# SuperMon 5.5 Operation Manual

## Table of Contents:

**Section 1: SuperMon overview**

**Section 2: Configuration and operation of SuperMon Monitor**

**Section 3: Configuration and operation of SuperMon Sniff and Spoof**

**Section 4: Configuration and operation of SuperMon Logging**

**Section 5: File format of Logging files**

**Section 6: Configuration and Operation of Protocol Specific Screens**

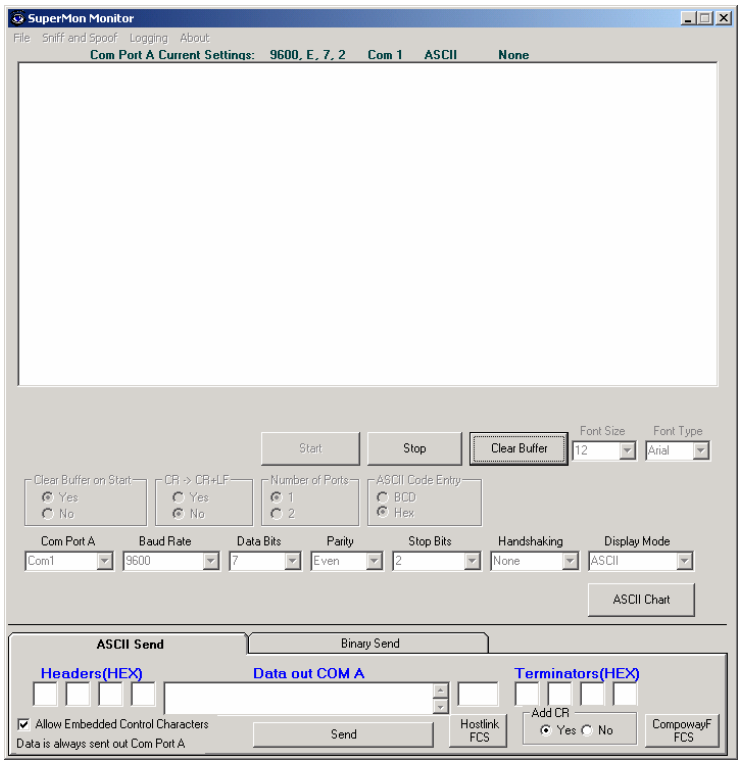
**Section 7: Cable Diagrams for use with SuperMon**

**Section 8: Revision History**

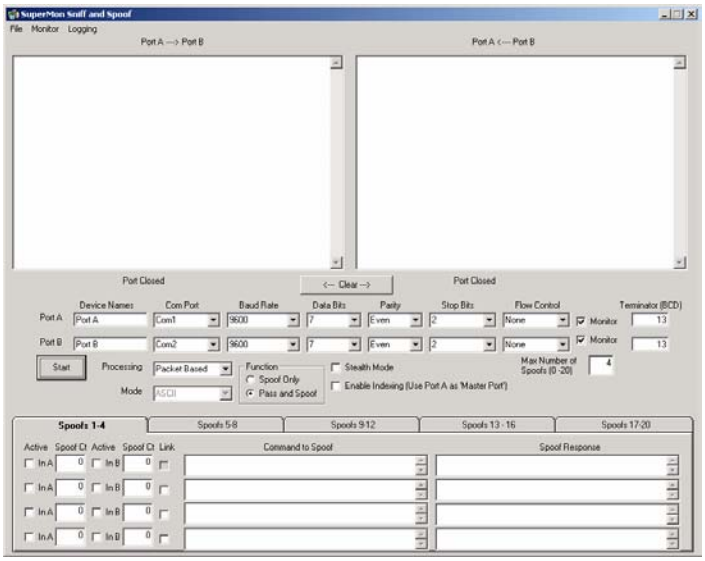
# Section 1. SuperMon Overview:

SuperMon is comprised of the following 3 functions:

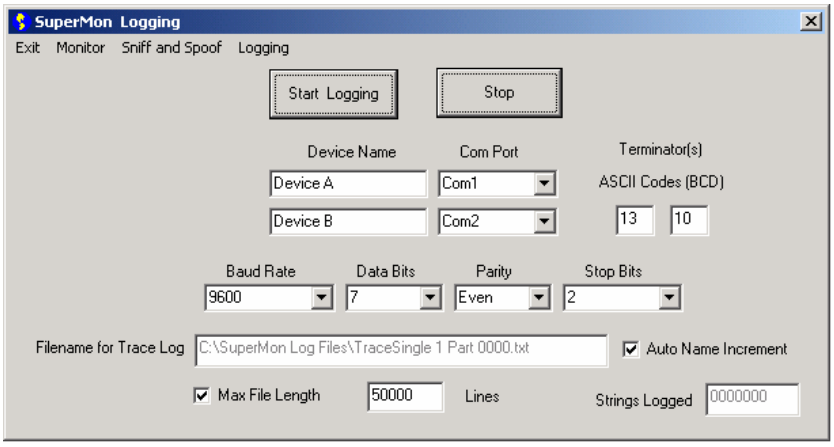
1. **Monitor**: This is a 1 or 2 port serial monitoring application. It taps into the serial data flow via a special cable (see section 6) if 2 devices are used, or it functions as a terminal emulator if 1 device is used.



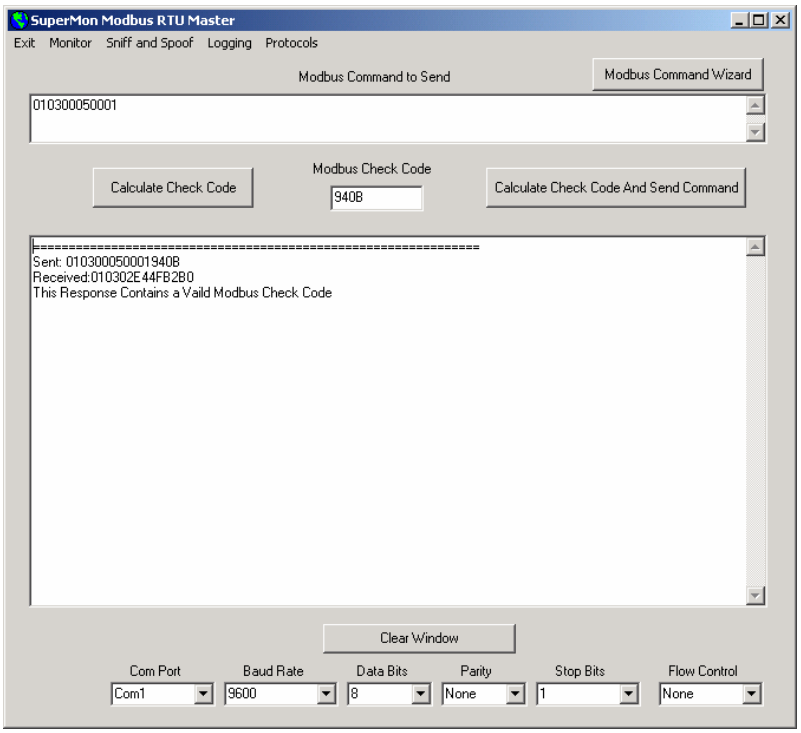

2. **Sniff and Spoof:** Sniff and Spoof is a utility intended to aid technical support and integration professionals in connecting 2 devices together for serial communications. This software is a 'pass through' application. This means that the computer running Sniff and Spoof passes data from 1 serial port to another, allowing 2 devices to communicate through the computer. This allows the user to monitor the serial traffic using the cables used to program the 2 devices, and test communications without having to build a custom cable between the 2 devices. The application also allows the user to specify certain serial commands to which a 'spoof' response should be created. In the case of a spoof, the serial command is intercepted by Sniff and Spoof, and a preprogrammed response is generated by Sniff and Spoof.



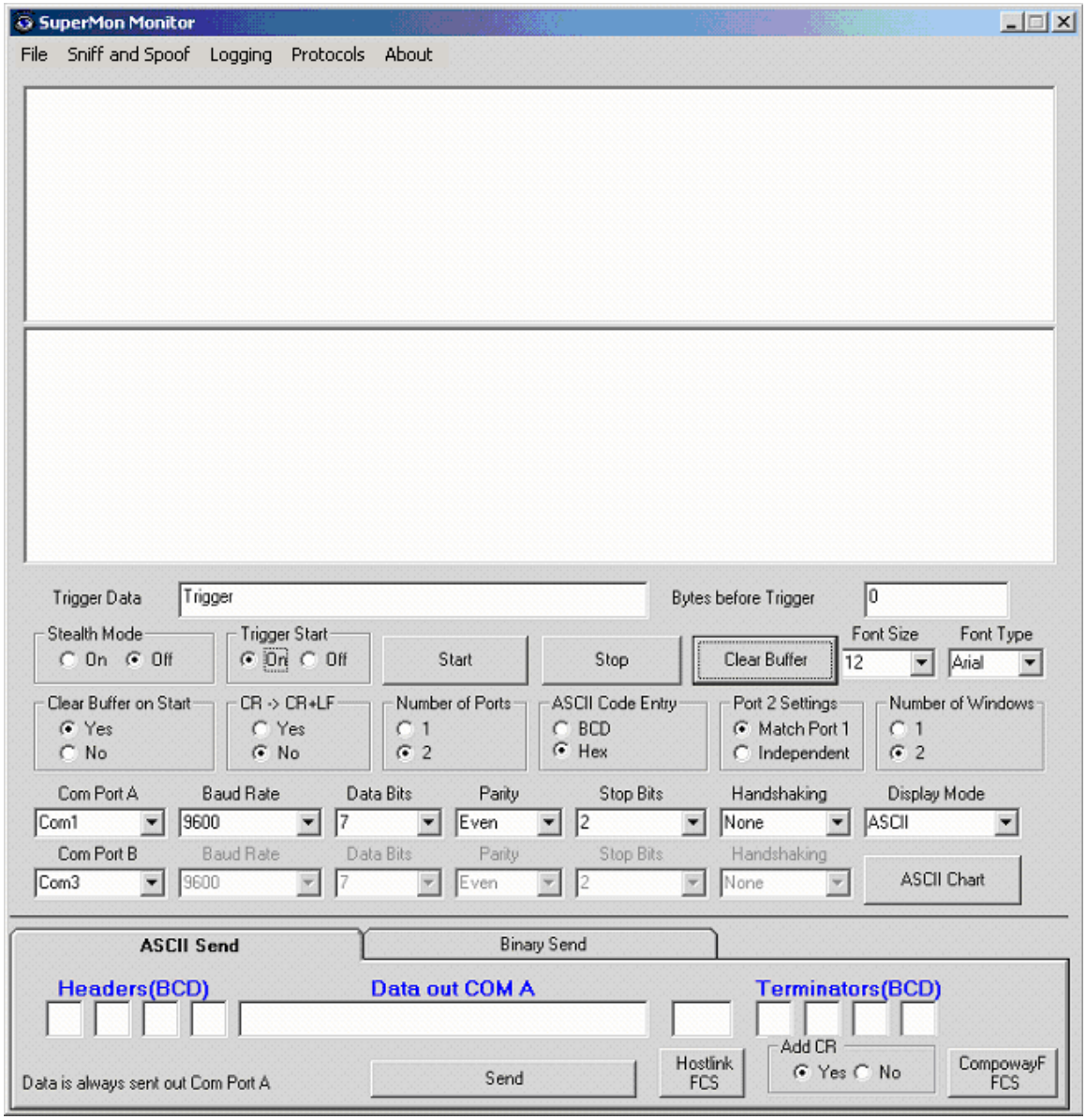

3. **Logging:** This application will log all serial activity from 1 or 2 ports to a text or CSV file along with a time and date stamp. This is intended to record long periods of serial activity.



4. **Protocol Specific Support:** Specific support is given for Omron HostLink, Omron CompoWay/F and Modbus. These capabilities include automatic header, terminator, and check code support, where applicable.



# Section 2. SuperMon Monitor Configuration + Operation




note: not all configuration options shown will be available simultaneously.

**Configuration of SuperMon Monitor:**

**Clear Buffer on Start**: Select whether or not to clear the display window(s) when the start button is pressed.

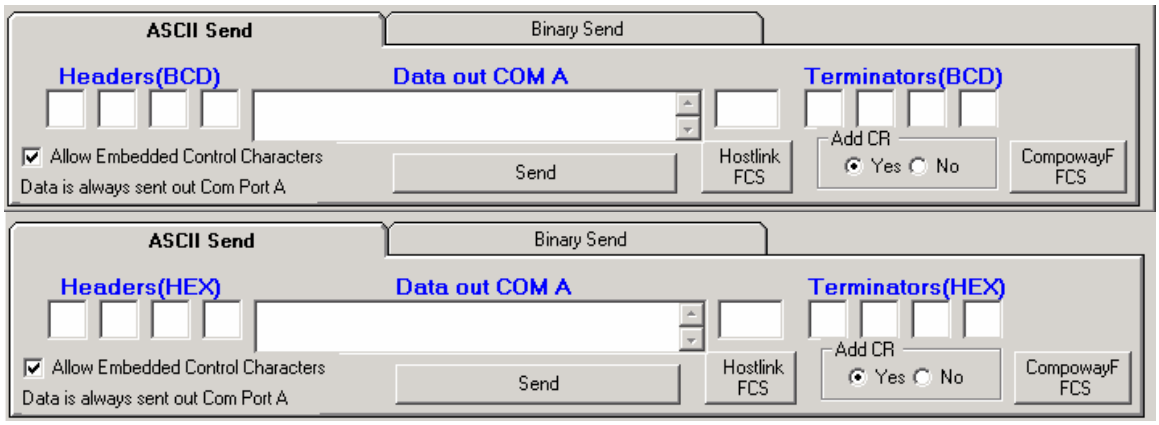
**CR -> CR+ LF**: Select whether or not incoming Carriage Returns are turned into Carriage Returns + Line Feeds for display purposes.

**Number of Ports**: Selects whether 1 or 2 serial ports are used for monitoring.

If 2 ports are selected, choose whether 1 or 2 monitoring windows will be used

If 2 windows are used, Port A data appears in the upper window, and Port B appears in the lower
If 1 window is used, the data from port A and B appear in 1 window. When the Stop button is pressed, the data is color coded to show which port received the information
1 window is best used when the device being monitored uses a command and response format.

**ASCII Code Entry**: Selects whether the ASCII headers and terminators in Send data are entered in BCD or HEX (ie 13 vs 0D for a carriage return).




**Port 2 Settings**: Selects whether the Com B port has baud rate, parity, data bits, and stop bits that match Com A, or if they are set independent of Com A.

**Number of Windows**: Selects whether 1 or 2 display windows are used. This selection is only enabled if 2 comm ports are used.

**Display Mode:** Selects Binary or ASCII mode for monitoring. Binary mode shows the raw hex values received. A '1' would appear as a 1 in ASCII mode, or a [31] in Binary mode.

**Stealth Mode:** When using 2 port 1 window mode, Stealth Mode can be used to ensure a more accurate command / response format of the display. The program does not waste time updating the screen during data capture in stealth mode.

**Trigger Start**: This is a mode that enables SuperMon to start recording data after a certain 'trigger' sequence has been received. The Trigger Data field can be filled out with the specific string to act as a trigger. To enter characters in the Binary format, use [xx] [xx] [xx] as the format. There is a space between each set of brackets. Example: to trigger on a CRLF use [0D] [0A] as the trigger start string.

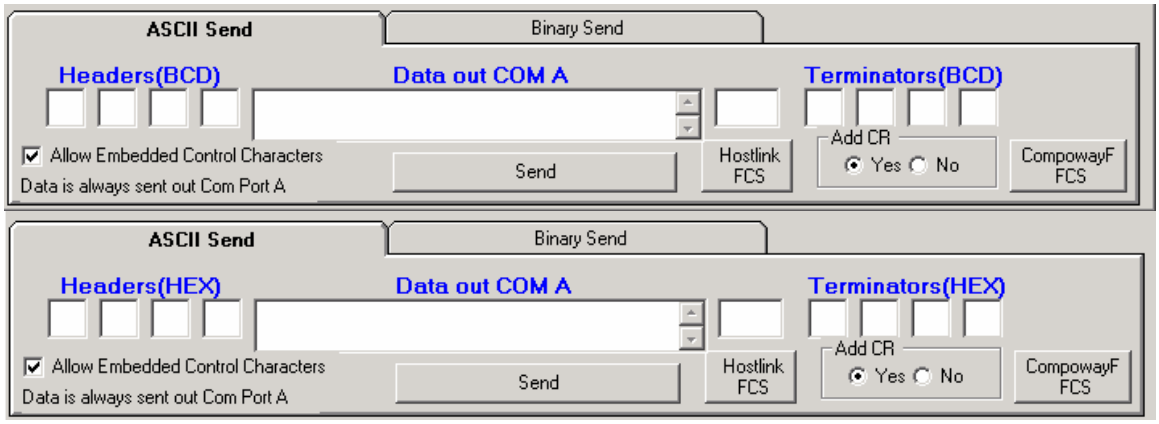
**Bytes before Trigger**: This field allows the user to specify how many bytes prior to the serial trigger string are recorded in the monitor window.

**Operation of SuperMon Monitor:**

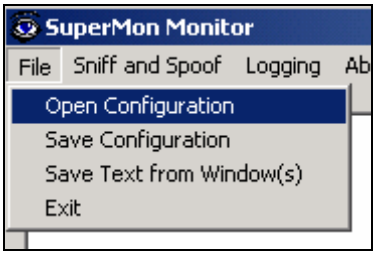
When all settings have been made, press the **Start** key to start monitoring data
Press the **Stop** button when finished.

During monitoring, data can be sent out Com A using the ASCII or Binary Send data tabs. Headers and Terminators can be specified, and a Hostlink or Compoway/F checksum can also be added in the ASCII Send mode.

**Allow Embedded Control Characters**: This allows users to insert non printable control characters into the ASCII text message. The format used is "[003]", ie square brackets around the control code number. If the ASCII Code Entry setting is set for HEX, then enter the control characters in HEX. If the ASCII Code Entry setting is set for BCD, enter the control characters in BCD. For example if the Data out COM A was set for "THIS IS A TEST[00D]. Did [003] it work? [002]", then the control characters 0D, 03, and 02 would be inserted as shown in the string.

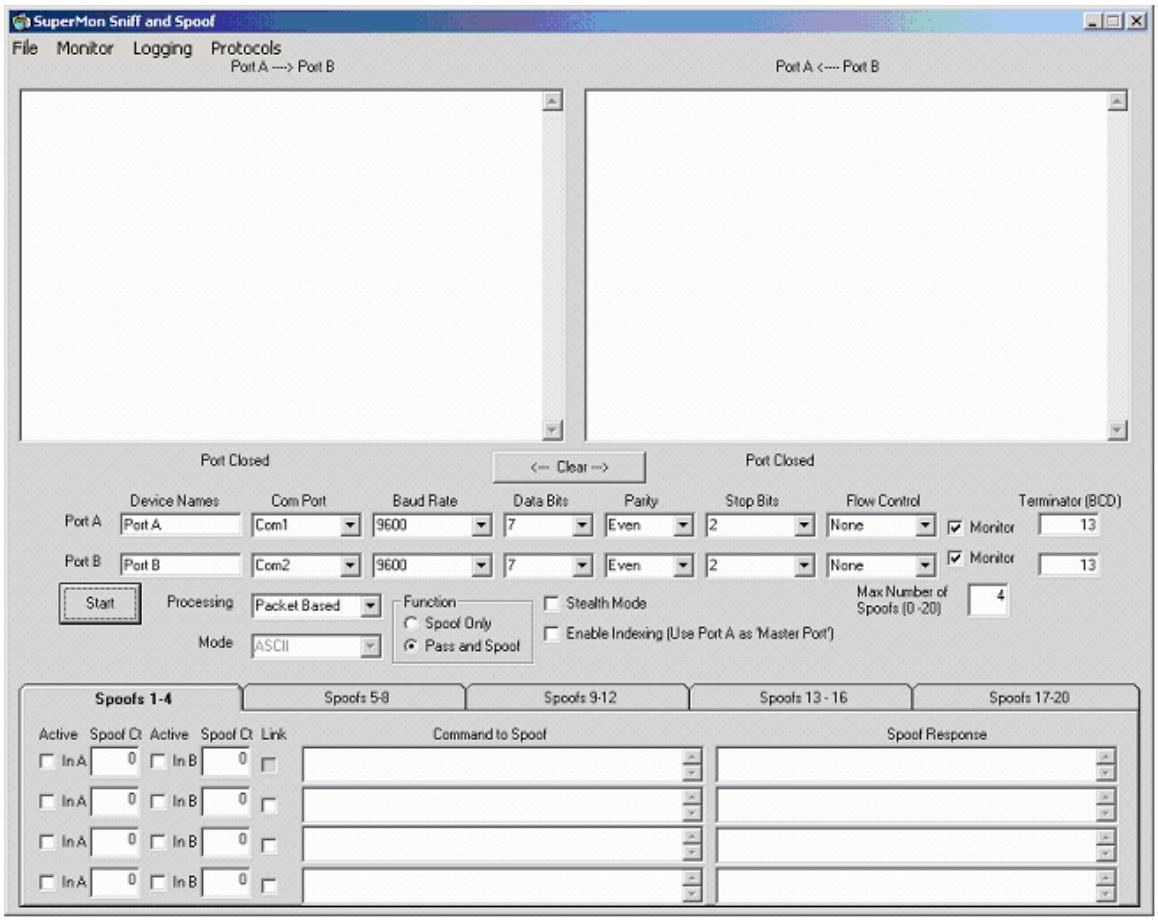



The configuration of the Monitor can be saved or loaded from disk. The text from the window(s) can also be saved. These functions are accessed from the File menu. The file is saved with a .smn file extension.




As a general practice, as SuperMon is using as much of the computers CPU resource as possible, avoid using other application when using SuperMon. This will make the data recording and display more accurate.

# Section 3. SuperMon Sniff and Spoof




note: not all configuration options shown will be available simultaneously.

**Configuration of SuperMon Sniff and Spoof:**

**Processing:**

a. **Processing**: **Packet:** This mode monitors and passes complete serial commands and responses, as determined by the specified terminator. Spoofs can only be used in Packet Mode.
b. **Byte:** This mode monitors and passes every byte of data. This mode is faster, but spoofs cannot be used.

**Mode**: This determines if ASCII or Binary displays are used for Port A and Port B. ASCII displays the ASCII characters ('ABC'). Binary displays the hexadecimal ASCII codes ('<41><42><43>'). Binary Mode can only be used with Packet processing mode.

**Function:**

a. **Spoof Only:** This is a single port operation mode. All responses are created using spoofs. This can be used in a limited fashion to simulate a hardware device.
b. **Pass and Spoof:** This is normal mode of operation in which data is passed back and forth between the 2 ports, and spoofs are used when indicated by the spoof settings.

**Stealth Mode**: This is an option to improve throughput of Sniff and Spoof. The data is not displayed in Port A and Port B display windows until the Stop button is pressed.

**Monitor**: Selects whether the port data is displayed in the monitor window. For faster throughput when monitoring is not necessary, uncheck the Monitor function for both ports.

**Terminators:** This is the BCD value of the terminator character for each of the 2 ports. This is used in Packet Based mode to identify the end of the serial string.

**Max. Number of Spoofs**: This is the maximum number of spoofs (0 – 20) that are checked by the software. Fewer spoofs improves throughput of the software.

**Spoof Settings:**

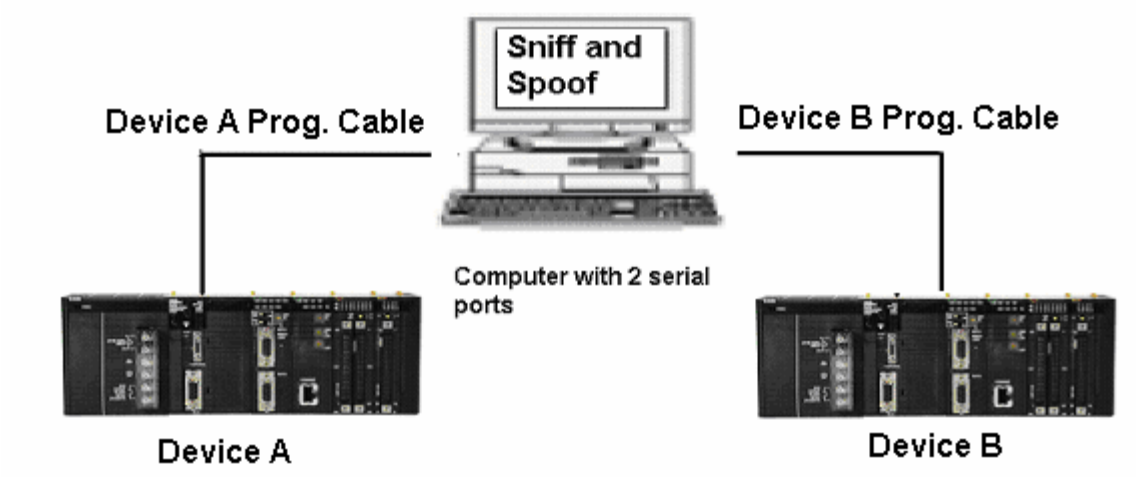
**Active:** Use the 'In A' and 'In B' check boxes to indicate for each spoof command and response pair whether or not the spoof is for data coming in Port A and / or Port B. For example: if the spoof is only to be used to answer commands coming in Port B, only check 'In B'.

**Link:** This is a special case for spoof operation. Checking the 'Link' box for a spoof indicates that this spoof will only occur when the command immediately follows the previous spoof. For example: if the spoof box is checked for spoof 3, the spoof response will only be activated if spoof 2 was the previous command.

**Command to Spoof:** This is the serial string to look for to issue the spoof response. This can be entered as a complete string, or as the start of a string, followed by a '*' for a wildcard character. For example: Command1234 or Comm* would both cause the application to respond with the spoof for the command.

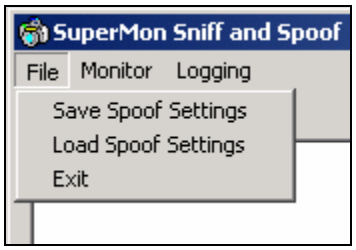
**Spoof Response:** This is the response to issue to the command when a spoof is being used.

**Operation of SuperMon Sniff and Spoof:**



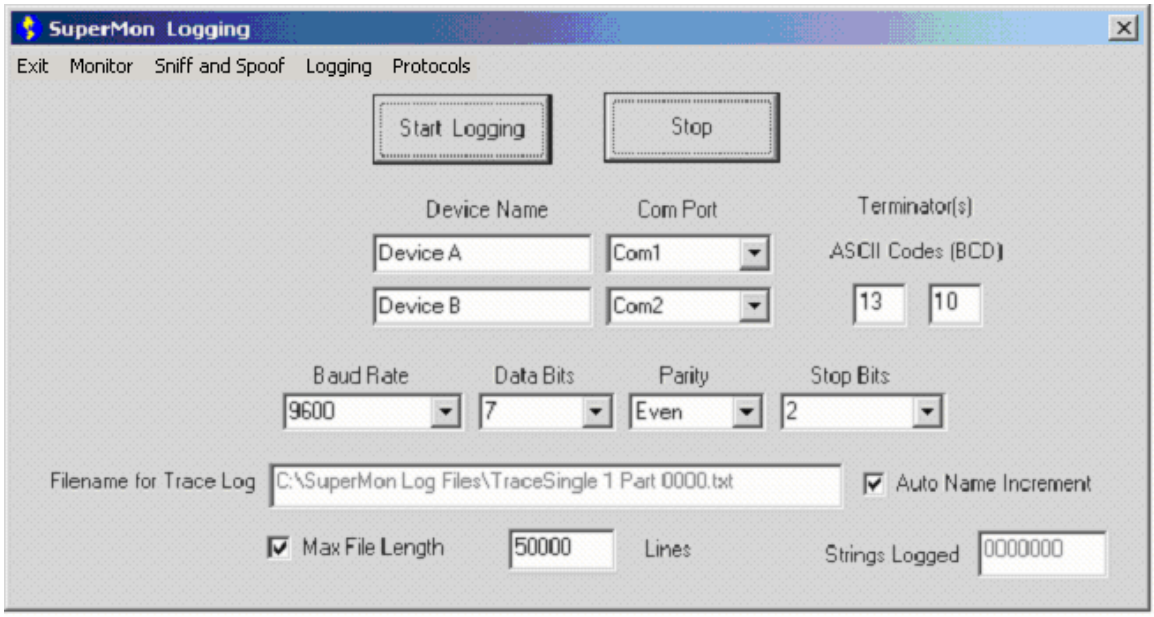

Connect the cables to each of the 2 devices (or to the single device if only 1 port is being used.)
Configure Sniff and Spoof and press the start key.
When finished, press the Stop Key

The configuration of the spoof settings can be saved or loaded from disk. These features are accessed through the File menu. The file is saved with a .spf file extension.




As a general practice, as SuperMon is using as much of the computers CPU resource as possible, avoid using other application when using SuperMon. This will make the data recording and display more accurate.

# Section 4. SuperMon Logging Configuration + Operation




note: not all configuration options shown will be available simultaneously.

**Configuration of SuperMon Logging:**

**Device Name:** This allows the user to add a name to the devices attached to the ports for use in the log file created.

**Terminators(s):** Terminators are used in Uni-Directional mode to determine the end of each individual string. If as single character (13 for example) is used, either box can be filled in, and the other left blank.

**Max File Length:** Specifies how many lines will be logged per file. After this number has been reached, a new file will be created.

**Auto Name Increment**: This allows the application to automatically increment the filename each time a new trace is started. The format is 'C:\SuperMon Log Files\TraceSingle ' xxxx 'Part ' yyyy '.txt'. xxxx is incremented each time a new trace is started with the 'Start' key. Yyyy is incremented each time the max file length is exceeded and a new file is started.

**Filename for Trace Log:** When Auto Name Increment is not selected, the user can enter a filename for the log file created.

**Operation of SuperMon Logging:**

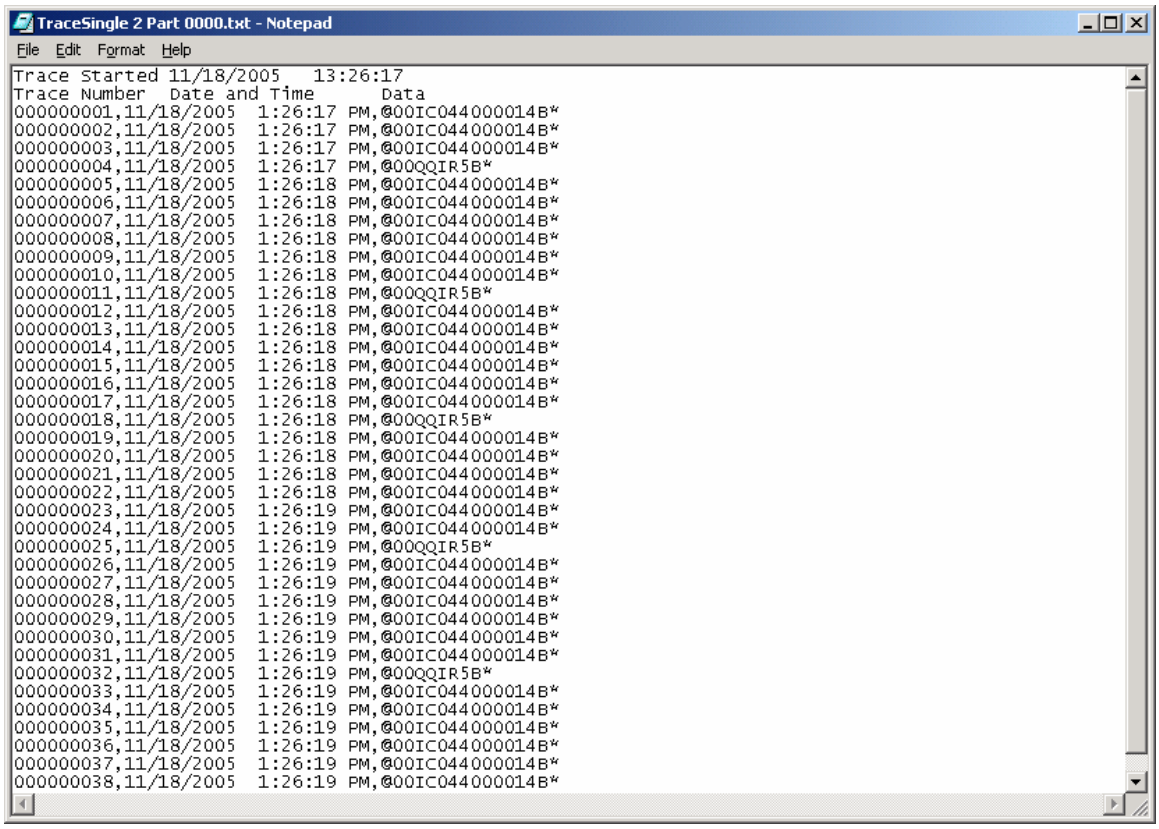
When all settings have been made, press the **Start** key to start logging data

Press the **Stop** button when finished.

As a general practice, as SuperMon is using as much of the computers CPU resource as possible, avoid using other application when using SuperMon. This will make the data recording and display more accurate.

# Section 5: File format of Logging files

**Uni Directional (one port)**
The file format is comma deliminted (can be renamed as a .csv file for Excel).
The fields are (in order)
1. Trace number. This is an incrementing number used as a reference pointer only.
2. Real Time Clock time and date.
3. Data shows the actual data sent.


TraceSingle 2 Part 0000.txt - Notepad

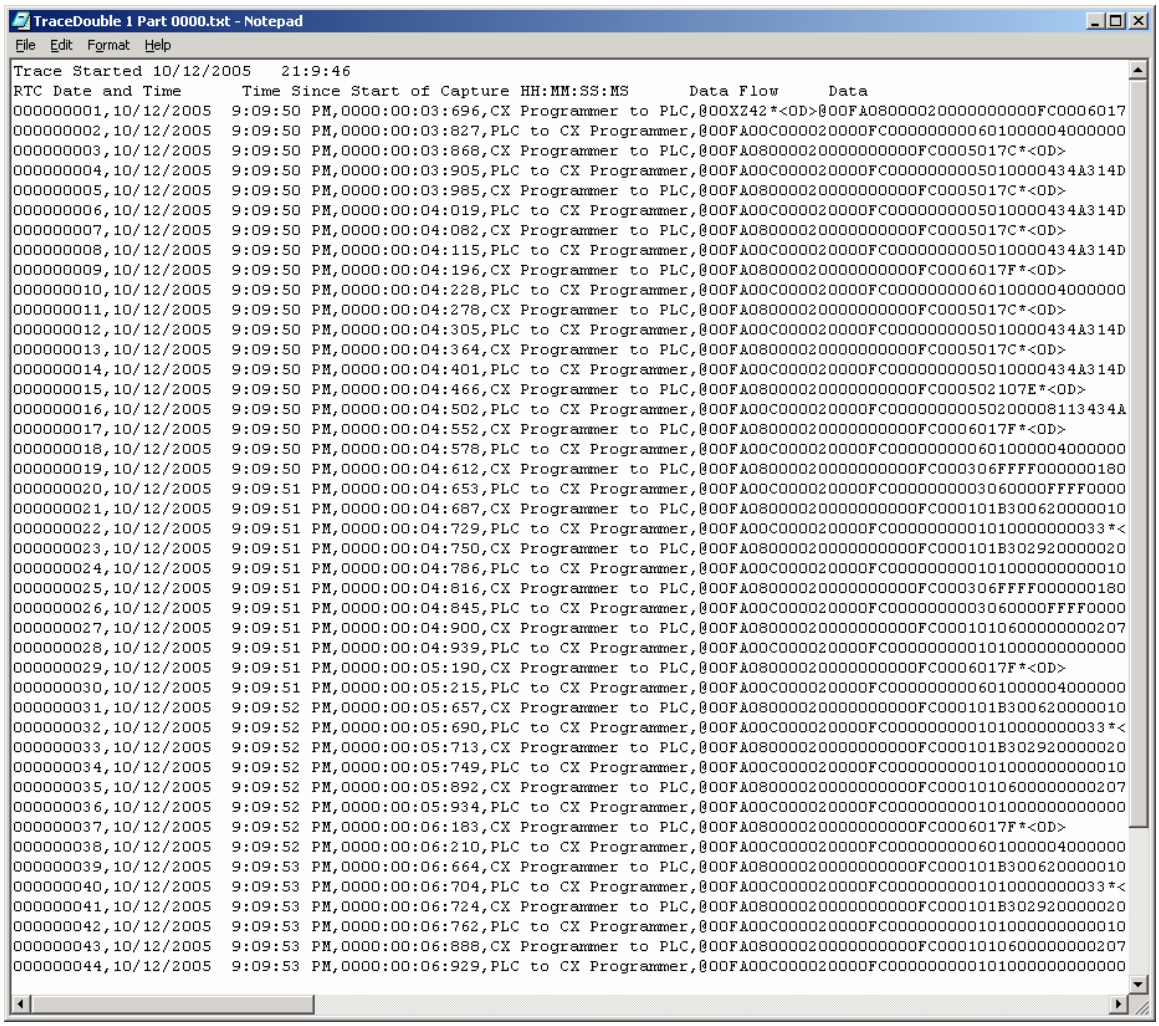
File Edit Format Help

```
Trace Started 11/18/2005   13:26:17
Trace Number  Date and Time      Data
000000001,11/18/2005  1:26:17 PM,@00IC044000014B*
000000002,11/18/2005  1:26:17 PM,@00IC044000014B*
000000003,11/18/2005  1:26:17 PM,@00IC044000014B*
000000004,11/18/2005  1:26:17 PM,@00QQIR5B*
000000005,11/18/2005  1:26:18 PM,@00IC044000014B*
000000006,11/18/2005  1:26:18 PM,@00IC044000014B*
000000007,11/18/2005  1:26:18 PM,@00IC044000014B*
000000008,11/18/2005  1:26:18 PM,@00IC044000014B*
000000009,11/18/2005  1:26:18 PM,@00IC044000014B*
000000010,11/18/2005  1:26:18 PM,@00IC044000014B*
000000011,11/18/2005  1:26:18 PM,@00QQIR5B*
000000012,11/18/2005  1:26:18 PM,@00IC044000014B*
000000013,11/18/2005  1:26:18 PM,@00IC044000014B*
000000014,11/18/2005  1:26:18 PM,@00IC044000014B*
000000015,11/18/2005  1:26:18 PM,@00IC044000014B*
000000016,11/18/2005  1:26:18 PM,@00IC044000014B*
000000017,11/18/2005  1:26:18 PM,@00IC044000014B*
000000018,11/18/2005  1:26:18 PM,@00QQIR5B*
000000019,11/18/2005  1:26:18 PM,@00IC044000014B*
000000020,11/18/2005  1:26:18 PM,@00IC044000014B*
000000021,11/18/2005  1:26:18 PM,@00IC044000014B*
000000022,11/18/2005  1:26:18 PM,@00IC044000014B*
000000023,11/18/2005  1:26:19 PM,@00IC044000014B*
000000024,11/18/2005  1:26:19 PM,@00IC044000014B*
000000025,11/18/2005  1:26:19 PM,@00QQIR5B*
000000026,11/18/2005  1:26:19 PM,@00IC044000014B*
000000027,11/18/2005  1:26:19 PM,@00IC044000014B*
000000028,11/18/2005  1:26:19 PM,@00IC044000014B*
000000029,11/18/2005  1:26:19 PM,@00IC044000014B*
000000030,11/18/2005  1:26:19 PM,@00IC044000014B*
000000031,11/18/2005  1:26:19 PM,@00IC044000014B*
000000032,11/18/2005  1:26:19 PM,@00QQIR5B*
000000033,11/18/2005  1:26:19 PM,@00IC044000014B*
000000034,11/18/2005  1:26:19 PM,@00IC044000014B*
000000035,11/18/2005  1:26:19 PM,@00IC044000014B*
000000036,11/18/2005  1:26:19 PM,@00IC044000014B*
000000037,11/18/2005  1:26:19 PM,@00IC044000014B*
000000038,11/18/2005  1:26:19 PM,@00IC044000014B*
```

**Bi Directional (two ports)**

The file format is comma deliminted (can be renamed as a .csv file for Excel).
The fields are (in order)

1. Trace number. This is an incrementing number used as a reference pointer only.
2. Real Time Clock time and date, accurate to the second.
3. Time since start of capture is a 1ms resolution timer to indicate the command / response timing between devices.
4. Data Flow indicates which direction the data is flowing based on the Device Names in the logging application.
5. Data shows the actual data sent.



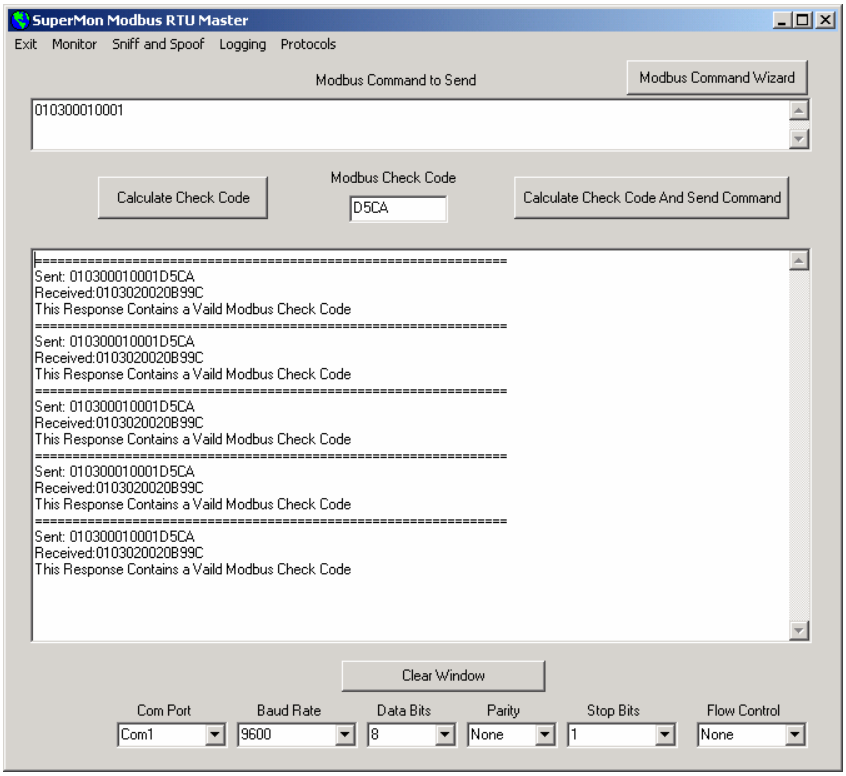
TraceDouble 1 Part 0000.txt - Notepad

File Edit Format Help

```
Trace Started 10/12/2005   21:9:46
RTC Date and Time      Time Since Start of Capture HH:MM:SS:MS      Data Flow     Data
000000001,10/12/2005  9:09:50 PM,0000:00:03:696,CX Programmer to PLC,@00XZ42*<0D>@00FA0800002000000000FC0006017
000000002,10/12/2005  9:09:50 PM,0000:00:03:827,PLC to CX Programmer,@00FA00C000020000FC000000006010000040000000
000000003,10/12/2005  9:09:50 PM,0000:00:03:868,CX Programmer to PLC,@00FA0800002000000000FC0005017C*<0D>
000000004,10/12/2005  9:09:50 PM,0000:00:03:905,PLC to CX Programmer,@00FA00C000020000FC000000005010000434A314D
000000005,10/12/2005  9:09:50 PM,0000:00:03:985,CX Programmer to PLC,@00FA0800002000000000FC0005017C*<0D>
000000006,10/12/2005  9:09:50 PM,0000:00:04:019,PLC to CX Programmer,@00FA00C000020000FC000000005010000434A314D
000000007,10/12/2005  9:09:50 PM,0000:00:04:082,CX Programmer to PLC,@00FA0800002000000000FC0005017C*<0D>
000000008,10/12/2005  9:09:50 PM,0000:00:04:115,PLC to CX Programmer,@00FA00C000020000FC000000005010000434A314D
000000009,10/12/2005  9:09:50 PM,0000:00:04:196,CX Programmer to PLC,@00FA0800002000000000FC0006017F*<0D>
000000010,10/12/2005  9:09:50 PM,0000:00:04:228,PLC to CX Programmer,@00FA00C000020000FC000000006010000040000000
000000011,10/12/2005  9:09:50 PM,0000:00:04:278,CX Programmer to PLC,@00FA0800002000000000FC0005017C*<0D>
000000012,10/12/2005  9:09:50 PM,0000:00:04:305,PLC to CX Programmer,@00FA00C000020000FC000000005010000434A314D
000000013,10/12/2005  9:09:50 PM,0000:00:04:364,CX Programmer to PLC,@00FA0800002000000000FC0005017C*<0D>
000000014,10/12/2005  9:09:50 PM,0000:00:04:401,PLC to CX Programmer,@00FA00C000020000FC000000005010000434A314D
000000015,10/12/2005  9:09:50 PM,0000:00:04:466,CX Programmer to PLC,@00FA0800002000000000FC000502107E*<0D>
000000016,10/12/2005  9:09:50 PM,0000:00:04:502,PLC to CX Programmer,@00FA00C000020000FC000000005020000B113434A
000000017,10/12/2005  9:09:50 PM,0000:00:04:552,CX Programmer to PLC,@00FA0800002000000000FC0006017F*<0D>
000000018,10/12/2005  9:09:50 PM,0000:00:04:578,PLC to CX Programmer,@00FA00C000020000FC000000006010000040000000
000000019,10/12/2005  9:09:50 PM,0000:00:04:612,CX Programmer to PLC,@00FA0800002000000000FC000306FFFF000000180
000000020,10/12/2005  9:09:51 PM,0000:00:04:653,PLC to CX Programmer,@00FA00C000020000FC000000003060000FFFF0000
000000021,10/12/2005  9:09:51 PM,0000:00:04:687,CX Programmer to PLC,@00FA0800002000000000FC000101B300620000010
000000022,10/12/2005  9:09:51 PM,0000:00:04:729,PLC to CX Programmer,@00FA00C000020000FC000000001010000000033*<
000000023,10/12/2005  9:09:51 PM,0000:00:04:750,CX Programmer to PLC,@00FA0800002000000000FC000101B302920000020
000000024,10/12/2005  9:09:51 PM,0000:00:04:786,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000010
000000025,10/12/2005  9:09:51 PM,0000:00:04:816,CX Programmer to PLC,@00FA0800002000000000FC000306FFFF000000180
000000026,10/12/2005  9:09:51 PM,0000:00:04:845,PLC to CX Programmer,@00FA00C000020000FC000000003060000FFFF0000
000000027,10/12/2005  9:09:51 PM,0000:00:04:900,CX Programmer to PLC,@00FA0800002000000000FC000101060000000207
000000028,10/12/2005  9:09:51 PM,0000:00:04:939,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000000
000000029,10/12/2005  9:09:51 PM,0000:00:05:190,CX Programmer to PLC,@00FA0800002000000000FC0006017F*<0D>
000000030,10/12/2005  9:09:51 PM,0000:00:05:215,PLC to CX Programmer,@00FA00C000020000FC000000006010000040000000
000000031,10/12/2005  9:09:52 PM,0000:00:05:657,CX Programmer to PLC,@00FA0800002000000000FC000101B300620000010
000000032,10/12/2005  9:09:52 PM,0000:00:05:690,PLC to CX Programmer,@00FA00C000020000FC000000001010000000033*<
000000033,10/12/2005  9:09:52 PM,0000:00:05:713,CX Programmer to PLC,@00FA0800002000000000FC000101B302920000020
000000034,10/12/2005  9:09:52 PM,0000:00:05:749,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000010
000000035,10/12/2005  9:09:52 PM,0000:00:05:892,CX Programmer to PLC,@00FA0800002000000000FC000101060000000207
000000036,10/12/2005  9:09:52 PM,0000:00:05:934,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000000
000000037,10/12/2005  9:09:52 PM,0000:00:06:183,CX Programmer to PLC,@00FA0800002000000000FC0006017F*<0D>
000000038,10/12/2005  9:09:52 PM,0000:00:06:210,PLC to CX Programmer,@00FA00C000020000FC000000006010000040000000
000000039,10/12/2005  9:09:53 PM,0000:00:06:664,CX Programmer to PLC,@00FA0800002000000000FC000101B300620000010
000000040,10/12/2005  9:09:53 PM,0000:00:06:704,PLC to CX Programmer,@00FA00C000020000FC000000001010000000033*<
000000041,10/12/2005  9:09:53 PM,0000:00:06:724,CX Programmer to PLC,@00FA0800002000000000FC000101B302920000020
000000042,10/12/2005  9:09:53 PM,0000:00:06:762,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000010
000000043,10/12/2005  9:09:53 PM,0000:00:06:888,CX Programmer to PLC,@00FA0800002000000000FC000101060000000207
000000044,10/12/2005  9:09:53 PM,0000:00:06:929,PLC to CX Programmer,@00FA00C000020000FC000000001010000000000000
```

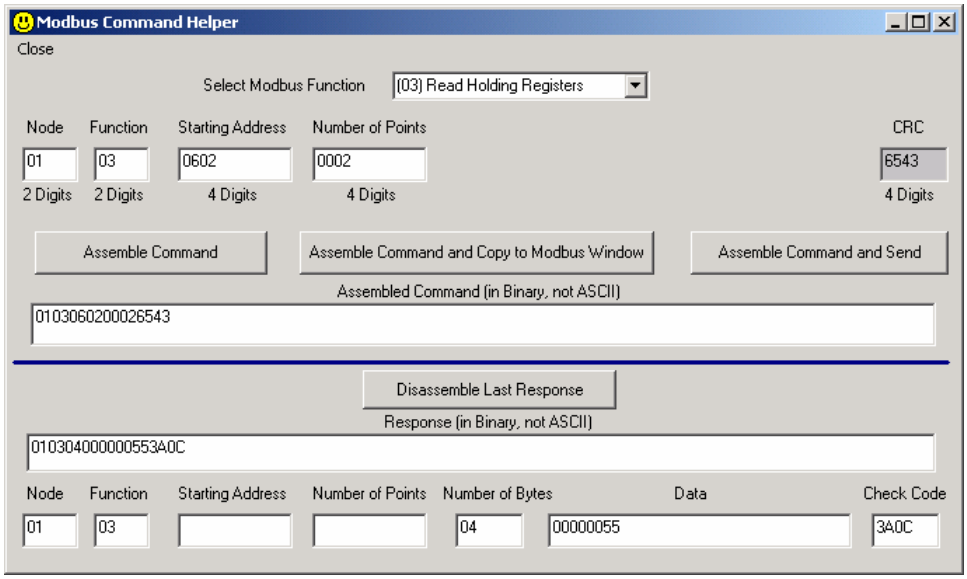
# Section 6. Configuration and Operation of Protocol Specific Screens

Modbus RTU Master:




This mode allows SuperMon to act as a master on a Modbus RTU network. Commands can be sent and the responses viewed.
All data should be entered as ASCII (ex 010300010001) even though Modbus RTU uses a binary protocol. SuperMon will send out the equivalent binary codes for the ASCII text entered.
SuperMon automatically calculates and appends to the command the Modbus RTU CRC16 check code.

The **Modbus Command Wizard** assists the user in building a Modbus command and understanding a response. This can be accessed from the main Modbus window via the **Modbus Command Wizard** button.



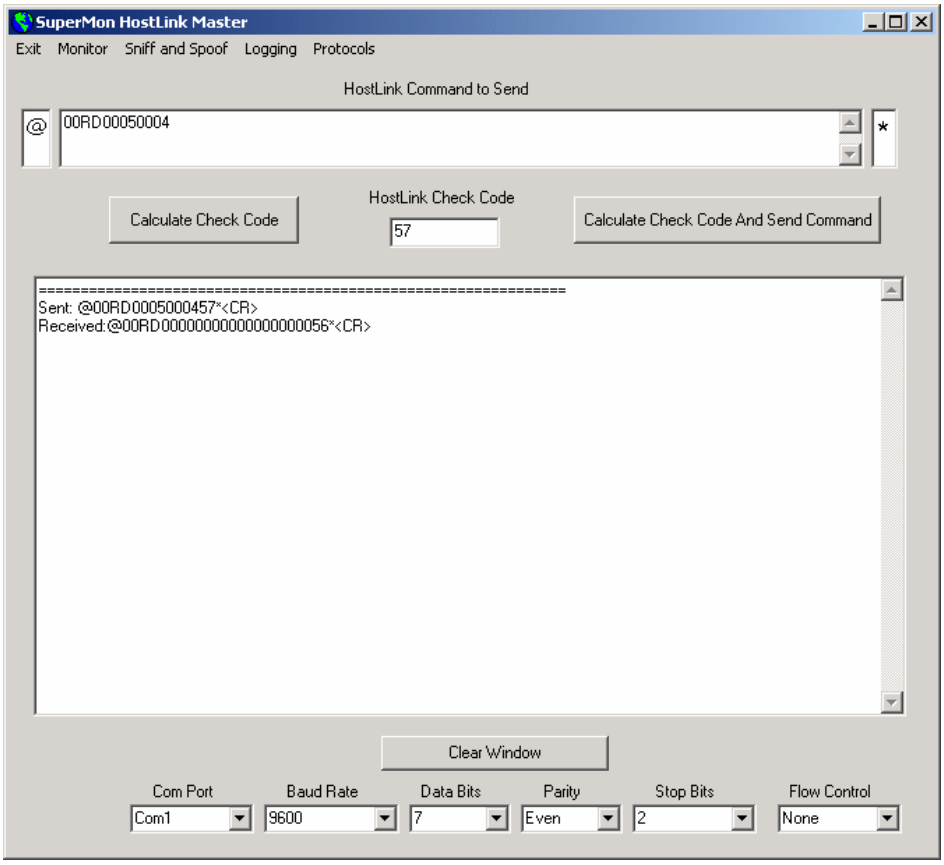

**Assemble Command:** This shows the assembled Modbus command, including the CRC.

**Assemble Command and Copy to Modbus Window:** This displays the assembled Modbus command, and copies it to the command window of the Modbus RTU Master screen.

**Assemble Command and Send:** This assembles the Modbus command, copies it to the Modbus RTU Master screen, and sends the command out the port. The response is then automatically disassembled in the disassembly area. This utilizes the communication settings (port, baud, parity, etc) set on the main Modbus RTU Master window.

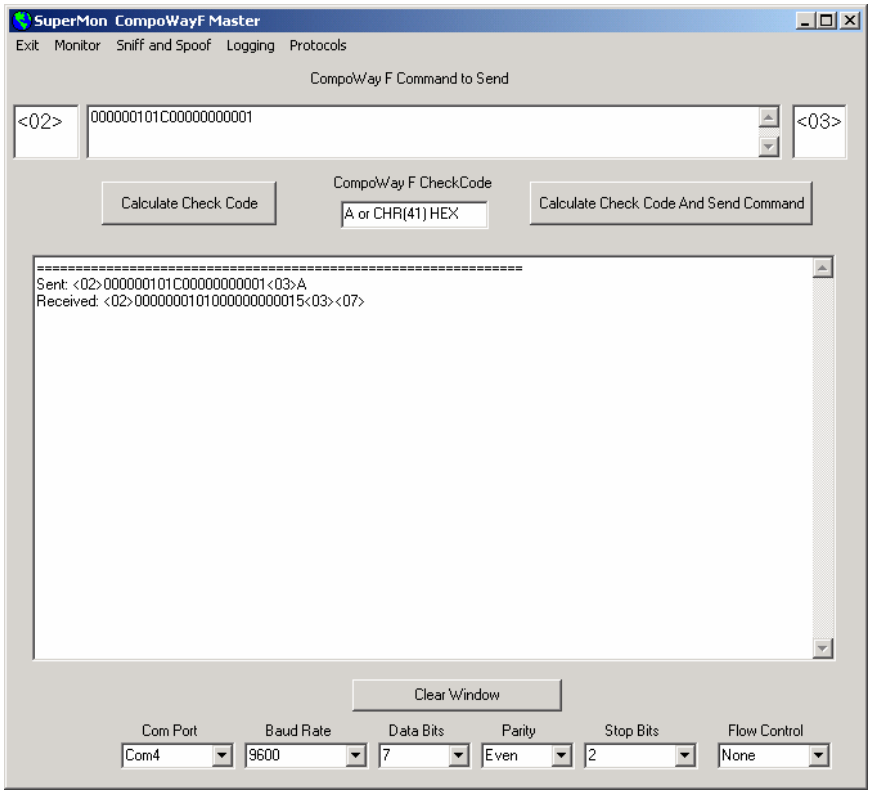
**Disassemble Last Response:** This take the last response from the Modbus RTU Master screen and disassembles it into its components. The command need not have been sent from the command wizard to disassemble it.

HostLink Master:




This mode allows SuperMon to act as a master on a HostLink network. Commands can be sent and the responses viewed.
All data should be entered as ASCII (ex 010300010001).
SuperMon automatically adds the Hostlink header "@' to the beginning of the command, and the FCS, "*" and <CR> at the end.

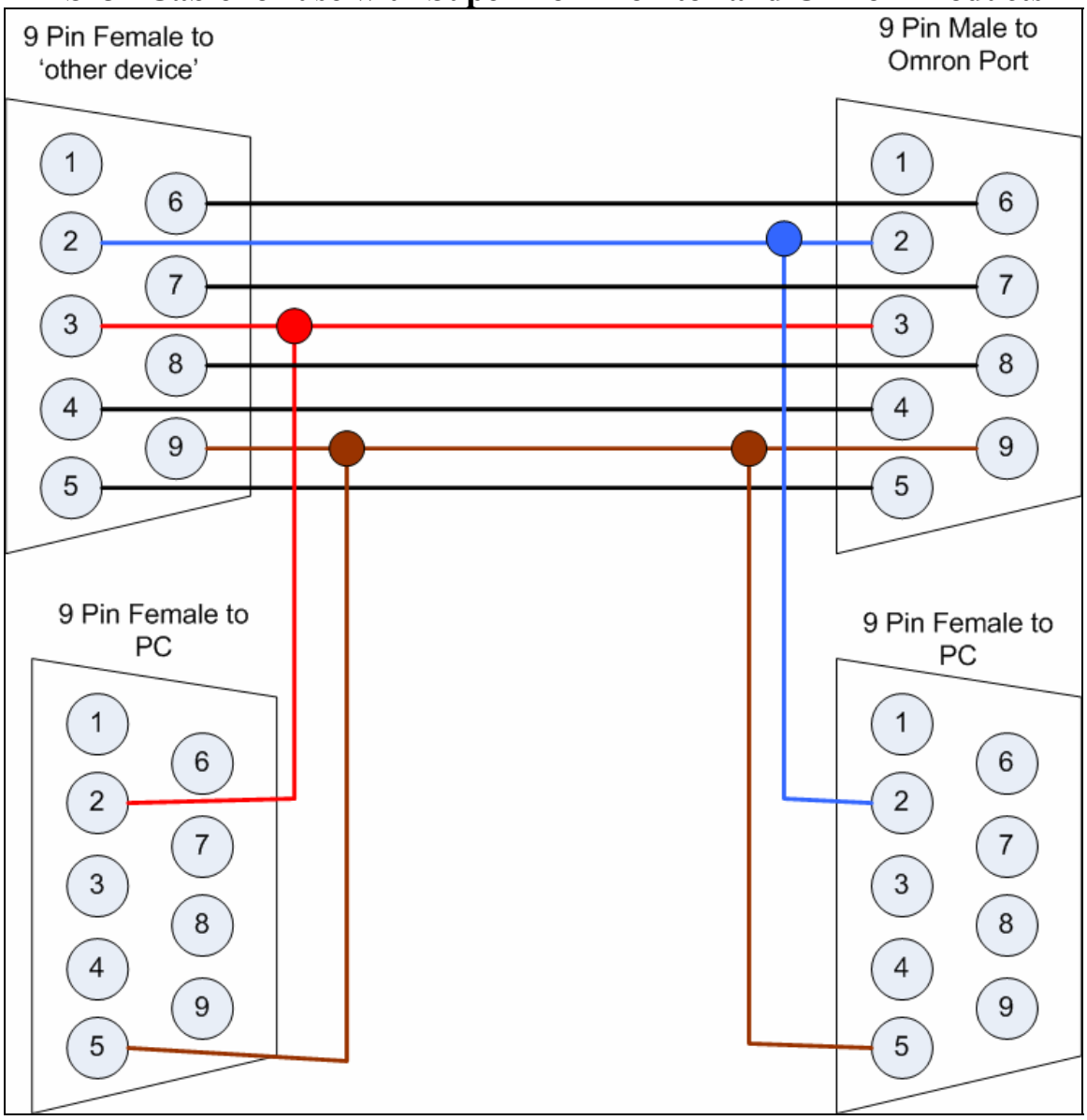
CompoWayF Master:



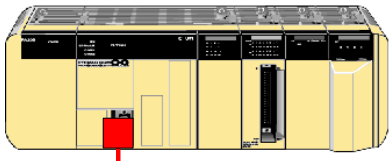

This mode allows SuperMon to act as a master on a CompWay/F network. Commands can be sent and the responses viewed.
All data should be entered as ASCII (ex 010300010001).
SuperMon automatically adds the CompoWay/F header <STX> to the beginning of the command, and the <ETX> and FCS at the end.

# Section 7. Cables for use with SuperMon

## RS232 Cable for use with SuperMon Monitor and Omron Products



Using your Home Made Omron Data Sniffing Cable



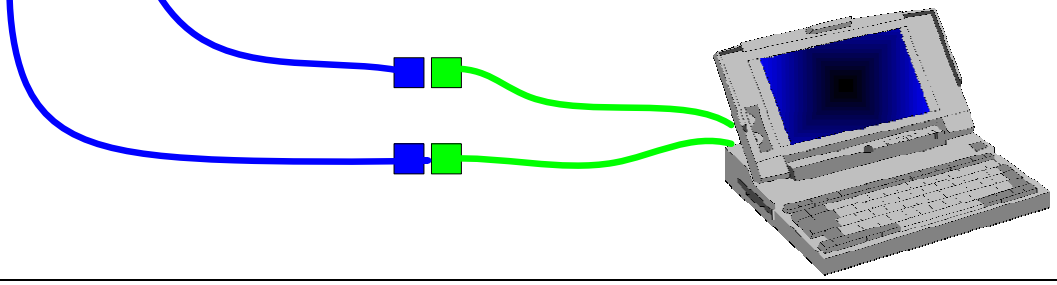
Normal Device Connection

3rd Party Device

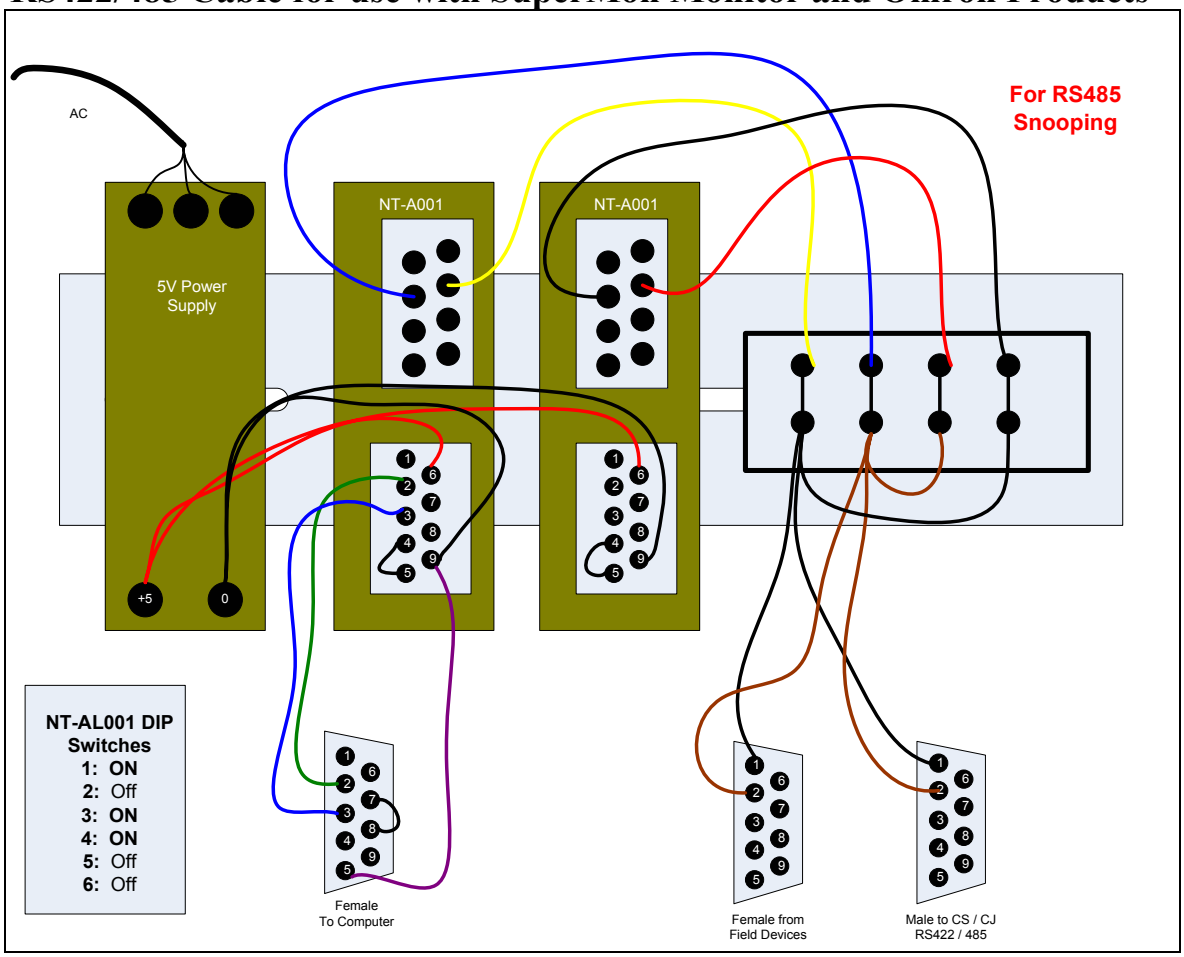


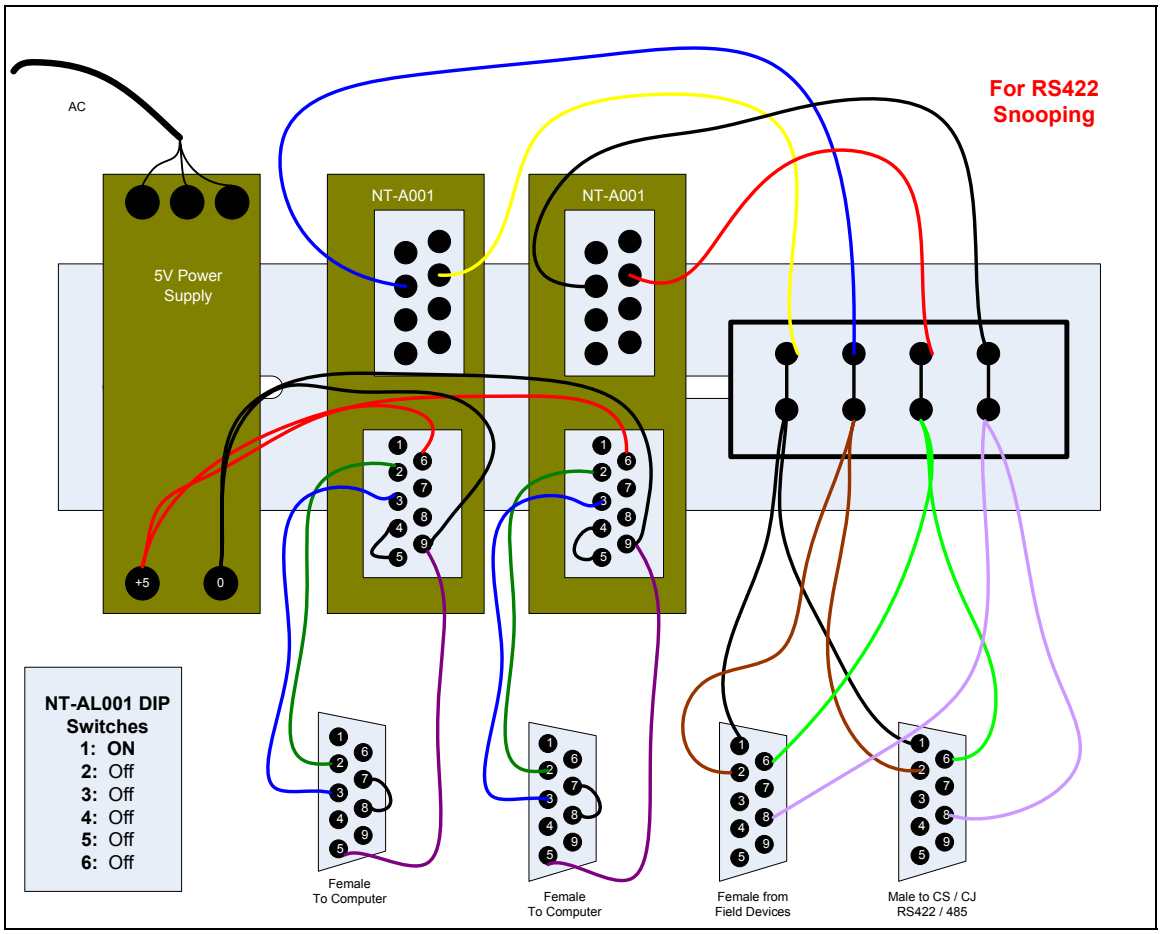
Connection With Data Sniffing Cable

3rd Party Device



# RS422/485 Cable for use with SuperMon Monitor and Omron Products




For RS422
Snooping
AC
5V Power
Supply
+5
0
NT-A001
NT-A001
NT-AL001 DIP
Switches
1: ON
2: Off
3: Off
4: Off
5: Off
6: Off
Female
To Computer
Female
To Computer
Female from
Field Devices
Male to CS / CJ
RS422 / 485

# Section 8: Revision History

5.0 Merging of SuperMon, Sniff and Spoof, and Data Logging
5.1 Added embedded control characters in ASCII send on main for of SuperMon
5.5 Added Protocol Specific Screens for Modbus RTU, HostLink, and CompoWay/F