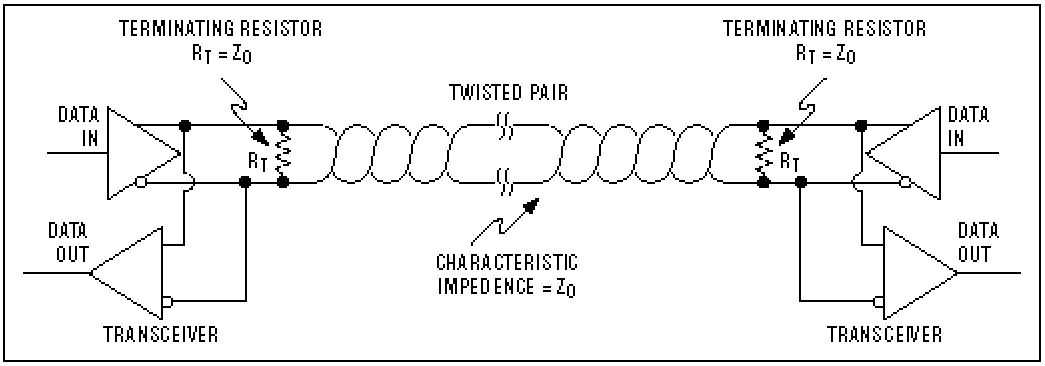


Figure 8. A two-transceivers RS-485 network

Multiple Transceivers

Figure 9 shows a multiple-transceivers network. As in the one-transmitter and multiple-receivers example, it is important to keep the distances from the twisted pair to the receivers as short as possible.

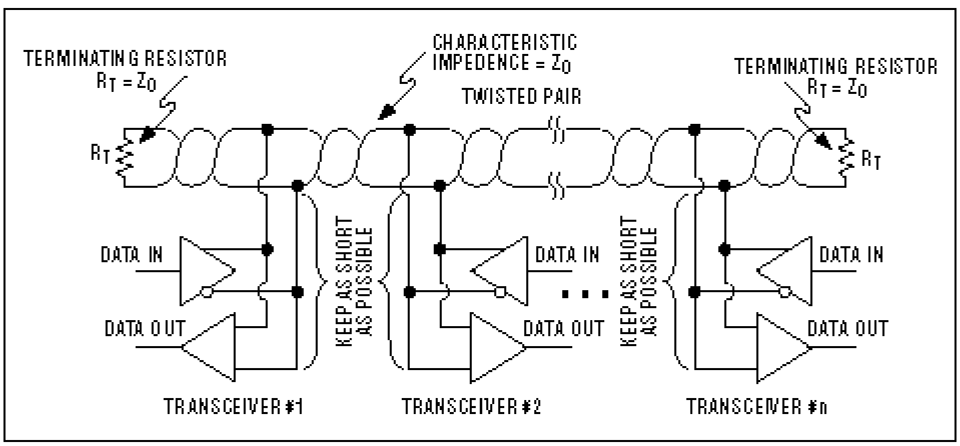


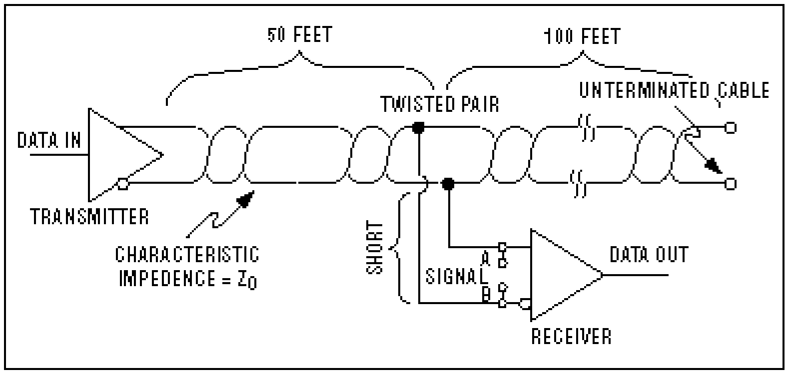
Figure 9. A multiple-transceivers RS-485 network

Examples of Improper Networks

The diagrams below are examples of improperly configured systems. Each example shows the waveform obtained from the improperly designed network and compares it to a waveform from a properly designed system. The waveform is measured differentially at points A and B (A-B).

Unterminated Network

In this example, the ends of the twisted pair are unterminated. As the signal propagates down the wire, it encounters the open circuit at the end of the cable. This constitutes an impedance mismatch, bringing about reflections. In the case of an open circuit (as shown below), all of the energy is reflected back to the source, causing the waveform to become very distorted.



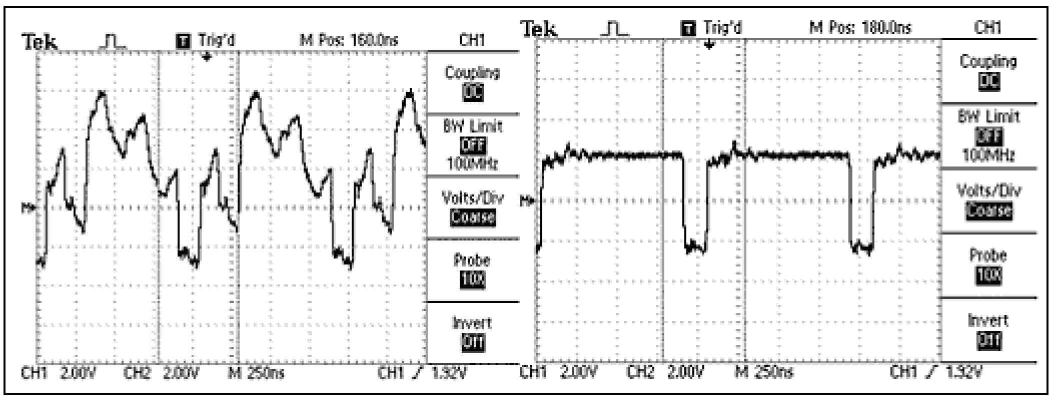
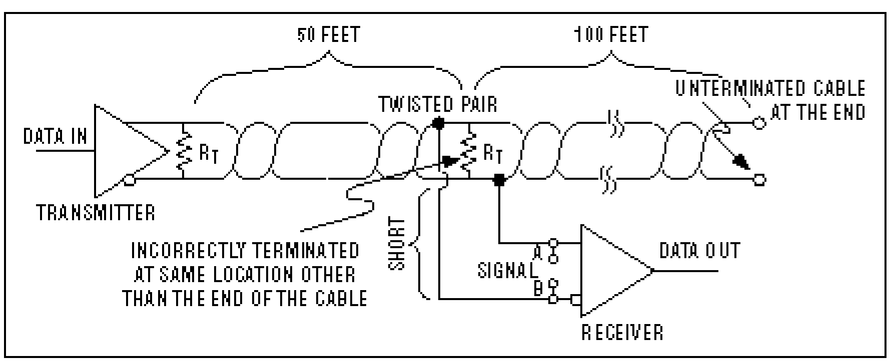


Figure 10. An unterminated RS-485 network (top) and its resultant waveform (left), compared with a waveform obtained from a correctly terminated network (right)

Wrong Termination Location

Figure 11 shows a termination resistor, but it is located in a position other than the far end of the cable. As the signal propagates down the cable, it encounters two impedance mismatches. The first occurs at the termination resistor. Even though the resistor is matched to the characteristic impedance of the cable, there is still cable after the resistor. This extra cable causes a mismatch and therefore reflections. The second mismatch is at the end of the unterminated cable, leading to further reflections.



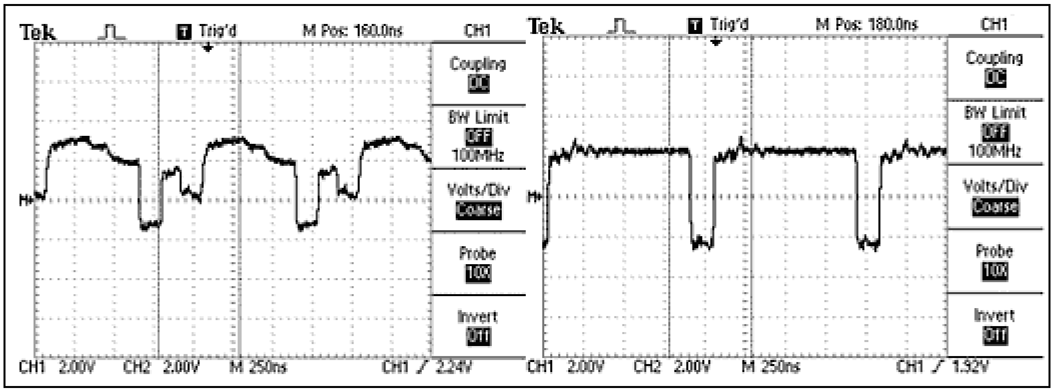


Figure 11. An RS-485 network with the termination resistor placed at the wrong location (top) and its resultant waveform (left), compared to a properly terminated network (right)

Multiple Cables

In Figure 12, there are multiple problems with the layout. The first problem is that RS-485 drivers are designed to drive only a single, properly terminated twisted pair. Here, the transmitters are each driving four twisted pairs in parallel. This means that the required minimum logic levels cannot be guaranteed. In addition to the heavy loading, there is an impedance mismatch at the point where multiple cables are connected. Impedance mismatches again mean reflections and therefore signal distortions.

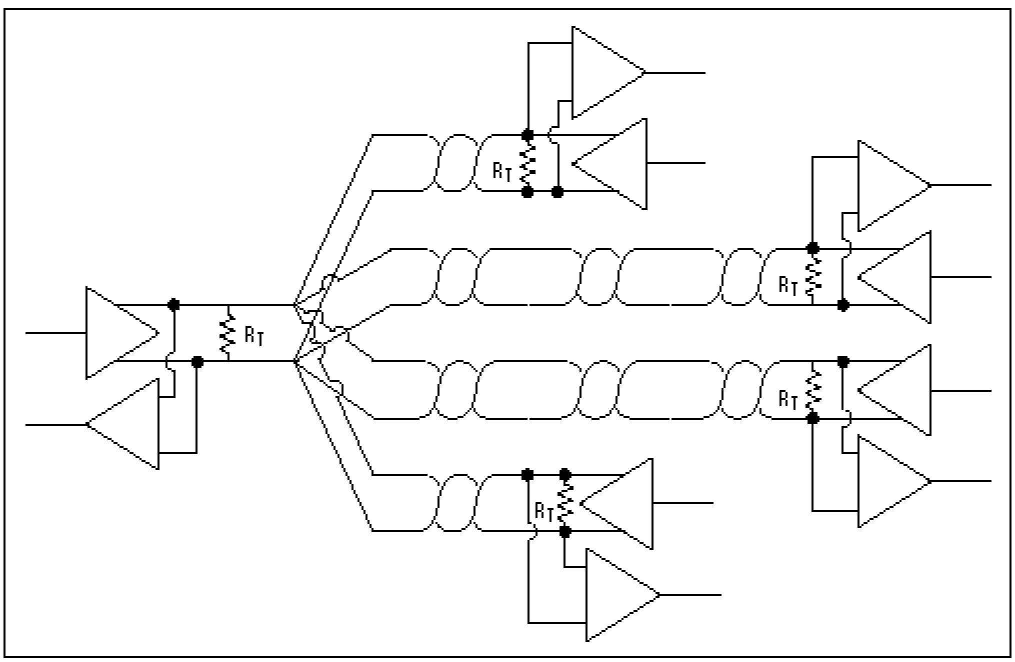


Figure 12. An RS-485 network that uses multiple twisted pairs incorrectly

Long Stubs

In Figure 13, the cable is properly terminated and the transmitter is driving only a single twisted pair; however, the connection point (stub) for the receiver is excessively long. A long stub causes a significant impedance mismatch and thus reflections. All stubs should be kept as short as possible.

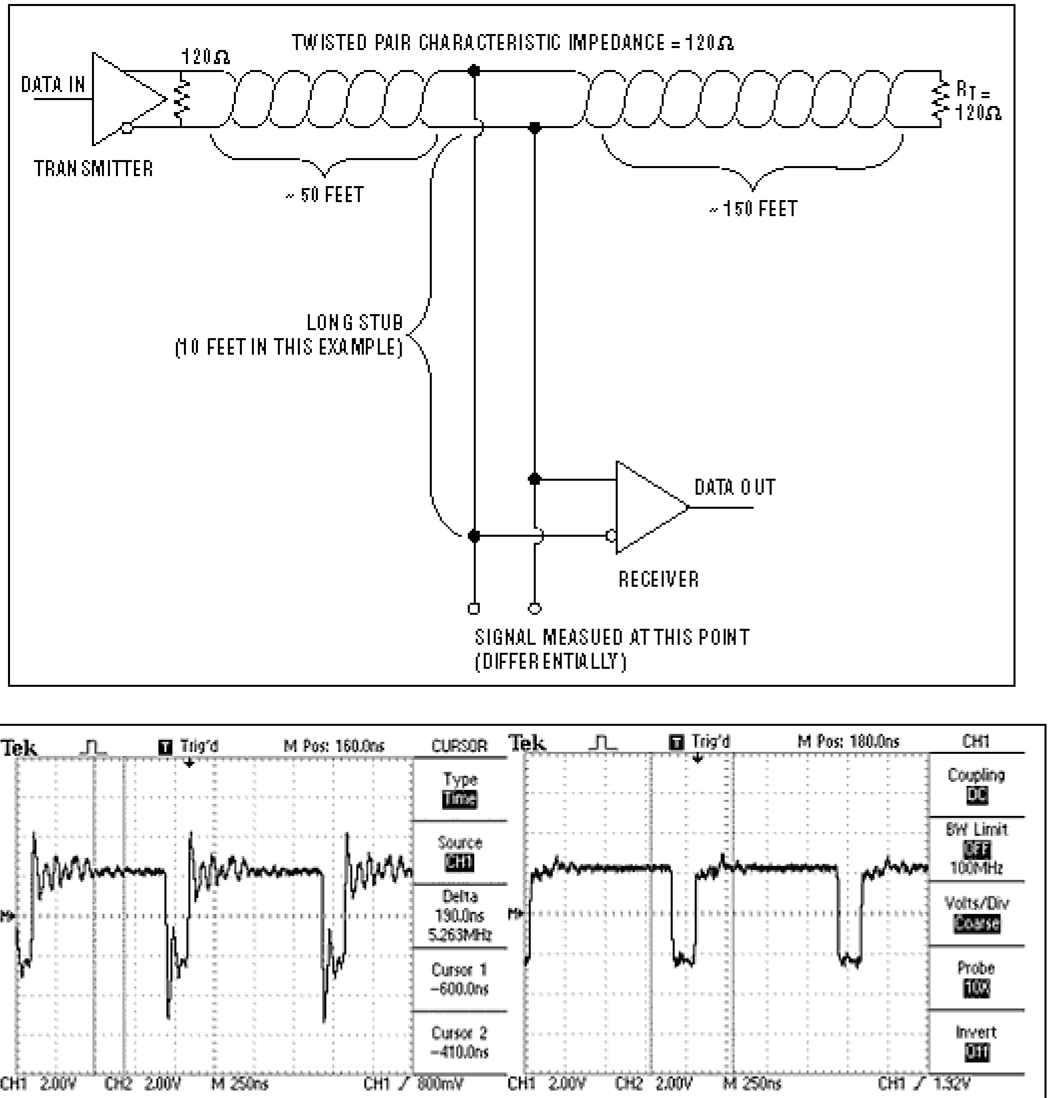


Figure 13. An RS-485 network that has a 10-foot stub (top) and its resultant waveform (left), compared to a waveform obtained with a short stub

References

1. TIA/EIA-485-A *Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*
2. TSB89 *Application Guidelines for TIA/EIA-485-A*