



PRELIMINARY INFORMATION

R.E.Smith I/O Commander USER GUIDE

Serial Port Communication Utility

(Software Version Rescomdr-20071231.exe)

Document Date 12-31-2007

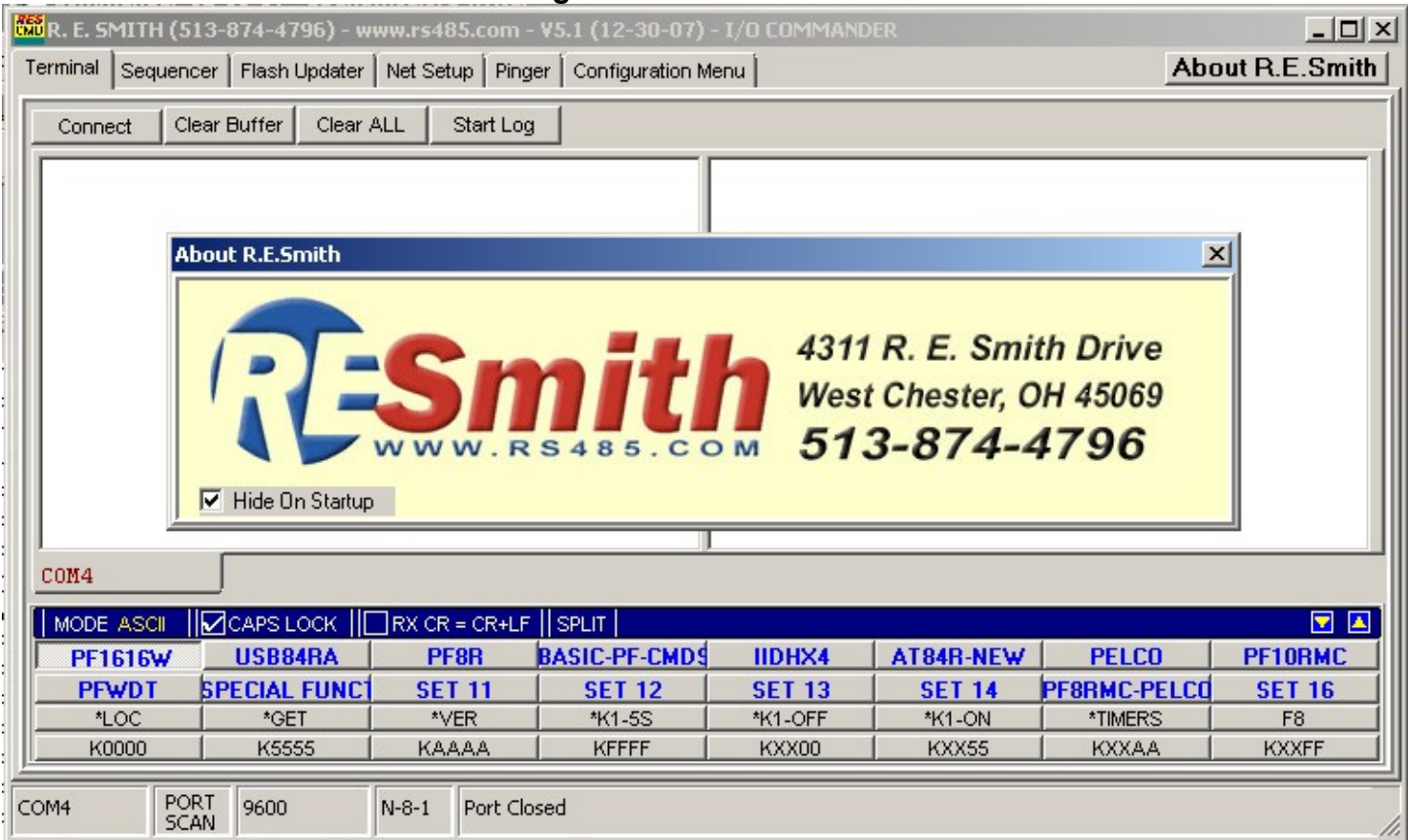
Table of Contents

INTRODUCTION.....	3
Figure 1. I/O Commander.....	3
Setup.....	4
Figure 2. Communications Parameters.....	4
Useful Functions.....	5
Figure 3. I/O Commander Configuration Menu Tab.....	5
Figure 4. I/O Commander Terminal.....	6
Sequencer Commands Configuration Commands.....	7
Additional Sequencer Commands.....	8

INTRODUCTION

The R.E.Smith I/O Commander is a multifunction communication Microsoft Windows™ compatible utility that enables the user to control and configure R.E.Smith Communications and I/O products. Many functions of the utility will work with most serial port devices manufactured by other sources.

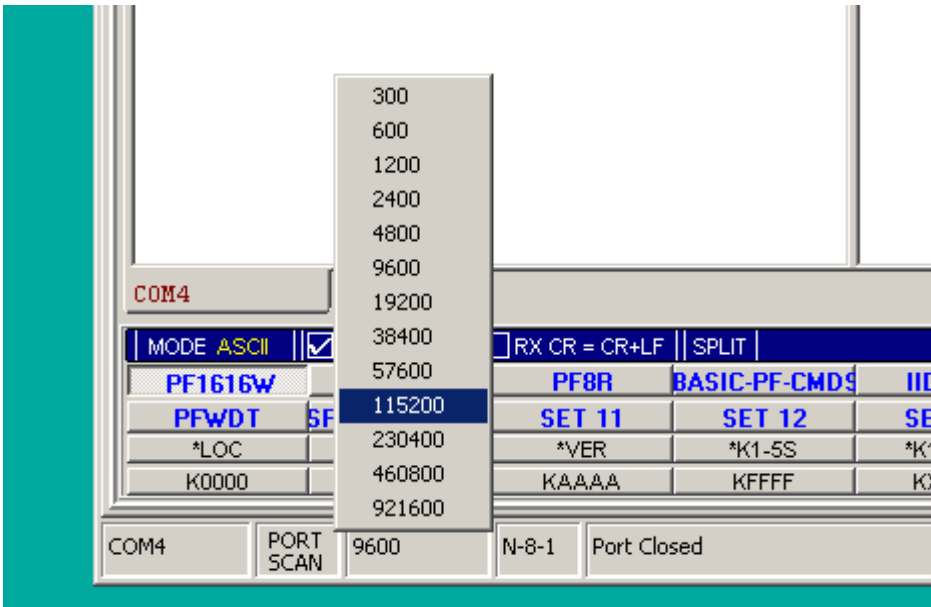
Figure 1. I/O Commander



Setup

Setting the communication settings for working with RS232 devices is easy. Clicking the areas on the bottom of the application will bring up pop-up menus for the communication settings for communications port, baud rate, and communications specification.

Figure 2. Communications Parameters



Useful Functions

The functions of the I/O Commander are grouped into a tab organization. The two main functions are the Terminal and the Sequencer these will be discussed in detail below. Many settings are modifiable on the Configuration Menu tab, for example by default the text color for transmitted data is red and data received is shown in green. The user can change these defaults on the "Program Setup" tab of the main Configuration Menu .

Figure 3. I/O Commander Configuration Menu Tab

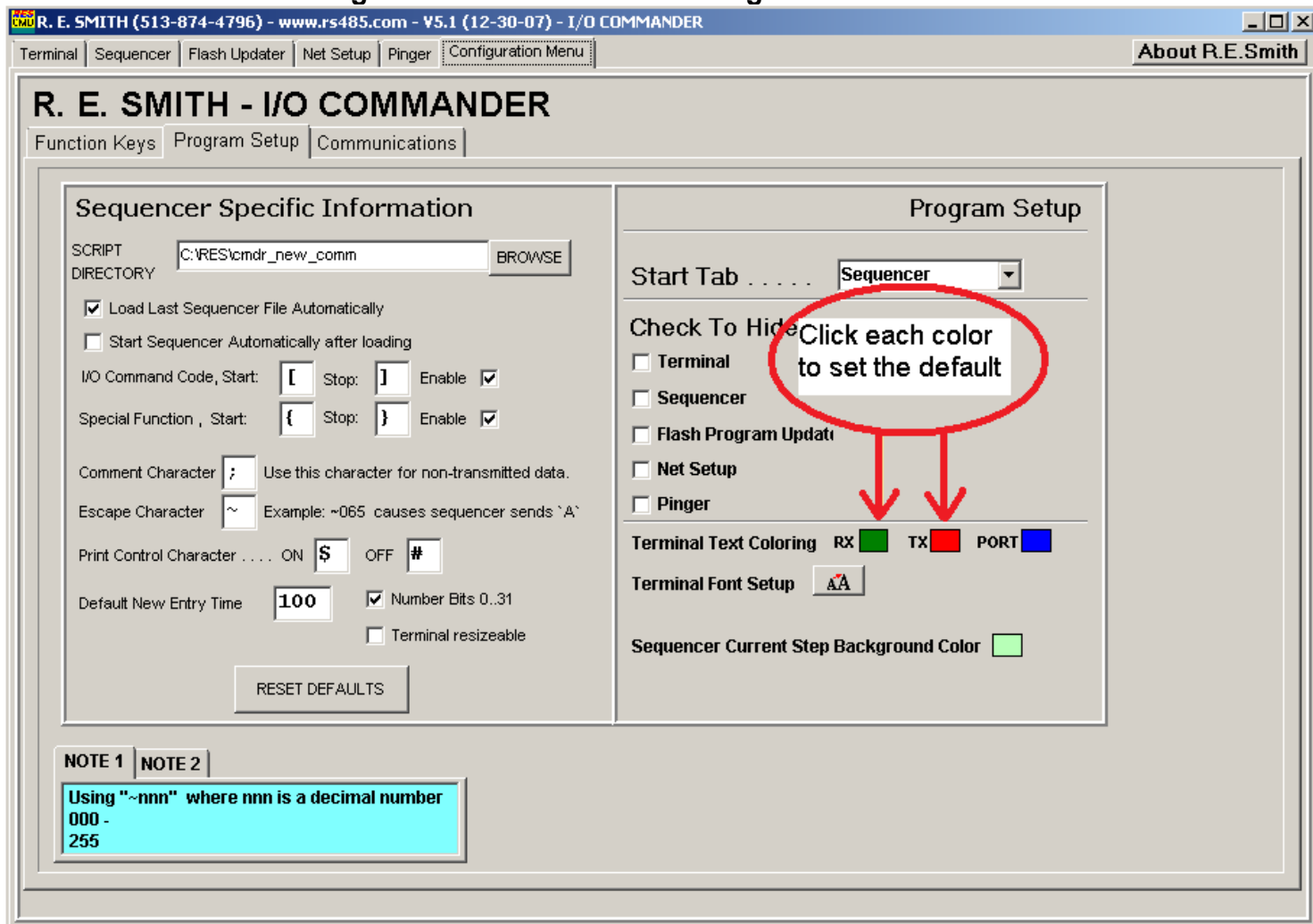



Figure 4. I/O Commander Terminal


R. E. SMITH (513-874-4796) - www.rs485.com - V5.1 (12-30-07) - I/O COMMANDER

Terminal
Sequencer
Flash Updater
Net Setup
Pinger
Configuration Menu
About R.E.Smith

Disconnect
Clear Buffer
Clear ALL
Start Log

```

*LOC(0FH) LUNIT(0FH) -0000#
*VER(0FH) VER-1.2A-20071027#

```

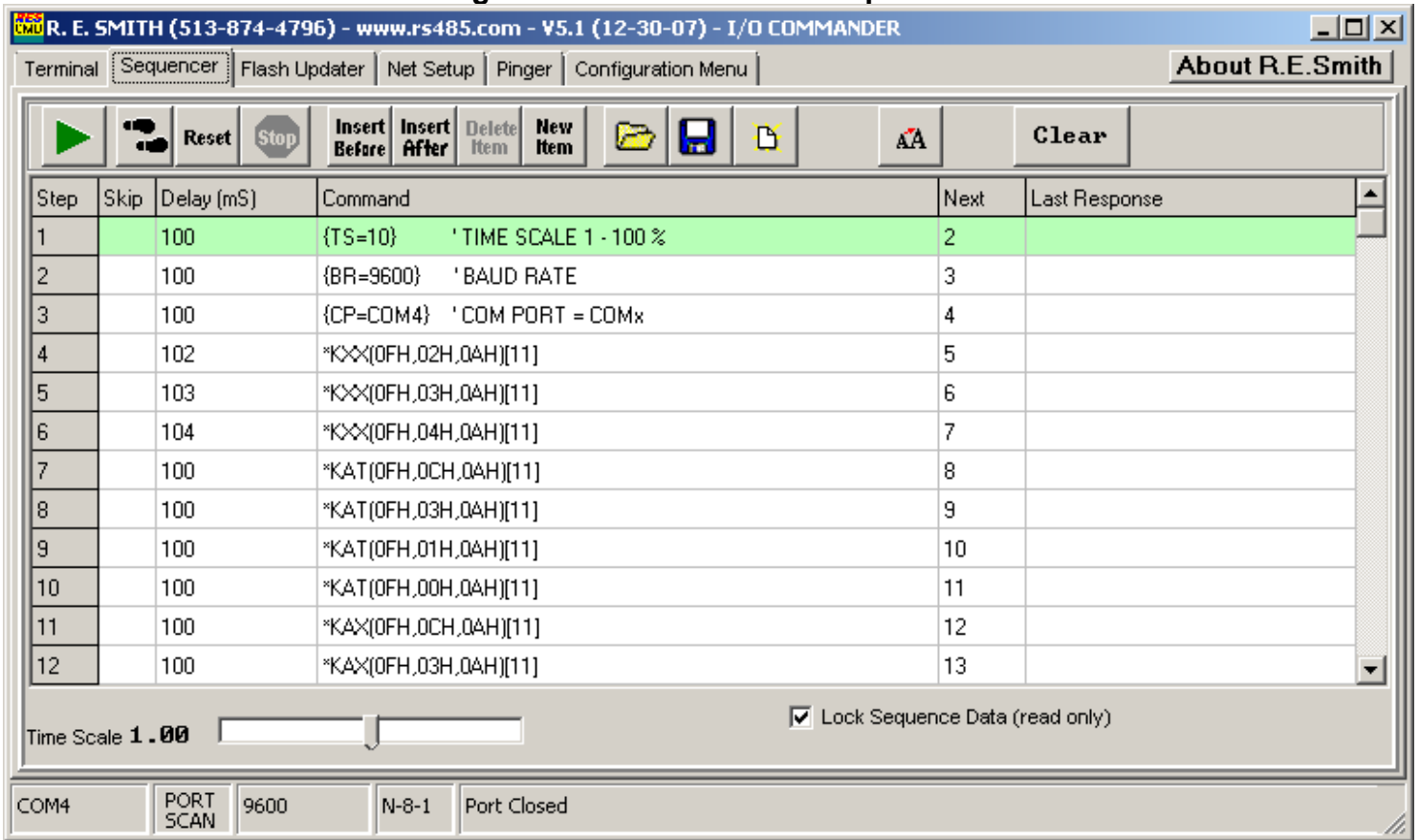
COM4

MODE ASCII
☒ CAPS LOCK
☐ RX CR = CR+LF
FULL

PF1616W	USB84RA	PF8R	BASIC-PF-CMDS	IIDHX4	AT84R-NEW	PELCO	PF10RMC
PFWDI	SPECIAL FUNCT	SET 11	SET 12	SET 13	SET 14	PF8RMC-PELCO	SET 16
*LOC	*GET	*VER	*K1-5S	*K1-OFF	*K1-ON	*TIMERS	F8
K0000	K5555	KAAAA	KFFFF	KXX00	KXX55	KXXAA	KXXFF
K00XX	K55XX	KAAXX	KFFXX	KFXXX	KXFXX	KXXFX	KXXXF
K0XX0	KFXFX	KX00X	KXFFX	OPT-READ	OPT-EVNT=Y	OPT-EVNT=N	OPT-AUTO=10
*TEST	*TIMERS						

COM4
PORT SCAN
9600
N-8-1
Port Open - COM4,9600,N-8-1,NO FLOW

Figure 5. I/O Commander Sequencer



Sequencer Commands Configuration Commands

It is possible to configure some of the communications port settings as well as the time scale setting using a special command on a sequencer line.

Command	Description
{BR=baud}	Set the baudrate on a sequence step
{TS=value}	Set the time scale slider to a value from 1-100, 1=.02X time scale, 50=1X time scale, 100=10X time scale
{CP=comX}	Set the current communications port.

Additional Sequencer Commands

The following table shows some characters have been reserved to implement some special functions on the data that is received. These functions include conditional branching based on the returned data. Also some characters can be transmitted using an escape sequence starting with a special user definable character. Settings on the Program Setup tab control the defaults.

Any character sequence can be part of a command, however, commands should not include the following characters [,], !, * and ~.

!, * and ~ can be reconfigured on the Configuration Menu/Program Setup tab.

[n]	Receive n characters, wait until n characters are received or timeout before going to next sequencer step.
[C(n, comptype, compval, tseq, fseq)]	Compare received hex value of n digits to compare compval with comparison type comptype and jump to false sequence number fseq or true sequence number tseq .
[B(n, comptype, compval, tseq, fseq)]	Bit Test received hex value val to compval jump to false sequence number (fseq) or true sequence number tseq .
!	Turn display of receive characters off. Note this character is configurable on the Configuration Menu/Program Setup tab.
*	Turn display of receive characters on. Note this character is configurable on the Configuration Menu/Program Setup tab.
~065	Send ascii character 65.

[n] – This operator is used to consume a number (n) characters from the external device. This is used to consume the responses from commands sent to the external unit. This data is taken from the communication port input buffer and is not stored.

[B(n, CompareType, BitNum, BranchTrue, BranceFalse)] – This operator is used to consume a number (n) characters from the external device, it then constructs an appropriate integer value from the input and performs the given bit test using the BitNum. If the test is successful then execution continues on line indicated by BranchTrue, otherwise execution continues on line indicated by BranceFalse parameter.

[C(n, CompareType, HexValue, BranchTrue, BranceFalse)] – This operator is used to consume a number (n) characters from the external device, it then constructs an appropriate integer value from the input and performs the given comparison using the HexValue. If the test is successful then execution continues on line indicated by BranchTrue, otherwise execution continues on line indicated by BranceFalse parameter.

Comparison Type	Branch On Bit	Branch on Comparison
EQ - Equal		X
NE - Not Equal		X
GT - Greater		X
GE - Greater or Equal		X
LT - Less Than		X
BS - Branch if Bit Set	X	
BC - Branch if Bit Clear	X	

Example: "R5C[C(2,GE,01,3,5)][2]" This command addresses and reads the contents of register 5C, the IO board responds with data for 5C and 5D. This command compares the first two characters returned as an integer constructed with two hex digits to the value 01 hex. If the value is greater than or equal to 01 then execution continues on line 3 otherwise execution continues on line 5.

Example of hex value comparison: [C(4,NE,FFFF,8,9)]

The Comparison is of the hex value represented by the next 4 received characters being not equal to the unsigned value 0xffff. If the value is not equal the sequencer continues on to sequence statement 8 otherwise it goes to sequence statment 9.