MONITORING & CONTROL REFERENCE

Objective

ComBlock modules can be controlled and monitored remotely from an external computer over serial link, LAN/TCP-IP, USB, PCMCIA and CardBus. This document describes the addressing scheme and the messages.

Most ComBlock users take advantage of the user-friendly graphical user interface (ComBlock Control Center) to monitor and control ComBlock assemblies, in which case this document is not needed.

Programmers wishing to develop alternative graphical user interface or automated controls should read this document and concurrently trace the M&C messages using the terminal emulator within the ComBlock Control Center.

Electrical Interface

Comblock(s)– Control Entity

A single connection is sufficient for the external computer to communicate with all microprocessor equipped ComBlock module in a given assembly.

LAN: TCP-IP socket in listen mode. ComBlock port is 1028.

Serial: asynchronous serial, 115.2 Kbps, 8 bit, no parity, one stop bit, send a carriage return with every line feed. See Appendix for a typical Hyperterminal configuration.

USB (support for multiple plug and play USB devices)

PCMCIA/CardBus: (support for multiple plug and play PCMCIA/CardBus devices).

Comblock – Comblock

Each module-to-module communication connection is a duplex asynchronous serial link comprising two signals: transmit and receive. The settings is 115 Kbaud/s, 8 bit, no parity, 1 stop bit. Levels are low voltage TTL (3.3V).

Message Structure

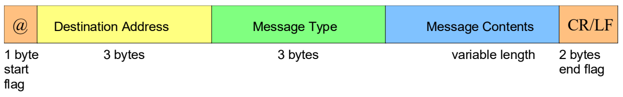
A message is delineated with a start character '@' and end characters CR/LF. These characters shall never occur in the middle of a message.

Each message comprises three fields:

(a) Three-character destination address field.

(b) Three-character message type (cap sensitive)

(c) Variable-length message contents field. Always an integer number of bytes. Length is implied by the message type.



All fields use readable ASCII characters, in upper case letters when applicable. The least significant bit of the most significant byte (left bit) is transmitted first. The maximum legal message length is 272 bytes, including the overhead, or 273 bytes when including the 0 end of string character within programs.

Address

Each ComBlock is identified uniquely by its address. Addresses are assigned by the ComBlock control center at power up. Special addresses are reserved:

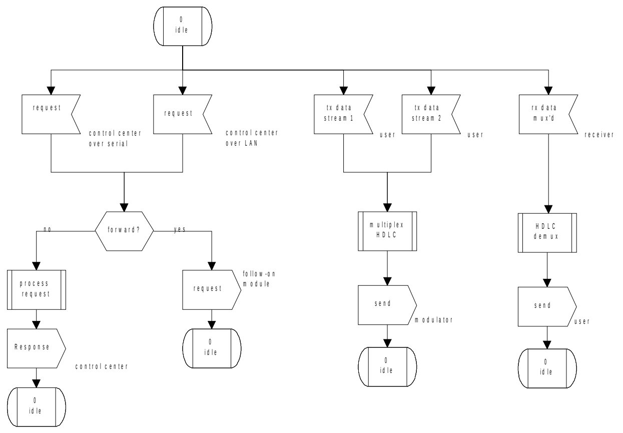
- 000 is the default address of a ComBlock module at power up.

- 999 is the address of the ComBlock control center.
- 111 is the broadcast address. Messages are forwarded to all modules.

A basic rule is that ComBlock modules only respond to queries. They never take the initiative of sending messages.

The architecture is a star network. The ComBlock modules can only communicate with the external computer. Module to module communication is not permitted at this time.

ComBlock Message Processing



Message Type

Most messages are specific to the module context. They are therefore described in the module specifications document. The common messages are described below.

Successive commands with no response should be separated by 5ms to give the module time to execute the command.

Messages are always in upper case letters.

Quick reference table

Command	***Name***	***COM-1000/8000/ 1100/1200/1300/1400/ 1500/1600/1700/3011/ 1800/1900 FPGA-based platforms***	***COM-2001, 3001/2/3/4/5/6, 3501, 3505, 3506, 4004***	***COM-4001/2/3/5/6, 4101 platform***	***Specific board support***
EFS	Erase File Sector	x	n/a	n/a	
GAI	Get Assembly Identification	x	3505		
GAS	Get Assembly Serial number	x	3505		
GCA	Get Configuration start Address	x	n/a	n/a	
GCS	Get CheckSum	x	n/a	n/a	
GDC	Get Default Configuration	x	3505	n/a	
GMI	Get Module Identification	x	x	x	
GRG	Get control Register	x	x	x	
GMR	Get Multiple control Registers	1500, 1700,1800,1900	3506		
GRT	Get control Register Temporary	x	x		
GMT	Get Multiple control Registers Temporary	1500, 1700,1800,1900	x		
GSN	Get Serial Number	x	x	x	
GSR	Get Status Register	x	x		
GMS	Get Multiple Status registers	1500,1800,1900	3506		
MFW	Message FordWarding	x	x	x	
PCF	Protect ConFiguration	x	n/a	n/a	
RMD	Read Memory Data	x	n/a	n/a	
RSP	Reset IP Port	n/a	n/a	n/a	
RST	Reset	x	x	x	
SAC	Set Address Command	x	x	x	
SCA	Set Configuration start Address	x	n/a	n/a	
SDC	Set Default Configuration	x	3505	n/a	
SMI	Set Module Identification	x	x	x	
SPW	Set Password	n/a	n/a	n/a	
SRG	Set Register	x	x	x	
SRT	Set Register	x	x		

	Temporary				
WFS	Write File Sector	x	n/a	n/a	

Message RST "ReSeT" Command

Resets the module address to zero, reloads FPGAs as needed, reconfigure the FPGA with the default register values stored in non-volatile memory.
Message type: **RST**
Message contents: N/A.
Message contents length: 0 byte.
No response is required.

Message RSP "ReSet IP Port" Command

Resets a specific IP port. This message is typically sent to the network interface card (COM-5001) over a UDP-IP datagram in order to reset a 'stuck' TCP-IP port remotely.
Message type: **RSP**
Message contents:

- 4-digit port number expressed in readable ASCII text. For example @001RSP1024 to reset port 1024.

Message contents length: 4 bytes.
No response is required.

Message SAC "Set Address Command"

Message type: **SAC**
Message contents: address.
Message contents length: 3 bytes.
No response is required.

Message MFW "Message Forwarding"

Message type: **MFW**
Message contents:

- 0 to stop previously defined forwarding.
- n to forward all subsequent messages from the control center to port n, where n is in the range from 1 to 4.
- 9 to forward all subsequent messages to all adjacent modules.

Message contents length: 1 byte.
No response is required.

Message SMI "Set Module Identification"

The "Set Module Identification" command allows the user to declare the several FPGA configurations stored within a given ComBlock module.For example, @000SMI011027<enter> declares that configuration 01 is to be the FSK demodulator 1027. `For example: @000SMI1027<enter>

Message type: **SMI**
Message contents :

- optional 2 digit configuration number [NEW for modules shipped prior to 10/04]. Configurations range from 01 to 99. Applicable only for FPGA-based ComBlocks.If no configuration number is supplied, the SMI command is applied to the default configuration selected by the SDC command.
- 4 character module type
- 1 character module option (typically a letter, can be a space if basic module). Not applicable for FPGA based ComBlocks.
- 1 character module revision (typically a digit). Not applicable for FPGA based ComBlocks.

Message contents length: 4 or 6 bytes for FPGA-based ComBlocks. 6 bytes for ComBlocks without FPGA.

Message GMI "Get Module Identification command"

Message type: **GMI**
Message contents:

- none to inquire about the current default configuration. Example : @000GMI, ***or***
- use two-digit "00" configuration number to inquire about the ComBlock hardware platform. Example : @000GMI00. ***or***
- use two-digit configuration number to read one of several possible configurations (personalities) stored within the ComBlock. For example, @000GMI01 inquires about configuration 1. Configurations range from 01 to 99.

Message contents length: 0 byte or 2 bytes.

Reply with the module identification message MID for the selected configuration.
Store the port number through which the command was received. It is the return path for all responses back to the control center.

Message MID "Module IDentification"

Message type: **MID**
Message contents :

- 4 character module type
- 1 character module option (typically a letter, can be a space if basic module)
- 1 character module revision (typically a digit). Refers to the FPGA firmware for FPGA-based ComBlock modules.
- 1 character port number through which command was received.
- optional 1 character for future use.
- optional 1 hexadecimal character for status:
    - bit 0 = configuration authorization
    - bit 1 = configuration protection
    - bit 2 = FPGA DONE pin is high

Message contents length: 7 (legacy software) or 9 bytes.
This message is sent back to the control center in response to a "Get module identification command" GMI message.

Warning: depending on the module, it may not be possible to read the FPGA option and revision for personalities others than the current active one.

Message GSN "Get Serial Number" command

Message type: **GSN**
Message contents: n/a
Message contents length: 0 byte.
Reply with the module serial number response MSN.
The 10 character serial number is programmed once at manufacturing.

Message MSN "Module Serial Number" response

Message type: **MSN**
Message contents :

- 10 character serial number

Message contents length: 10 bytes.
This message is sent back to the control center in response to a « Get Serial Number » GSN command.

Message GAI "Get Assembly Identification command"

This field is provided for users to identify complete ComBlock assemblies. This information is generally stored in the first module interfacing directly with the host computer (i.e. LAN interface or serial interface).
Message type: **GAI**
Message contents: n/a
Message contents length: 0 byte.
Reply with the assembly identification message AID.

Message AID "Assembly IDentification"

Message type: **AID**
Message contents :

- 4 character assembly type
- 1 character assembly option (typically a letter, can be a space if baseline assembly)
- 1 character assembly revision (typically a digit)

Message contents length: 6 bytes.
This message is sent back to the control center in response to a « Get assembly identification command » GAI message.

Message GAS "Get Assembly Serial number" command

Message type: **GAS**
Message contents: n/a
Message contents length: 0 byte.
Reply with the module serial number response ASN.
The 10 character serial number is programmed once at manufacturing.

Message ASN "Assembly Serial Number" response

Message type: **ASN**
Message contents :

- 10 character serial number

Message contents length: 10 bytes.

This message is sent back to the control center in response to a « Get Assembly Serial number » GAS command.

Message EFS "Erase File Sector" command

A file is segmented into one or multiple sectors. Prior to writing into a sector, flash memories require that the sector is erased. The number and size of sectors is determined by the Flash memory technology used in each ComBlock module. This information is known a priory by the comblock control center.

COM-100x uses 3 sectors of size 64 KB each.
COM-5001 uses 3 sectors of size 64 KB each.
COM-700x uses 3 sectors of size 64 KB each.
COM-800x uses 4 sectors of size 64 KB each.
COM-1100 uses 8 sectors of size 64 KB each.
COM-1200 uses 15 sectors of size 64 KB each.
COM-1300 uses 4 sectors of size 64 KB each.
COM-1400 uses 4 sectors of size 64 KB each.
COM-1500 uses 33 128KB NAND flash blocks
COM-1600 uses 4 128KB NAND flash blocks
COM-1700 uses 12 128KB NAND flash blocks
COM-3011 uses 13 128KB NAND flash blocks
COM-1800 uses 75 128KB NAND flash blocks
COM-1900 uses 75 128KB NAND flash blocks

Message type: **EFS**
Message contents:
Sector number (3 digits, range 001 - 999)
Message contents length: 3 bytes.
Reply with positive (ACK) or negative (NAK).
A negative acknowledgement (NAK) is returned if the sector number is not within the allowed range or if the sector is protected.

Message WFS "Write File Sector" command

The data to be programmed into a given sector is segmented into several packets. Each packet contains a sequence number. Each packet is transmitted with the maximum data field size of 128 bytes, except for the last packet. The maximum packet size of 128 bytes is dictated by the RAM memory available on some ComBlock Atmel 8-bit AVR microprocessor. The last packet is identified by the special address 9999. The data is written consecutively into the flash memory (no gaps).

Message type: **WFS**
Message contents:

- Sector number (3 digits, range 001 - 999)
- Packet sequence number (4 digits, range 0001 - 9999). 9999 indicates the last packet in the file.
- Data field (up to 128 data bytes represented in ASCII hexadecimal (caps, 2 ASCII characters per byte).

Message contents length: variable from 9 to 263 bytes.
After each WFS command, an ACK message is returned with the 6-digit cumulative number of bytes received since the start of sector. This is a form of flow control. The ComBlock control center shall not send the following packet until an acknowledgement is received first. A negative acknowledgement (NAK) is returned if the sector number is not within the allowed range or if the sector is protected.

Message GCS "Get CheckSum" command

After a file sector has been downloaded, one can query the resulting checksum. The checksum is computed as the sum of all bytes within the specified sector. Bytes not previously written to are initialized to 0xFF by default. The 16-bit checksum is returned as part of the CKS checksum message. Returns NAK in case of error (wrong sector number).

Message type: **GCS**
Message contents:

- Sector number (3 digits, range 001 - 999)

Message contents length: 3 bytes.

Message CKS "ChecKSum" response

Message sent in response to the get checksum command GCS.

Message type: **CKS**
Message contents:

- 16-bit checksum. 4 Hex ASCII characters.

Message contents length: 4 bytes.

Message SPW “Set PassWord” command

This command is to authorize a specific software personality on a specific ComBlock module. The 16 hex digits password is provided by ComBlock at the time of purchase. The password is a unique for a given ComBlock module serial number and software personality. The SPW command sends the

Message type: **SPW**
Message contents:

- 2 digit configuration number in the range from 01 to 99.
- 4-digit personality (for example 1001, 8002, etc)
- 16 hex characters key1, as provided at the time of purchase.

Message contents length: 22 bytes.
No response is provided. The user can verify that the SPW command resulted in a valid configuration authorization by reading the configuration status using the GMI command.

Message SRG “Set Register” command

ComBlock modules are monitored and controlled by means of 8-bit registers. The register value is stored in persistent (non-volatile) memory within the ComBlock, so that it can be automatically reloaded at power up or reset without intervention by the control center.
Message type: **SRG**
Message contents:

- Register number (2 digits, range 01 - 99)
- Register byte: ASCII Hex. 2 characters.

Message contents length: 4 bytes.
No response is provided.

Message SRT “Set Register (Temporary)” command

Same as SRG “Set Register” command, except that the register value is stored in volatile (RAM) memory. This command should be used when frequently updating registers so as not to exceed the 100,000 write cycle limit of the EEPROM non-volatile memory.

Message type: **SRT**
Message contents:

- Register number (2 digits, range 01 – 99, or 3 digits, range 001 - 999)
- Register byte: ASCII Hex. 2 characters.

Message contents length: 4 or 5 bytes.
No response is provided.

Message GRG “Get Register” command

ComBlock modules are controlled by means of 8-bit command registers, the value of which is stored in non-volatile memory within each ComBlock. This requests a specific command register value to be returned. The response message is RGV.
Message type: **GRG**
Message contents:

- Register number (2 digits, range 01 - 99)

Message contents length: 2 bytes.

Message GMR “Get Multiple Registers” command

Read multiple control registers in one transaction. The response message is MRV.
Message type: **GMR**
Message contents:

- Number of control registers to read (3 digits, range 000 - 128)
- Start register number read (3 digits, range 000 -255)

Message contents length: 6 bytes.

Message GRT “Get Register (Temporary)” command

Same as GRG “Get Register” command, except that the register value read back is stored in volatile (RAM) memory.

Message GMT “Get Multiple Temporary registers” command

Same as GMR “Get Multiple Registers” command, except that the registers values read back are stored in volatile (RAM) memory.

Message GSR “Get Status Register” command

ComBlock modules are monitored by means of 8-bit registers. This requests a specific register value to be returned. The response message is RGV.
Message type: **GSR**
Message contents:

- Register number (1, 2 or 3 digits, ranging from 0 to 999). No space allowed.

Message contents length: 1,2 or 3 bytes.

Message GMS “Get Multiple Status registers” command

Read multiple status registers in one transaction. The response message is MRV.
Message type: **GMR**
Message contents:

- Number of status registers to read (3 digits, range 000 - 128)
- Start register number read (3 digits, range 000 -255)

Message contents length: 6 bytes.

Message RGV “ReGister Value” response

Message in response to a get register command.
Message type: **RGV**
Message contents:

- Register value in ASCII Hex. 2 characters.

Message contents length: 2 byte.
Make sure any leading zero is included in the response.

Message MRV “Multiple Registers Values” response

Message in response to a get multiple registers command.
Message type: **MRV**
Message contents:

- Register value in ASCII Hex. 2*n characters.

Message contents length: 2*n bytes, where n is the number of registers queried.
Make sure any leading zero is included in the response.

Message RMD “Read Memory Data” command

Data stored in the ComBlock memory (generally the FPGA dual port RAMs) can be exported by using the RMD command.

Message type: **RMD**
Message contents:

- Offset address: 4 bytes expressed in hexadecimal format, 8 characters in natural reading order (left is MSB, right is LSB.
- Number of bytes to be read: 4 bytes expressed in hexadecimal format, 8 ASCII characters in natural reading order (left is MSB, right is LSB.

Message contents length: 16 bytes.

After each RMD command, the ComBlock generally returns a MDA message containing the data read. The ComBlock module could also return a negative acknowledgement (NAK) in case of error.

Message MDA “Memory DAta” command

The message is used to return the data read in response to a ‘RMD read volatile memory command to the host computer.

Message type: **MDA**
Message contents:

- Data field (length equals twice the number of bytes to be read specificed in the RMD command. Bytes read are expressed ASCII hex format (caps, 2 ASCII characters per byte).

Message contents length: variable

Message ACK “Acknowledgement” response

This positive acknowledgement message is returned upon execution of a command, whenever specified. The acknowledgement can be followed by attributes, as specified in the command that triggered it.
Message type: **ACK**
Message contents:

- Context specific. Depends on the command.

Message contents length: variable.

A negative acknowledgement NAK is returned if the register number is out of range for the specified module.

Message NAK "Negative AcKnowledgement" response

This negative acknowledgement message is returned upon failure to execute a command, whenever specified. This negative acknowledgement can be followed by attributes, as specified in the command that triggered it.

Message type: **NAK**
Message contents:
- Context specific. Depends on the command.

Message contents length: variable.

Message SDC "Set Default Configuration command"

The "Set Default Configuration" command selects the default FPGA configuration to be used at power up or reset. This configuration is stored in non-volatile memory within the ComBlock. This command only applies to FPGA-based ComBlock modules. To enact the change in default configuration, use the RST reset command.

Message type: **SDC**
Message contents:
- two-digit decimal configuration number to point one of several possible configurations stored within the ComBlock. For example, @000SDC01 declares configuration 01 as the default. Configurations range from 01 to 99.
  Format requires a leading 0.

Message contents length: 2 bytes.
No response is provided.

Valid range for the configuration number depends on the ComBlock platform:
COM-1000: 01 to 02
COM-8000: 01 to 02
COM-1100: 01 to 01
COM-1200: 01 to 04
COM-1300: 01 to 15
COM-1400: 01 to 07
COM-1600: 01 to 08
COM-1700: 01 to 08
COM-1800: 01 to 04
COM-1900: 01 to 04

SDC commands outside the allowed configuration number range are ignored.

Message GDC "Get Default Configuration" command

Use the "Get Default Configuration" to read back the default configuration within a given ComBlock. Usage: @000GDC<enter>.

Message type: **GDC**
Message contents: n/a
Message contents length: 0 byte.
ComBlock replies with the "Default Configuration" DCF message.

Message DCF "Default ConFiguration" message

Message type: **DCF**
Message contents :
- two-digit configuration number pointing to the default configuration. For example, @999DCF01 statest that the default configuration is 01.

Message contents length: 2 bytes.
This message is sent back to the control center in response to a "Get Default Configuration" GDC command.

Message PCF "Protect ConFiguration" command

Each FPGA configuration stored in non-volatile (Flash) memory can be individually protected / unprotected. This command only applies to FPGA-based ComBlock modules. Once protected, a given configuration cannot be erased or written over. The protection bit is set in non-volatile memory and can be read back using the GMI command.

Important note: many FPGA-based Comblock platforms are shipped with configuration 01(the 'boot' configuration) already protected. Removing the protection on this 'boot' configuration may result in total loss of communication with the ComBlock if this boot configuration is subsequently accidentally erased.

Message type: **PCF**

Message contents:

- two-digit configuration number to point one of several possible configurations stored within the ComBlock. Configurations range from 01 to 99.
  Format requires a leading 0.
- one digit 'P' for protected, 'U' for unprotected.

Message contents length: 3 bytes.
No response is provided.

Message SCA "Set Configuration start Address command"

The SCA command tells the ComBlock where a given FPGA configuration is mapped within the Flash memory. The SCA associates a start address to a configuration number. This command only applies to FPGA-based ComBlock modules.
Message type: **SCA**
Message contents:

- two-digit configuration number to point one of several possible configurations stored within the ComBlock. Configurations range from 01 to 99.Format requires a leading 0.
- Start address: 4 bytes expressed in hexadecimal format, 8 characters in natural reading order (left is MSB, right is LSB.

Message contents length: 10 bytes.
No response is provided.

Usage: @000SCA010123456F<enter> states that the start address in the Flash memory for FPGA configuration 01 is 0x01 23 45 6F.

Message GCA "Get Configuration start Address command"

Read back the start address for a specific FPGA configuration file stored in flash memory. The configuration is identified by its two digit configuration number. The ComBlock responds to the query with a CSA "Configuration Start Address" message.

Message type: **GCA**
Message contents:

- two-digit configuration number to point one of several possible configurations stored within the ComBlock. Configurations range from 01 to 99.Format requires a leading 0.

Message contents length: 2 bytes.

Message CSA "Configuration Start Address" message

Message type: **CSA**
Message contents:

- Start address: 4 bytes expressed in hexadecimal format, 8 characters in natural reading order (left is MSB, right is LSB.

Message contents length: 8 bytes.

Example

The following is a typical sequence from the control center to the ComBlock assembly, as part of the 'discovery' process, whereby the control center is trying to find out how modules are interconnected.

From control center	From ComBlock module	Note
@111SAC000		Broadcast to all: "Reset module ID to 0". Message is sent once.
@111MFW0		Broadcast to all: "Turn off all forwarding". Sent once.
@000SAC001		Set first module ID to 1
@001GMI		Find out module 1's type
	@999MID1001A11	Type is modulator (COM-1001) option A, rev 1. Connection to control center is via port 1 (serial port)
@001MFW2		Route subsequent messages to port2 (right connector)
@000SAC002		Set next module ID to 2
@002GMI		Find out module 2's type
	Timeout	No module connected to port 2
@001MFW3		Route subsequent messages to port3 (bottom connector)
@000SAC002		Set next module ID to 2
@002GMI		Find out module 2's type
	@999MID2001 11	Type is digital to analog board (COM-2001), no option, rev 1. Connection is via

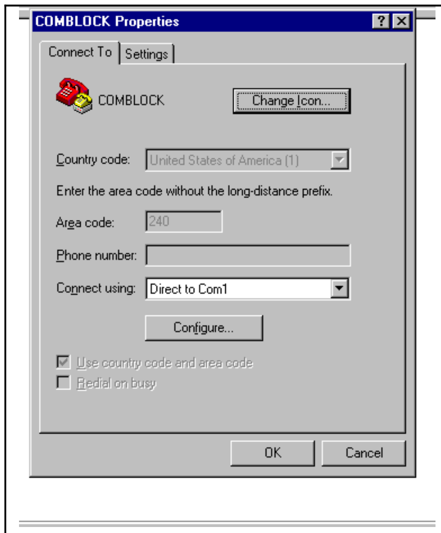
		port 1
Etc……		
@111MFW9		Last command: broadcast to all modules: “forward messages to all ports”
@111MFW9		Last command: broadcast to all modules: “forward messages to all ports”
@111MFW9		Last command: broadcast to all modules: “forward messages to all ports”
Repeat a number of times (depending on the level depth).		

Tracing messages

The messages exchanged between host computer and ComBlock assembly can be traced and saved using the ComBlock Control Center graphical user interface. The trace is enabled by activating the terminal emulator function of the ComBlock Control Center (right most button). A window pane will open on the right with all messages to and from the ComBlock assembly. One can cut and paste these messages into another document.

Appendix : Hyperterminal configuration

Hyperterminal is a utility found in most Windows operating systems. It allows to communicate with the ComBlock over a simple serial link using text-like commands and responses. It is usually found under the Start | Programs | Accessories menu in Windows.



COMBLOCK Properties

Connect To | Settings

COMBLOCK [Change Icon...]

Country code: United States of America (1)

Enter the area code without the long-distance prefix.

Area code: 240

Phone number:

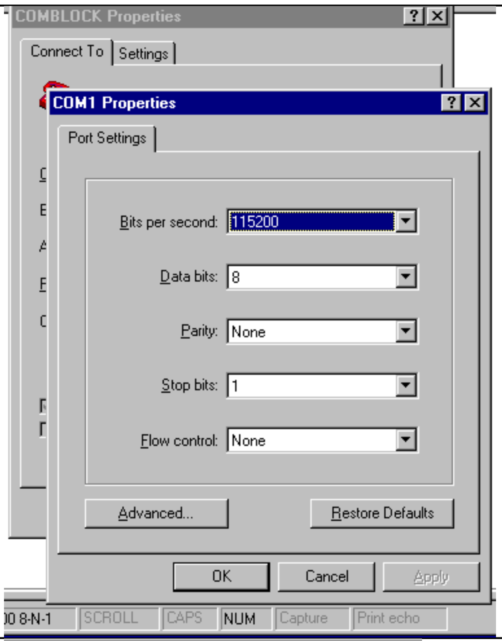
Connect using: Direct to Com1

[Configure...]

☑ Use country code and area code

☐ Redial on busy

[OK] [Cancel]



COM1 Properties

Port Settings

Bits per second: 115200

Data bits: 8

Parity: None

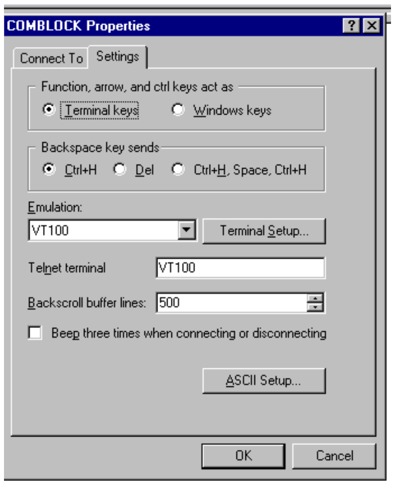
Stop bits: 1

Flow control: None

[Advanced...] [Restore Defaults]

[OK] [Cancel] [Apply]

00 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo



COMBLOCK Properties

Connect To | Settings

Function, arrow, and ctrl keys act as: [✓ Terminal keys] Windows keys

Backspace key sends: [✓ Ctrl+H] Del Ctrl+H, Space, Ctrl+H

Emulation: VT100 [Terminal Setup...]

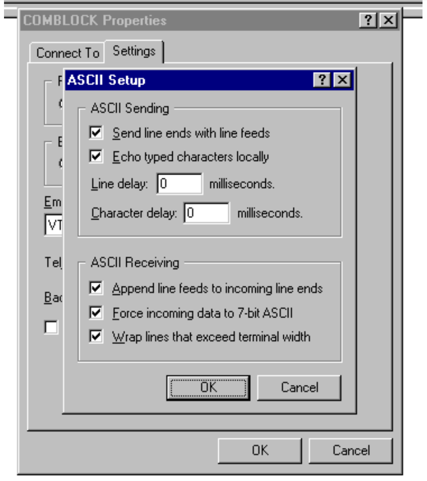
Telnet terminal: VT100

Backscroll buffer lines: 500

☐ Beep three times when connecting or disconnecting

[ASCII Setup...]

[OK] [Cancel]



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

ASCII Receiving

☑ Append line feeds to incoming line ends

☑ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

[OK] [Cancel]