Read Me

This older, Visual Basic 6 programming example was created using the Microsoft Visual Basic 6.

The program lets you define the serial port, baud rate and comm format. It will then take your command string and send it to the PLC and obtain a response string from the PLC and display on the response box. It takes care of time-out (user- definable) and will not block the program if the PLC is not present.

You can easily include this form into your project and display it for user to set the communication parameters as well as perform some quick test of the communication with the PLC. The full source code is provided for your convenience if you wish to make any modifications. **There are a 3 public functions defined in the sample file which you can call from your own application programs directly:**

1	**Public Function sendCommand (cmd As String)** You supply the entire command string in "cmd" and call this function to send out the command to the PLC. It will then wait to the return the response string received from the PLC or if time-out, it will return an empty string to the caller. If "cmd" is supplied without the "@" sign, it will treat the cmd as a point-to-point command and automatically handle the "Ctrl-E" protocol handshaking.
2	**Public Function AppendFCS (ID As Integer, cmd As String)** This function converts a point-to-point command (without the "*" terminator) into a multi-point one. It first adds the "@nn" to the "cmd" string where nn is the ASCII hex representation of the ID, the FCS for the resultant string is then computed and appended to the end of the "@nn"+cmd string, followed by the "*" character. E.g. command = AppendFCS(1, "RI00") The variable "command" will receive the resultant string "@01RI005A*".
3	**Public Function computeFCS (cmd As String)** This function computes the FCS of a given string "cmd". It is used by the AppendFCS function and is may also be used by the user to verify the integrity of a received response string.