Jupiter Model 538

Programmer's Reference Guide

www.rfsquared.com

www.tentec.com TEN-TEC

Contents

Introduction

The Ten-Tec Jupiter DSP HF transceiver is a product that is defined more by firmware than hardware. The transciever, front panel and PC Interface is controlled by firmware running on a single 16 bit DSP processor operating at 30 MHz.

Ten-Tec has produced this document as a starting point for software developers undertaking the development of a PC based Jupiter (Model 538) Interface program.

Interface Settings

The RS-232 serial interface on the Jupiter is controlled by a 16C550 UART located on the DSP board. The interface parameters are fixed at 57,600 baud, No Parity, 8 Data bits, 1 Stop bit. The UART uses hardware handshaking to control the data flow between the PC and the radio. The host PC should be set to use RTS/CTS signalling.

Conventions used in this manual

Numeric Types:

0x0A, 0Ah	Hexadecimal Numbers.
10	Decimal Number.
'A'	ASCII character code.
	example: ASCII 'A' is 0x41.
BCD	Binary Coded Decimal format
	example: decimal 25 is 0x25 in BCD format

The JUPITER Command Set

The JUPITER command set is similar to the command sets for the RX-320 & RX-350 receivers and Pegasus Transceiver. While the Jupiter can emulate the Pegasus transceiver it precludes front panel operation while using the PC interface. The command set presented here is a secondary interface which does not disable front panel operation. It is important not to mix Pegasus and Jupiter command sets. **Mixing commands could lead to unpredicatable operation.** Documentation for the Pegasus command set can be found in the Pegasus Programmer's Reference Guide available at www.rfsquared.com.

The command set is extensive and every effort has been made to keep the individual commands as simple as possible. Although the JUPITER interface operates at 57,600 baud short commands are used to keep the processing overhead to a minimum.

In general an JUPITER instruction is a two character command which may be followed by data and then terminated by a carriage return <cr>. The JUPITER operates with binary data and unless otherwise noted all commands that require numeric data will need that data to be in binary format. Using a binary format requires fewer bytes when compared to an ASCII format. In binary format a number like 25 decimal is represented as 19 hexadecimal or 19h or may be written as 0x19. All refer to the same number.

The DSP system provides control over the FRONT PANEL user interface as well as MODE, FREQUENCY, FILTER, AGC MODE, SPEAKER LEVEL and other features. In addition, the DSP can respond to requests for SIGNAL STRENGTH, DSP FIRMWARE REVISION and other status information.

In general, the JUPITER will accept any provided data as being valid. It is up the programmer to ensure that the supplied data is correct. Where a command has a limited number of data options, such as the AGC MODE command, failure to provide a valid selection may result in the radio choosing a default setting. There will be no notification that the data is invalid. Where a command is unrecognized the radio will send back a response consisting of a single letter 'Z' followed by a carriage return <cr>. Some commands may have data fields that may contain the binary number 0x0d which is also the carriage return character. The firmware in the JUPITER that is responsible for parsing the command strings has the intelligence to properly handle carriage returns (0x0d) embedded in the data field.

Because the JUPITER was built to be reprogrammed in-system the command set presented here is subject to change or enhancement. We will make every effort to make the system backward compatible with existing documented commands whenever possible. However, the JUPITER is an HF TRASNCEIVER PLATFORM that could host a variety of radio services. Persons or companies developing control software for the JUPITER should not assume that the radio is operating original factory firmware but rather should always query the radio's firmware revision to ensure compatibility.

Jupiter Programmer's Reference Guide
Feedback? Comments? ditsnbits@tentec.com.

Rev 1.1

JUPITER Command Set Quick List

Frimware version 1.18

*A	Vfo A Freq (binary)
*B	Vfo B Freq (binary)
*G	AGC Control
*H	Squelch Set
*I	RF Gain
*J	Attn
*K	NB, NR, AN - (NB 0-7, NR 0/1 AN 0/1)
*M	Set Receiver Modes
*P	PBT setting
*R	Recall mem
*S	Store Mem
*U	Set Speaker Volume
*W	Filter Selection
XX	DSP Restart Command
*Z	Flash Update Mode

?A	vfo A query
?B	query forward power
?G	query AGC mode
?H	query squelch setting
?I	query RF gain
?J	query Attenuator setting
?K	query NB, NR and AN settings
?M	query receive modes
?P	PBT query
?R	query reflected power
?S	query S-units
?U	Query Volume Setting
?V	Query Version
?W	bandwidth query
?X	Query DSP portion of signal strnegth
?Y	Query analog portion of signal strength

Mode Codes
  '0' - AM
  '1' - USB
  '2' - LSB
  '3' - CW
  '4' - FM

Restart and Notify

This command will cause the radio to do a software restart. Upon start-up the radio will issue the start message "RADIO START" indicating that it is initialized and ready to accept commands. Sending this command to the JUPITER will cause the DSP firmware to reinitalize and generate the "RADIO START" message. This command can be used to determine that a Ten-Tec Flash Based radio is attached.

format: 'X' 'X' <cr>

where: 'X'= is the ASCII 'X' character (0x58)

                         <cr> = ASCII carriage return (0x0D)

response: " RADIO START"

Receive Modes

The JUPITER supports AM,FM,LSB,USB,and CW modes.

format: '*' 'M' vfoAmode vfoBmode <cr>

where: '*' = the ASCII '*' symbol (0x2a).

                         'M' = the ASCII M character (0x4d).

                         vfoAmode, vfoBmode =

                                 ASCII '0' (0x30) for AM mode

                                 ASCII '1' (0x31) for USB mode

                                 ASCII '2' (0x32) for LSB mode

                                 ASCII '3' (0x33) for CW mode

                                 ASCII '4' (0x34) for FM mode

                         <cr> = ASCII carriage return (0x0D) character.

response: none.

example: *M11<cr> for USB on VFO-A, USB ON VFO-B.

                       M23<cr> for LSB on VFO-A, CW on VFO-B.

Speaker Output Control

The speaker volume control sets the gain of the audio amplifier. The control range is off (0x00) to full on (0x7f). The input to the audio amplifier is set by the Line Level Audio Control. Refer to the Line Level Audio command for further information.

format:	'*' 'U' nn <cr>
where:	'*' = the ASCII '*' symbol (0x2a).
	U = 'U' (Ascii 0x55) character.
	nn = speaker volume 0 to 127 (0x00 - 0x7f)
	<cr>=ASCII carriage return (0x0D)
response:	none
example:	*U 0x20 0x0D sets the speaker to level 32
default:	0 (no output)

Receive Filter

The JUPITER contains a large number of selectable filters that can be used in AM, LSB, USB, and CW modes.

format:	'*' 'W' fn <cr>
where:	'*' = the ASCII '*' symbol (0x2a).
	'W' = the ASCII W (0x57) character.
	fn = a filter number in the range 0 thru 33 (binary). From Table 1 below.
	<cr> = ASCII carriage return (0x0D) character.
response:	none.
example:	*W 0x10 <cr> for filter number 16, 2100 Hz bandwidth.

Filter #	Bandwidth	Filter #	Bandwidth	Filter #	Bandwidth
0	8000	12	2850	24	1050
1	6000	13	2700	25	900
2	5700	14	2550	26	750
3	5400	15	2400	27	675
4	5100	16	2250	28	600
5	4800	17	2100	29	525
6	4500	18	1950	30	450
7	4200	19	1800	31	375
8	3900	20	1650	32	330
9	3600	21	1500	33	300
10	3300	22	1350		
11	3000	23	1200		

Frequency Tuning

Tuning the JUPITER over the serial interface is accomplished by two commands that provide for frequency entry into either the A or B vfo. The frequency data is the desired frequency in Hz formatted as a 4 byte binary number.

format:	'*' ['A' or 'B'] b3 b2 b1 b0 \<cr\>
where:	'*' = the ASCII '*' symbol (0x2a).
	A = the ASCII 'A' character (0x41).
	B = the ASCII 'B' character (0x42).
	b3-b0 =
	Frequency in Hz as a 4 byte binary number. The most sugnificant byte is b3. The least significant byte is b0.
response:	none.
example1:	*A \<0x00\> \<0xe4\> \<0xe1\> \<0xc0\> \<0x0d\> would set the A vfo to 15,000,000 Hz or 15 MHz.
example1:	*B \<0x00\> \<0x5b\> \<0x2b\> \<0xd8\> \<0x0d\> would set the B vfo to 5,975,000 Hz or 5.975 MHz.
default:	none

Passband Tuning (PBT)

The passband tuning range of the JUPITER is +/- 8192 Hz. Setting the PBT to 0 will turn the passband tuning control OFF. The data format is a 2 byte binary number which represents the PBT value in Hz.

format:	'*' 'P' b1 b0 \<cr\>
where:	'*' = the ASCII '*' symbol (0x2a).
	P = the ASCII 'A' character (0x50).
	b1-b0 =
	PBT frequency in Hz as a 2 byte binary number. The most sugnificant byte is b1. The least significant byte is b0.
response:	none.
example1:	*P \<0x03\> \<0xe8\> \<0x0d\> would set the PBT to 1000 Hz.
example2:	*P \<0x00\> \<0x00\> \<0x0d\> would set the PBT to OFF.
default:	none

Jupiter Programmer's Reference Guide
Feedback? Comments? ditsnbits@tentec.com.

Rev 1.1

AGC Mode Control

The PEGASUS can operate in SLOW, MEDIUM or FAST AGC mode. At power-up the radio will default to MEDIUM AGC mode which will provide the most versatile response. The AGC mode may be changed at any time to take advantage of changing band conditions.

format:	'*''G' cc <cr>
where:	'*' = the ASCII '*' symbol (0x2a).
	G = the ASCII 'G' character (0x47).
	cc = Control Character
	ASCII '1' (0x31) for SLOW
	ASCII '2' (0x32) for MEDIUM (default)
	ASCII '3' (0x33) for FAST
response:	none.
example:	G 1 0x0D for SLOW AGC Mode.
default:	MEDIUM AGC

RF Gain Control

The relative RF gain can be controlled over a range of 0-127. A setting of 0 represents full RF gain whereas a setting of 127 (0x7f) represents the minimum RF gain level. Because this control directly affects the RF hardware this will directly affect S-Unit responses and Squelch settings.

format:	'*' 'I' gn <cr>
where:	'*' = the ASCII '*' symbol (0x2a).
	I= The ASCII 'I' character (0x49)
	gn = 7 Bit Attenuation level (0x00 to 0x7f)
	<cr> = ASCII carriage return (0x0D)
response:	none.
example:	* I <0x20><0x0d> sets the RF Gain to 0x20 or decimal 32.
default:	0 (minimum gain)

RF Attenuaton Control

An RF Attenuator may be switched in or out under software control. The Attenuator applies approximately 15 db of attenuation. Because this control directly affects the RF hardware this will directly affect S-Unit responses and Squelch settings.

format: '*' 'J' ch <cr>
where: '*' = the ASCII '*' symbol (0x2a).
              J= is the ASCII 'J' (0x4A) character
              ch = Setting
                       '1' (0x31) = ON
                       '0' (0x30) = OFF
              <cr> = ASCII carriage return (0x0D)
response: none.

example: * J <0x30><0x0d> sets the Attenuator OFF.

default: Off

Squelch (all mode)

The squelch is active in all modes. The adjustment range is 0 to 127 representing a 0 to 100% range.

format: '*' 'H' su <cr>
where: '*' = the ASCII '*' symbol (0x2a).
              'H'= is the ASCII 'H' (0x48) character
              s = squelch setting (0 - 127)
              <cr> = ASCII carriage return (0x0D)
response: none.

example: *H<0x08><0x0d> sets the squelch level to 8.

default: none

Jupiter Programmer's Reference Guide
Feedback? Comments? ditsnbits@tentec.com.

Rev 1.1

Memory Store

The JUPITER 128 memories. The memory store command will write the current contents of A and B VFOs to the indicated memory channell.

format: '*' 'S' nn <cr>

where: '*' = the ASCII '*' symbol (0x2a).

                 'S' = the ASCII S character (0x53).

                 nn = the memory channel in a one byte binary format.

                 <cr> = ASCII carriage return (0x0D) character.

response: none.

example: *S <0x10><0x0d> would write the current vfo A, vfo B state to memory channel 16 (0x10).

Memory Recall

The JUPITER 128 memories. The memory recall command will recall the selected memory channel. If the the memory is empty the operation will not be performed. If the memory data is valid the stored data will replace the vfo contents.

format: '*' 'R' nn <cr>

where: '*' = the ASCII '*' symbol (0x2a).

                 'R' = the ASCII S character (0x).

                 nn = the memory channel in a one byte binary format.

                 <cr> = ASCII carriage return (0x0D) character.

response: none.

example: *R <0x10><0x0d> would recall the contents of memory channel 16 <0x10|>.

Noise Blanker, Automatic Notch and Noise Reduction

The Noise Blanker (nb), Automatic Notch (an) and Noise Reduction (nr) are available for use in all modes. They can be used individually or together.

format: '*' 'K' nb an nr \<cr\>

where: '*' = the ASCII '*' symbol (0x2A)

        'K'= is the ASCII 'K' character (0x4B)

        nb = Noise Blanker State (0 =off action 1-7)

        an = Auto Notch State (1=on 0 =off)

        nr = Noise Reduction State (1=on 0=off)

        \<cr\> = ASCII carriage return (0x0D)

response: none.

example: *K\<0x05\>\<0x00\>\<0x01\>\<0x0d\>

        Sets the Noise Blanker to 5, Auto Notch OFF and Noise Reduction ON.

default: an, nr and nb all off.

12

Jupiter Programmer's Reference Guide
Feedback? Comments? ditsnbits@tentec.com.

Rev 1.1

Query

The JUPITER query command that allow a host PC to determine the state of various radio controls. In general, each control setting command has a query counterpart. See the specifc command for the format of the data returned.

format: '?' ch <cr>
where: '?'= is the ASCII '?' (0x3F) character
                ch = ASCII code of the parameter

	?A	vfo A query
	?B	vfo B query
	?G	query AGC mode
	?H	query squelch setting
	?I	query RF gain
	?J	query Attenuator setting
	?K	query NB, NR and AN settings
	?M	query receive modes
	?P	PBT query
	?S	query S-units
	?U	query Volume Setting
	?V	query Version
	?W	bandwidth query

                <cr> = ASCII carriage return (0x0D)

responses not listed here are formatted like the associated control parameter.

'?V' (Version Query)
String response in the form 'VER 1180-538' which indicates version 1.180 for model 538.

'S' (Signal Level)
Responds with the letter 'S' followed by the signal strength in 'S' units. The data format is a 4 byte ascii string. The upper part of the number is the signal strength in 'S' units while the lower part is the fractional part of the signal in 'S' units. For example: if the radio returns the string 'S0356' the signal strength is 3.56 'S' units.