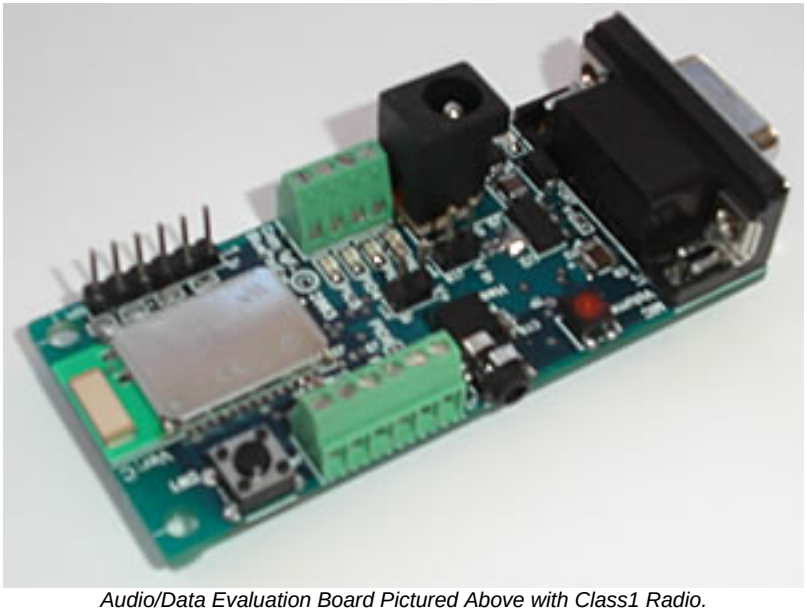
User Guide

For:

BlueRadios® Bluetooth®
Intelligent Serial Module AT Command Set



Audio/Data Evaluation Board Pictured Above with Class1 Radio.

By:

BlueRadios, Inc.

AT HOME. AT WORK. ON THE ROAD. USING BLUETOOTH WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.

Revision History

REV #	Date	Description	Author
0.9.6	8/07/2003	Initial Release	R.D. Jones
1.0.3	10/6/2003	Incorporated: PIN, Encryption, COD, and Factory Reset	M.J. Kramer
1.0.7	10/28/2003	Incorporated: Set Master with address of specific Slave	W.I. Tucker
1.1.0	11/3/2003	Production Release – Tuned performance	M.J. Kramer
1.1.3	12/1/2003	Fast Data Mode on BT connection. 3.1.5 set service name and pass BT address on connect, reorganized the ATSW25 commands, added 3.5 Audio SCO commands	R.D. Jones
1.1.5	12/24	Changed and modified ATSW24 & ATSW25 commands	R.D. Jones
1.1.6	1/16/04	ATMC executed over a remote RF link will ignore the data on the radios UART and ATMD or ATMF will enable it to pass again.	W.I. Tucker
1.1.7	2/1/04	ATSW26 password control locks the user definable settings Bluetooth connection default is now data mode on a BT connection to enable remote configuration of the radio over BT RF link connection	M.J. Kramer
1.1.8	3/15/04	Added ATSI,8 ATSI,9 ATSI,10 get status information, ATSW27 Set PIO5 pulse rate.	W.I. Tucker
1.1.9	4/21/04	Added ATMACLR command to clear ATSMA settings	M.J. Kramer
1.1.10	4/23/04	ATSIE1 (Get Status Information Extended to include more items)	R.D. Jones
1.1.11	4/28/04	ATSW25 added 4th flag setting for 1 = DUN Slave service option	
1.1.14	6/1/04	Beta version for testing purposes only	
1.1.15	7/1/04	Enabled ATSW21 Page Scan and Window commands. Modified Verbalization short form category 1 replies. See Appendix A	
1.2.7	10/20/04	Added headset service for ATSW25 flag, Fixed Sniff, not Park, up-dated the doc to remove <lf> for sending commands (not required).	
Firmware Structured Around 8Mbit Flash Modules			
2.8.1.0.0 *Production Release*	3/11/2005	Added ATSW25,**3**,X,X,X - Flag for slave undiscoverable. PIO(4) Triple mode design implementation Automatic RING auto answering for Headset ATDL – connect to last stored BT address ATLAST – display last stored BT address ATSESC – set escape sequence of ASCII characters ATSI,11 - PIO(5) pulse rate value ATSI,12 - esc character ATSI,13 - Inquiry and Master timeout settings ATSI,14 - TX Power factor ATSI,15 - PIN lock status ATSI,16 - Deep sleep status ATPARK – Park mode supported ATPAIR – Pair with another device ATUPAIR – Un pair from last device ATAPAIR – Address of last paired device ATSW20 – added stop and parity UART flag settings ATSP – Extended PIN length from 10 to 16 characters ATSW28 – Set Inquiry and Master connect timeouts ATSW29 – added 3-levels of security options ATSW30 - Enable deep sleep ATSPF – Sets Max Transmit power ATOP – Overwrite PIN ATFRST – Resets radio back to factory configuration PIO(3) – Wake up - input high 1msec. to interrupt CPU and to wakeup from deep sleep without losing fist byte of data on UART LINK,BTaddress - Verbalization added on Master and Slave when security is enabled	
2.8.1.1.0	5/17/2005	ATIL – Inquiry Last ATSSNIFF – Set/Store Sniff values permanently in Flash	

		ATCSNFF – Clears stored Sniff values ATSI,17 – Sniff settings ATSTORE – Variable storage ATREAD – Read back contents in storage ATLSTO – Set Link Supervisory Timeout ATSI,18 – Link Supervisory Timeout settings	
2.8.1.2.0	7/26/2005	ATUPAIR,X – Un Pair by index number: 00, 01, 02, and 03 ATUPAIRB,bdAddr – Un Pair by Bluetooth address. ATCPAIR – Clear all Paired or secured connected addresses ATSI,19 – View list of paired or secure connected device addresses Deleted ATAPAIR command since it is no longer needed ATSW25,0,0,0,3 Audio Gateway Server added	
2.8.1.3.0	10/12/2005	ATRSSI – Returns RSSI value when Bluetooth connected ATLQ - Returns Link Quality when Bluetooth connected ATSSW – Bypass PIO(4) hardware factory reset ATRSW – Read status of the above switch setting Added PAN Client functions section 3.0 ATDL – Permanently store last connected device in flash ATSI,14 up-dated to match the format for RSSI +/-00 ATSI,20 – Read IP auto connect address and port ATSSW,2,1 – Set Switch #2, auto connect IP (ON) ATSSW,2,0 – Set Switch #2, auto connect IP (OFF) ATRSW,2 – Read Switch #2 ATSW25,X,0,0,0 – added flag for auto connect BNEP on power up	
2.8.1.4.0	11/7/2005	ATSI,20 – Fixed a bug in the Port value hex conversion returned ATSW25,4,0,0,0 – Made Auto Connect BNEP/PAN more robust	

Note: To provide the best firmware architecture, design, and future profile support there is not 100% code backwards compatibility in regards to certain AT Commands and responses. Client's can specify any version of firmware that is under ***Blue*Radios** configuration control. We also have custom versions not listed above (i.e., repeater and multi-point connection firmware options).

1 Introduction

Scope: This AT Command Set document along with ***Blue*Radios®** evaluation board was created to enable developers and integrators an opportunity to evaluate wireless networks using *Bluetooth* technology. The goal is to make the transition to *Bluetooth* wireless networks as seamless and easy as possible for our clients. This document will explain how to establish *Bluetooth* communications between two ***Blue*Radios** for both data and voice applications in a point-to-point network (i.e., cable replacement, slave/master communications only).

"Our technology delivers a dynamic experience that comes out of the wireless delivery mechanism and the freedom to connect others."

*Mark J. Kramer – CEO of **Blue**Radios*

This document describes the hardware interface of ***Blue*Radios** Intelligent Serial Module. The Module is designed to be built into an embedded device and to provide a simple and low cost Bluetooth API interface. The module is designed to integrate with a wide range of applications and platforms with a simple electrical and software interface using AT commands.

Background: The ***Blue*Radios** evaluation board is designed to accommodate the Companies Class1 or Class2 *Bluetooth* radio modem serial modules with 2.4GHz RF ceramic chip antenna (pictured right). The ***Blue*Radios** SMT modules are *Bluetooth* ver1.2 compliant. The evaluation board enables a stable platform environment to test serial RS-232 cable replacement and audio communications over *Bluetooth* RF links before going directly to an embedded printed circuit board design and layout.

AT Commands: This document describes the protocol used to control and configure ***Blue*Radios** *Bluetooth* Serial Modules. The protocol is similar to the industry standard Hayes AT protocol used in telephone modems due to the fact that both types of devices are connection oriented. Appropriate AT commands have been provided to make the module perform the two core actions of a *Bluetooth* device, which is make/break connections and Inquiry. Additional AT commands are also provided to perform ancillary functions.

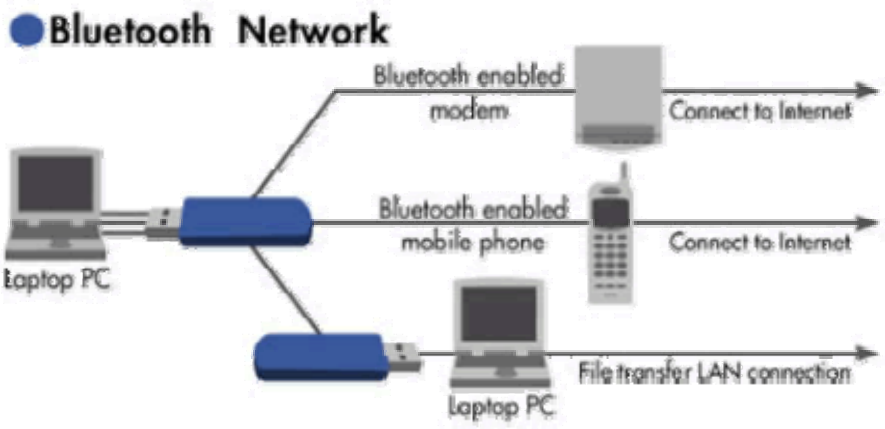
The CSR (Cambridge Silicon Radio) BC02 chipset in the ***Blue*Radios** modules is memory resource limited therefore it is NOT proposed that there be full implementation of the AT protocol ***similar*** to an AT modem. In fact, the protocol is similar enough so the existing source code written for modems can be used with very little modifications for use with this serial module.

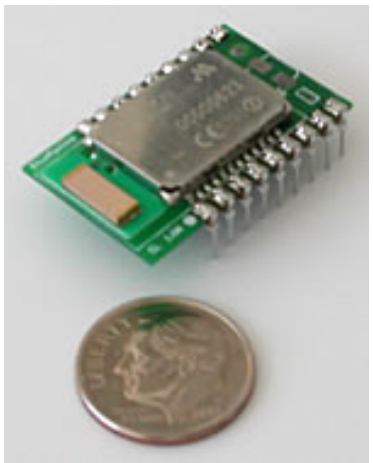
Just like telephone modems, the serial module powers up into an unconnected state and will respond to inquiry and connection requests. Then, just like controlling a modem, the host or client can issue AT commands which map to various *Bluetooth* activities. The command set is extensive enough to allow a host to make connections which are authenticated and encrypted or not. The ***Blue*Radios** serial radio modems can be configured, commanded, and controlled through simple ASCII strings through the hardware serial UART or over a remote *Bluetooth* RF connection.

1. All commands have the following format: “command”<cr>. Where “cr” represents carriage return 0x0D
2. Valid commands respond with a “<cr,lf>OK<cr,lf> or “<cr,lf>ERROR<cr,lf>. Where “lf” represents linefeed 0x0A. Only exceptions are ATSW20 and ATURST which do not reply.
    a. Only exceptions are when setting baud rate, commanding a CPU reset, or Factory Reset.
3. All response data after the command response have the following format <cr,lf>data<cr,lf>.

Applications: The ***Blue*Radios** evaluation board can be used for both embedded and PC product applications. It has a RS-232 DB-9 and J4 (0-3.3Vdc) direct UART interfaces to the module. There are radio modem input/output pins (PIO’s) connected to terminal lugs for applications that require external command and control. The design incorporates a 13bit mono audio codec, jack, and MIC volume control for wireless headset applications over SCO channel. The audio circuit has minimum filtering for noise, etc.

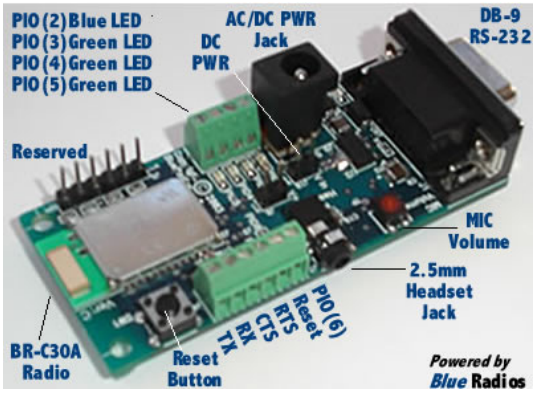
Pico-Nets: For applications that require more than point-to-point (2) devices communicating simultaneously – this is called a pico-net. These applications require one of the *Bluetooth* devices to manage all the network connections. The easiest implementation is using a Personal Computer (PC) that manages this activity with MS Windows *Bluetooth* stack software and USB *Bluetooth* Communicator plugged into the PC (see figure below).





BR-SC30A 18-pin DIP ***Blue*Stamp®**

Class1 USB Bluetooth Communicator (BR-USB)
HCI – Communications



RS-232 Audio and Data Evaluation PCB (BR-EC30A)

Note: AT Command interface protocol is not used for the USB Bluetooth communicator. The stack for this resides on the PC side not embedded in the unit like the serial SMT ***Blue*Radios** modules.

Making a Connection

Serial Interface

UART_TX, UART_RX, UART_RTS and UART_CTS form a conventional asynchronous serial data port. The interface is designed to operate correctly when connected to other UART devices such as the 16550A. The signaling levels are nominal 0V and 3.3V and are inverted with respect to the signaling on an RS232 cable. The interface is programmable. The default condition on power-up is pre-assigned in the external 8Mb Flash. Two-way hardware flow control is implemented by UART_RTS and UART_CTS. UART_RTS is an output and is active low. UART_CTS is an input and is active low. These signals operate according to normal industry convention.

***Blue*Radios** shows up under Service discovery defaulted as Serial Port Profile (SPP) Service "COM0 on BlueRadios", where COM0 is the arbitrary service name and BlueRadios is the local device name. All of these name settings are configurable by the user.

To connect to ***Blue*Radios**, browse for services, you should see: "***Blue*Radios** "Serial Port" as the Profile. ***Blue*Radios** uses SPP as default, and will be connected to a Virtual COM port on PCs, Palm Pilot's, PocketPCs, or other clients. Once connected, the *Bluetooth* address for each device is exchanged with the message CONNECT, displayed, and data will flow in both directions in regular data mode as if the serial port were locally attached. AT commands can be sent directly to the radios UART when not Bluetooth connected or by any remote *Bluetooth* RF device connection after typing **+++** followed by a carriage return. The **+++** is the factory default escape sequence to place the radio in command mode when there is a *Bluetooth* RF connection. The Slave radios UART will respond automatically with NO CARRIER when it is disconnected. This verbalization response can be changed to short/long or none if preferred using ATSW24 power up default settings.

***Blue*Radios** Class1 *Bluetooth* device with a high power transceiver (100meters/330 feet) or (10 meters/33 feet) for Class2 performance, however; actual range may vary due to environment, type of antenna, board layout, enclosure design or type of client device used to connect to ***Blue*Radios**. We have an AT Command to control and set the maximum RF output power.

NOTE: Only one device can make connection to ***Blue*Radios** at a time, and there is a limit of 8 simultaneous communicating devices in a *Bluetooth* pico-net network.

***Blue*Radios Evaluation Board Physical Ports (2nd Generation)**

***Blue*Radios Signal Name**	**RS-232 DB-9 Female**	**0-3.3Vdc TTL Terminal Lugs and Connector Posts**	**IO DIR**
Board PWR		J9-6 (AC/DC Jack)	Power IN (5.0 -12.0 Vdc)
Board GND	5	J4-1, J5-1, J8-1,	<—>
J1 Pin 1 - +V	NC	NC	Not connected
J1 Pin 2 - TX	2	J4-2 (3.3Vdc)	OUT→ 0 - 3.3Vdc
J1 Pin 3 - RX	3	J4-1 (3.3Vdc)	IN← 0 - 3.3Vdc Max Rated
J1 Pin 4 - DTR	4	NC	Not connected
J1 Pin 5 - GND	5	J4-1, J5-1, J8-1	<—>
J1 Pin 6 - DSR	6	NC	Output to PC
J1 Pin 7 - RTS	7	Yes J4-3 (3.3vdc)	OUT→ *(active low)
J1 Pin 8 - CTS	8	Yes J4-4 (3.3Vdc)	IN← * (active low) Max Rated
J1 Pin 9 - RING	9	or J5-6	Optional External Power →IN (5.0 -12.0Vdc)
PIO#2 J3-4	NC	Yes J3-4	Bluetooth connection made OUT→ 3.3Vdc (high state) Sink current is 4mA max.
PIO#3 J3-3	NC	Yes J3-3	User definable. IN← 3.3Vdc >1msec pulse interrupt to wakeup CPU from deep sleep without losing first byte of data on UART. Takes 5msec. for CPU to wakeup.
PIO#4 (**Triple Purpose)**	NC		1) Reset Default AT Settings IN← 3.3Vdc for 1 second during initial power up of module. Allow 5 full seconds for change. If not used tie to ground. 2) Strobe >5msec. to take radio out of Fast data into Command mode and maintain the Bluetooth RF connection. If not RF connected will place radio in Command mode. 3) Strobe >5msec. will auto connect to paired or last Bluetooth connected device if not already RF connected.
PIO#5 J3-1	NC	Yes J3-1	Pulses 1/sec. 0-3.3Vdc Sink current is 4mA max.
PIO#6 J4-1	NC		User definable. Sink current is 4mA max.
PIO#7	NC		User definable. Sink current is 4mA max.
6-Pin SPI			
GND J5-1	NC		Optional Ground
MOSI J5-2	NC		Reserved for BlueRadios

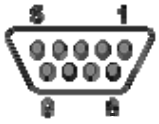
SPICK J5-3	NC		Reserved for BlueRadios
SPICS J5-4	NC		Reserved for BlueRadios
MISO J5-5	NC		Reserved for BlueRadios
PWR J5-6	NC		Optional External Power
J4-1 GND	J1-5	J4-1	Ground
J4-2 RST Terminal Lug (active high)	NC	J4-2	Soft boots on RST radio pin IN←3.3vdc strobe >5msec.
J4 pins 3,4,5,6	NC	0-3.3Vdc only not RS-232 levels	RTS,CTS, RXD,TXD connect directly into the radio
Reset Push Button Switch (black) active high		SW1	Soft boots CPU on RST radio pin IN←3.3vdc strobe >5msec.
RST Pin (active high)		J5-6	Soft boots on RST radio pin IN←3.3vdc strobe >5msec. Module has an internal 1K ohm pull down
GND J8-1	J1-5		Use to stimulate PIOs GND (low)
3.3Vdc J8-2			Use to stimulate PIO's 3.3Vdc (high)
Audio Jack (2.5mm)	NC	J7	Use with audio headset

Important Notes:

Placing 3.3Vdc into the PIO's while they are set as outputs will permanently damage the radio modules. The failure mode is short across GND and VCC. When experimenting with the evaluation board use a 10KΩ series resistor when applying power to the terminal screw PIOs directly on the UART.

- Make sure to connect a common ground when using the external TX, RX inputs on the 0 – 3.3Vdc terminal lug connector J4 of the evaluation board.
- If you strobe PWR or GND to the top of the terminal lug screw heads make sure the screw is tighten down or it may not connect the circuit (open circuit).
- For a 3 wire DB-9 interface (tx, rx, gnd only) connect/short CTS to RTS, (J1-7&8). Factory default is hardware flow control enabled CTS and RTS connected.
- PIO's are 0-3.3Vdc not 5 volt tolerant.
- Disconnect RS-232 cable if using 3.3Vdc TX&RX input on J4 terminal lug connector. The Maxim RS-232 chip senses which data input is used between J1&J4 and it sometimes reacts to noise on the DB-9 connector if it is still connected while using J4.
- Use standard pass through RS-232 serial cable. A null modem adaptor is not required.
- The module must be reset with terminal 5 "RESET" after turning on the power supply VDD. Reset terminal should be high for >5 msec. to cause a reset incase of electrical "brown-out" or poor input supplied VDD. Allow 500msec for module to fully reboot. Unit will not initially boot-up reliably if the VDD ramp rate is in milliseconds.
- Tie PIO(4) "Factory Reset" to ground if not in use to prevent inadvertent resetting of parameters during initial module power up for you modules. The remaining pins can float.

You can connect the RS-232 DB-9 evaluation board directly to the PC without a RS-232 pass through cable or null modem.



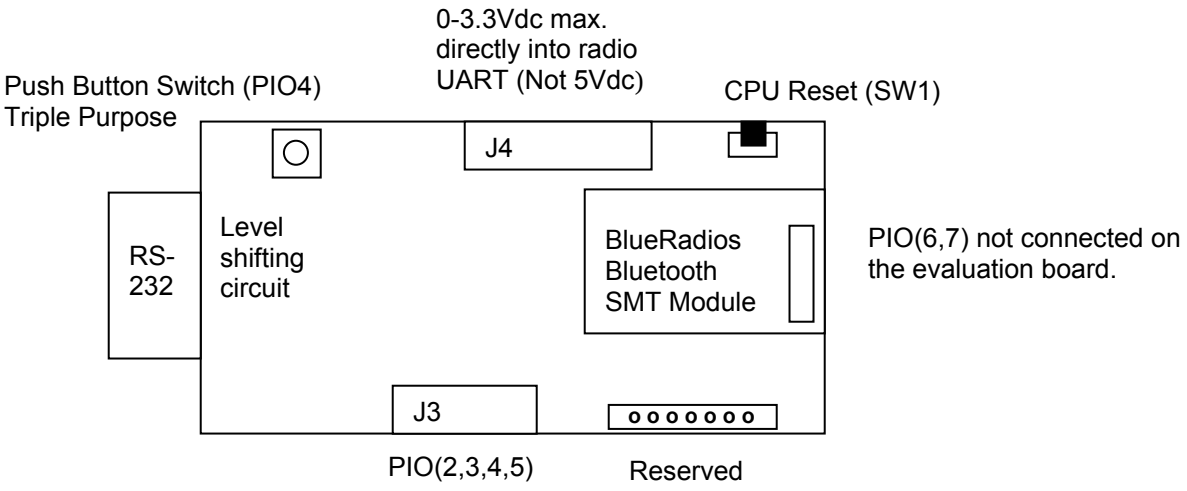
Female DB-9

Power Terminals for Evaluation Board

Inputs on J9 & J10 can be ≥5VDC and ≤12.0VDC. Worst case power draw for the entire evaluation board is 150ma when the *Bluetooth* radio/modem connection is established and transmitting. Power consumption is much lower depending on parameter settings.

Hardware UART Communications Connections for Modules and Eval Board
Radio module **TX UART** → **RX** of the application Micro Controller Unit (MCU)
Radio module **RX UART**← **TX** of the application Micro Controller Unit (MCU)
Radio module **RTS UART**→ **CTS** of the application Micro Controller Unit (MCU)
Radio module **CTS UART**← **RTS** of the application Micro Controller Unit (MCU)

Evaluation Board Block Diagram



LEDs

The ***RED*** LED next to power terminals should come on whenever the unit has power supplied. The ***Blue*** LED on the PIO(2) should go-on whenever the unit is connected to another Bluetooth device. The remaining **Green** LED's are defined for the following PIO table:

Radio Module I/O	**Class1 Radio Module BR-C30 ver1.2** **Class2 Radio Module BR-C29 ver1.2**	**LED Color on Evaluation Board**
PIO(2)	Yes	**Blue** *BT Connection Established*
PIO(3)	Yes	Input only - >1msec deep sleep wakeup interrupt. Takes CPU 5msec to wakeup.
PIO(4) **(Triple Purpose)**	Yes – Restores Factory Default Settings when held high for 1 second during initial power up. 2) >5msec pulse after power up will place radio into command mode. 3) >5msec pulse will connect to paired device or last connected device.	**Green**
PIO(5)	Yes – Pulses 1/sec. for Slave mode indication and if processing Master inquiry requests	**Green**
PIO(6)	Not connected on Eval Board	*User definable (defaults as input)*
PIO(7)	Not connected on Eval Board	*User definable (defaults as input)*
All PIO's	Max Sink Current is 4mA max.	

Changing Configuration

Parameters, such as the *Bluetooth* Name, Service Name, Class of Device and Serial Port settings can be viewed and configured. This can be done locally through the serial port UART or from a remote *Bluetooth* RF link. To access configuration, the ***Blue*Radios** must be in command mode and enable to except AT Commands. While in command mode ***Blue*Radios** will accept ASCII bytes as commands.

Use a normal RS-232 pass through cable from PC passing ASCII characters through the terminal to the ***Blue*Radios**. The communications settings should match the settings used when ***Blue*Radios** connects, for example: the default is 9600bps, 8 Data Bits, No Parity, 1 Stop Bit, and hardware flow control enabled. Once you change these parameters, you have the option to store them permanently in non-volatile memory.

Run your favorite terminal emulator, HyperTerminal or other program. Type "**AT**" on your terminal screen and follow it with a carriage return <cr>. You should see "**OK**" returned to you. This will verify that your cable and communications settings are correct on the radios hardware UART. Make sure you do not have the scroll lock enabled on the keyboard. When the radio is not connected to another *Bluetooth* device you can type the AT commands directly into the radios UART. If the radio is Bluetooth connected you will need to send **+++**<cr> on the local UART to take the radio out of data mode and place it in command mode. You can enter any of the AT commands in the up coming sections, followed by <cr>. Commands will return an "**OK**", for valid response, and invalid ones will reply ERROR. To go back to regular data mode while RF connected type ATMD to pass or receive data from a remote connected *Bluetooth* device.

Note: If changing communications parameter settings, remember to change your terminal or emulator comm. settings to correspond to the new parameter settings you just have made.

Also, we have seen some strange communications effects using HyperTerminal in conjunction with a PC using various *Bluetooth* stack and virtual com ports. As an example; communications works only in one direction. Requires closing both HyperTerminal programs and starting both HyperTerminal sessions again.

WARNING:

Refrain from streaming ASCII or binary data into the UART when the radio does NOT have a *Bluetooth* RF connection established while in Command Mode. This will overrun the UART Radio buffer and will not enable you to make a *Bluetooth* connection. When the radio is in the command parser mode it is looking for valid AT commands followed by <cr>. Either monitor PIO(2) going high, wait for the connection to occur, have the radio come up automatically in Fast Data Mode before you start sending data, or change the power up default settings ATSW25 to ignore UART data while unconnected.

The module must be reset with terminal 5 "RESET" after turning on the power supply VDD. Reset terminal should be high for >5 msec. to cause a reset incase of electrical "brown-out" or poor input supplied VDD. Allow 500msec for module to fully reboot. Module will not initially boot-up reliably if the VDD ramp rate is in milliseconds.

2 AT Command Prefix

AT	***AT The attention command prefix*** The prefix **AT** must precede every valid command. The remainder of the command script contains commands for the radio. The command line must end with a carriage return, **<cr> = <0x0d carriage return>.** *Note: If using HyperTerminal the following check box should be disabled: Send line ends with line feeds, if not the commands will not be submitted correctly.* **EXAMPLE:** *TYPE :* **AT<cr>*** *REPLY:* **<cr_lf>OK<cr_lf>** AT Commands can be upper or lower case. The only exception is the radios Personal Identification alphanumeric Number (PIN) is caps sensitive, ATVER,ver1 command, and ATOP.

**** All commands are typed exactly as shown***
<cr> = <0x0d carriage return>
<cr_lf> = <0x0d carriage return> <0x0a linefeed>

BlueRadios Firmware Version

AT	***VER,ver1*** Get Radios firmware version (***ver1*** is lower case sensitive). **EXAMPLE:** *TYPE :* **ATVER,ver1<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>Ver 2.8.1.4.0<cr_lf>** AT Commands can be upper or lower case. The only exception is the radios Personal Identification alphanumeric Number (PIN) is caps sensitive, the ***ver1*** above, and ATOP Command. Make sure this version number matches this document version before proceeding.

2.1 Get/Set Radio Information

2.1.1 Get Status Information

Status Information can be obtained directly from the *Bluetooth* Radio. This information is important when managing a connection list of devices in a local area and current settings of the radio.

SI	***,<n>*** ***Information*** **0** **BlueRadios AT** Module Type **1** Asks ***my*** radio for its product ID code (Bluetooth address ID).

2 Asks ***my*** radio for its local/friendly name.
3 Get current connection status of ***my*** radio.
(0 = master, 1 = slave, 2 = idle),(0 = disconnected, 1 = connected)
4 Get Service Name
5 Get Class of Device (COD) setting is undefined from factory.
6 Get *ATSW24 status {responseType, authMode, autoSCO, minorFilter}*
See page 12 for details – defaults for power up only.
7 Get *ATSW25 status {connectMode, comMode, unConnectedUartMode, Service}*
See page 13 for details – defaults for power up only.
8 Get radios UART baud rate, parity, and number of stop bits in **(HEX)**"0027,0000,0000" which is not set but defaults to 9600bps.
9 Get autoConnect Masters stored Slaves *Bluetooth* address & service if never used replies "Not Set!"
10 Get Slaves scan interval and window in **(HEX)** "0400,0200,0400,0200" respectively (number of time slots).
11 Get PIO(5) pulse rate in **(HEX) 04B0** = 1200msec.
12 Get radios escape character setting in **(HEX) 2B** = "+"
13 Get Inquiry and Masters timeout in seconds **(HEX)** 0014,003C
14 Get radios maximum transmit power level. Defaulted to max value in **(HEX)** +0F = +15dBm. Recommend settings of (*15dBm, *12dBm, 4dBm, and 0dBm) (*) Class 1 radio only.
15 Password Lock Code Status: 00 – Default normal, 01 – UART only, 02 – UART and RF. **(HEX)**00
16 Get Deep Sleep Status: 00 – No Deep Sleep (default), 01 – Deep Sleep enableed. **(HEX)**01
17 Get Stored Sniff Settings 0000,0000,0000,0000 **(Hex)** or Not Set!
18 Get link supervisory timeout setting **(HEX)** 04 (default), Integer 2-41
19 Get list of paired or secured connected device addresses
20 Get IP auto connect address and port number in **(Hex)**

EXAMPLE(s):

TYPE : **ATSI,0<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>BlueRadios AT<cr_lf>**

TYPE : **ATSI,1<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>112233445566<cr_lf>**

TYPE : **ATSI,2<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>BlueRadios<cr_lf>**

TYPE : **ATSI,3<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>1,0<cr_lf> // 1 = slave, 0 = unconnected**

TYPE : **ATSI,4<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>**

TYPE : **ATSI,5<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>00000000<cr_lf>**

TYPE : **ATSI,6<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>0,0,0,0<cr_lf>**

TYPE : **ATSI,7<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>0,1,0,0<cr_lf> // 1 = data/command mode**

TYPE : **ATSI,8<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>0027,0000,0000<cr_lf> // HEX values**

TYPE : **ATSI,9<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>**
or
REPLY: **<cr_lf>OK<cr_lf><cr_lf>00A09606E8EF,1101<cr_lf>**

TYPE : **ATSI,10<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>0400,0200,0400,0200<cr_lf> // HEX values**

	TYPE : **ATSI,11<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>04B0<cr_lf> // HEX value**
	TYPE : **ATSI,12<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>2B<cr_lf> // HEX value**
	TYPE : **ATSI,13<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>003C,003C<cr_lf> // HEX value**
	TYPE : **ATSI,14<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>default<cr_lf> // or HEX value if set +/-00**
	TYPE : **ATSI,15<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>00<cr_lf> // HEX value**
	TYPE : **ATSI,16<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>01<cr_lf> // HEX value**
	TYPE : **ATSI,17<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf> // or HEX values if set**
	TYPE : **ATSI,18<cr>** *REPLY:* **<cr_lf>OK<cr_lf><cr_lf>04<cr_lf> // HEX value**
	TYPE : **ATSI,19<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **00,<cr_lf>** **01,<cr_lf>** **02,<cr_lf>** **03,<cr_lf>**
	TYPE : **ATSI,20<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>00000000,0000<cr_lf> // HEX value**

2.1.2 Set the Radio name

When another Radio performs a Discovery, this will be the name that is passed to that radio. Please take note, unlike the name, the Radio's *Bluetooth* address is fixed (48bit) at the factory and is unique to every *Bluetooth* device manufactured.

SN	***,<name>*** **Set the RADIO name** Sets the Radios friendly name (16 alphanumeric characters MAX).
	EXAMPLE: *TYPE :* **ATSN,MYRADIOS_0123456<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**

2.1.3 Write Memory locations

SW	***,<n>*** ***Write to an S register*** The S registers refer to memory locations used for configuration. The S commands are used to assign values to various registers in the radio's Flash Memory that are stored in nonvolatile memory.
20	**,<baudrate>,<parity>,<stopbits>,<store> {UART Settings}**

Baud Rates: 1200 – 921.6Kbps (see table below).

flowControl - is always enabled and shorting CTS/RTS together if not used.

Parity:
0 = PARITY_NONE,
1 = PARITY_ODD,
2 = PARITY_EVEN,
Stop Bits:
0 = STOP_ONE,
1 = STOP_TWO,
Store Parameters:
0 = Do Not Store
1 = Store Parameters in Flash

EXAMPLE:
TYPE : **ATSW20,39,0,0,1<cr>** *// 9600 8,N,1 store in flash*
REPLY:

This unique Command does not reply with "OK" or "ERROR" because of internal UART data processing limitations and response timing.

Notes: factory default is: 9600 8, N, 1 hardware flow control RTS/CTS enabled. You can not change the number of data bits from 8.

To reconfigure Radio back to default factory settings apply 3.3vdc on PIO#4 duiring intial power up for 1sec.

Baud Rate	*Ascii Value*	*Error*
same	*0*	*-*
1200	*5*	*1.73%*
2400	*10*	*1.73%*
4800	*20*	*1.73%*
9600	*39*	*-0.82%*
19.2k	*79*	*0.45%*
38.4k	*157*	*-0.18%*
57.6k	*236*	*0.03%*
115.2k	*472*	*0.03%*
230.4k	*944*	*0.03%*
460.8k	*1887*	*-0.02%*
921.6k	*3775*	*0.00%*

Note: Contact ***Blue*Radios** *for calculating and setting custom baud rates not listed above. As long as the the equation BAUDRATE * 0.004096 produces an integer value, then there will be 0% error in clocking for the baud rate.*

21 **,<integer value>,<integer value>,<integer value>,<integer value> {Page Scan Interval and Window }** for **Slave device not connected** in time slots (N).

ATSW21,psInterval,psWindow,isInterval,isWindow // factory default is {1024,512,1024,512}

	Warning: Setting these will affect the inquiry and connection time. You could inadvertently set the scan interval too long and the window to short on the slave for a master connect request. Unless your application is battery powered slave and power conservation it is critical leave at the factory default settings. The minimum Window allowed by the Bluetooth spec is 11.25msec. If you set isWindow = 0 the Slave device will not be discovered by any Master but you can still use the Slaves BT address and connect directly to it from a remote Master. Page scan interval, Range 0x0012 to 0x1000, Time = N * 0.625msec, Range = 11.25msec to 2560msec, If set to 0 then page scanning is turned off **EXAMPLE:** *TYPE :* **ATSW21,4096,18,4096,18<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Requires a reset for the settings to go into affect.
22	**,<PIO#>,<state>,<store> {Set PIO State}** Configure state of PIO. 2 = PIO(2) Hard coded as output only (indicates a *Bluetooth* connection) 3 = PIO(3) (1 = Output, 0 = Input) defaults as input (user definable). (0 = Input only) >1msec. interrupt to wake-up CPU out of deep sleep mode if enabled. 4 = PIO(4) (0 = Input only) *This is used for resetting factory defaults on power up and breaking out of Fast Data mode while Bluetooth connected, also if strobed will auto connect to last paired or last connected device. (triple purpose) PIO.* 5 = PIO(5) (1 = Output) Hard coded as an output only *Strobes at 1 cycle per second indicating slave or master inquiry in process.* 6 = PIO(6) (1 = Output, 0 = Input) defaults as input (user definable) 7 = PIO(7) (0 = Input only) >1msec. defaults as input (user definable) ****CAUTION** this has to be done before using pio's as outputs** **Inputting 3.3Vdc into a PIO assigned as output will permanently damage the radio.** **Store Parameters:** 0 = Do not store 1 = Store Parameters in Flash **EXAMPLE:** *TYPE :* **ATSW22,3,0,1<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**
23	**,<PIO#>,<value>,<store> {set PIO Logic}** 2 = PIO(2), (0 = off,0v) or(1 = on,+V) **(Reserved:** Indicates BT connection) 3 = PIO(3), (0 = off,0v) or(1 = on,+V) (user definable): default input >1msec. interrupt to wake-up CPU out of deep sleep if enabled. 4 = PIO(4), (0 = off,0v) or(1 = on,+V) **Reserved** Triple Purpose: **(Resetting factory defaults on power up or dropping into command mode at anytime, auto connects to last paired or last connected device)** 5 = PIO(5), (0 = off,0v) or(1 = on,+V) **(Strobes 1/sec. for Slave indication or performing a Master inquiry)** 6 = PIO(6), (0 = off,0v) or(1 = on,+V) (user definable) 7 = PIO(7), (0 = off,0v) or(1 = on,+V) (user definable)

Store Parameters:
0 = Do not store
1 = Store Parameters in Flash

EXAMPLE:
TYPE : **ATSW23,3,1,1<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

24 **,<value>,<value>,<value>,<value> {Power Up Default Settings}**

Change power up responseType, security, audio, and filter modes

ATSW24,responseType, authMode, autoSCO, minorFilter //set defaults on power up

reponseType	authMode	autoSCO	minorFilter
0: Long Response	0: no authorization	0: no automatic SCO connect	0: no minor COD filter
1: Short Response	1: Authorization required	1: SCO connect upon radio connect	1: minor filter on COD
2: No response except for events: inquiry, etc.)			

Factory default is ATSW24,0,0,0,0

authMode = authentication security mode {56bit encryption is automatically enabled when "**1**" is selected and the default PIN = "**default**"} UART can reply LINK,BTaddress before or after the CONNECT,BTaddress is returned.

Default filter = 00000000 // no filter, finds all devices.

ATSW24,responseType, authMode, autoSCO, minorFilter // definitions
ATSW24,long response, no security,no SCO,no minor filter // settings

EXAMPLE:
TYPE : **ATSW24,0,0,0,0<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

Note: Requires a software or hardware reset for change to take affect
Set AT command Response in SHORT, LONG or NONE form. The default is LONG FORM
0 = LONG Form, 1 = SHORT FORM, 2 = NONE.

Verbalize Result Codes Summary Table:
Reference Appendix A in the back

0 = Long form	1 = Short form	2 = None	Explanation
OK	***00***		Command is correct and has been completed
CONNECT	***01***		Connection established
RING			Incoming ring detected
NO CARRIER	***03***		No connection or lost the carrier
ERROR	***04***		Bad command
NO ANSWER			See ATD . . .@. . .
SCO CONNECT	***09***		(Audio)SCO connection established
SCO FAILED			(Audio)SCO connection failed
SCO DISCONNECT	***11***		(Audio)SCO disconnected
DONE	***12***		Inquiry Complete

25 **,<value>,<value>,<value> {Auto Connecting Modes}**

Change power up Connect, Security, audio, and filter modes

ATSW25,connectMode, dataMode, unConnectUartMode //set defaults on power up

connectMode	*comMode*	*unConnectUartMode*	*Service*
0: Slave	0: *Fast Data	0: allow data to pass	0: SPP
1: Autoconnect Master	1: Data	1: ignore UART while unconnected ***	1: DUN
2: Idle Mode	2: Command		2: **Headset
3: Slave Undiscoverable			3: Audio Gateway
4: Autoconnect BNEP/PAN 10 second auto retry			

Factory default is ATSW25,0,1,0,0

If ***comMode = 0*** "Fast Data Mode" in a Master unit during a manual inquiry "ATDI" and/or connection request the radio connects in slow data mode not fast. This is because you will need the capability to issue commands because in fast data mode the AT command parser is turned off.

In idle mode the radio is neither, slave or master and draws 1.6mA of current but you can not communicate over the *Bluetooth* link in this state only through the TX & RX hardware UART.

You can not perform an Autoconnect and set service flag = to 1: DUN. The service selection for ATSMA,{slave address},UUID takes care of this for Master

** Headset service will auto answer on RING command from *Bluetooth* cell phone. You will have to change the COD from 00000000 to 00200404 for Headset.

Warning: The only way to communicate to the radio after setting the radio in "Fast Data Mode" and "ignore UART while unconnected" is to apply 3.3Vdc on PIO(4) during initial power up for 1 sec. These settings are used if you have no control over the source of streaming data into the radio, or you do not know when the radio has made a *Bluetooth* connection, and do not plan on sending any AT commands.

```
ATSW25,connectMode,dataMode,unConnectUartMode,Service      // definitions
ATSW25,slave,data mode,allow data to pass,SPP              // settings
```

EXAMPLE:
TYPE : **ATSW25,0,0,0,0<cr>** //Slave radio connects in fast data mode
REPLY: **<cr_lf>OK<cr_lf>**

Note: Requires a software or hardware reset for change to take affect

26 **,<value>,<value> {Lock User Definable Settings}**
,old PIN, Lock or unlock

Use this command to prevent unauthorized local & remote changing of settings.

```
ATSW26,PIN,LOCK or unlock            // where lock = 0 : unlocked, 1 : locked
```

EXAMPLE:
TYPE : **ATSW26,default,1<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

Note: PIN for this command is the same for authentication PIN = "default"
Factory default is unlocked user settings. This will also lock the password

	from changing.
27	**,<value> {Set LED Pulse Rate on PIO(5)}** **EXAMPLE:** *TYPE :* **ATSW27,1200<cr>** //1200 msec. factory default *REPLY:* **<cr_lf>OK<cr_lf>** Note: Integer decimal value can range from 1 msec. to 60,000 msec. Use to indicate slave mode operation and inquiry in process. Duty cycle remains 50%.
28	**,<iTimeout>,<mTimeout> {Inquiry Timeouts}** iTimeout = Inquiry Time (in seconds) mTimeout = Master connect request timeout (in seconds) **EXAMPLE:** *TYPE :* **ATSW28,20,60<cr>** // factory default *REPLY:* **<cr_lf>OK<cr_lf>** Note: Integer decimal value can range from 1 sec. to 60,000 sec. Does NOT require a reset for command to take affect.
29	**,PIN,<value> {Set PIN Lock Code}** Maximum alphanumeric characters (16) includes spaces. ***Caps sensitive.*** This command enables ATOP in the next section 0 – Normal operation this is disabled (factory default) 1 – Allow command through UART only 2 – Allow command through UART and over RF Link **EXAMPLE:** *TYPE :* **ATSW29,default,1<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**
30	**,<value> {Set Deep Sleep Mode}** 0 – Normal Operation never go into deep sleep (factory default) 1 – Go into deep sleep whenever possible ***(while idle, Page Scan or Sniff mode)*** **EXAMPLE:** *TYPE :* **ATSW30,1<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** • On bc02 the uart_rx line needs to be pulled high if not active before power is applied to the radio module. • If there is an active uart RF link the device will need Sniff Mode enabled to allow it to drop into sleep mode when there is no traffic. • When in deep sleep, the uart will miss the first character while waking up. Send a preamble byte to allow it to wake up and immediately thereafter send the AT Command or data in less than 1 second or the device will go back to deep sleep again. If you are using PIO(3) CPU interrupt a preamble byte is not needed. • No bytes are lost if sending commands down over the remote RF link side. This setting is stored in flash and does not require a reset to take affect. Takes approx. 1 second before the current will drop down to 90uA. Allow 5msec. for the CPU unit to come out of deep sleep.

2.1.4 Security

SP	**,<newvalue>,<oldvalue> {Set Personal Identification Number (PIN)}** Set PIN, Max alphanumeric characters (16) includes spaces. ***Caps sensitive*** Factory default PIN = "**default**" **EXAMPLE:** **ATSP,newPIN,oldPIN** *TYPE :* **ATSP,1234567890123456,default<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Note: Be careful when entering a new PIN. There is no way to obtain PIN *status after it is changed.* If the PIN is changed after two units have already authenticated and connected you will have perform a software or hardware reset for the devices to use the new PIN's if not the two units will still connect using the old stored PIN.
OP	**,PIN {Overwrite PIN}** Set Overwrite PIN so you do not need to enter old PIN to set a new PIN. Max alphanumeric characters (16) includes spaces. ***Caps sensitive*** **EXAMPLE:** *TYPE :* **ATOP,1234<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Note: This command is used in conjunction with ATSW29 command.

2.1.5 (COD) Class of Device

SC	**,<value>** Change Class of Device (COD) requires exactly 8, 16-bit hex values (0 thru F) based on the *Bluetooth* COD specification names published and maintained by the *Bluetooth* SIG. Factory default is 00000000 – undefined since this is set by the user based on the final OEM device installed in. **EXAMPLE:** *TYPE :* **ATSC,00020114<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Note: Requires a software or hardware reset for change to take affect

2.1.6 Set Service Name

SSN	**,<value>** Set the *Bluetooth* Service Name. Maximum of 16 alphanumeric characters. Factory default is COM0. **ATSSN,1234567890123456** **EXAMPLE:** *TYPE :* **ATSSN,COM0<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Note: Requires a software or hardware reset for change to take affect

2.1.7 Read Memory locations

SR	***,<n>*** ***Read an S register*** The S registers refer to memory locations used for configuration.
21	**,<PIO#>** 2 = PIO(3) (0 = off,0v) (1 = on,+V) **(Reserved: Indicates BT connection)** 3 = PIO(3) (0 = off,0v) (1 = on,+V) **(dual purpose, also user definable)** 4 = PIO(4) (0 = off,0v) (1 = on,+V) **(Reserved: Triple Purpose)** 5 = PIO(5) (0 = off,0v) (1 = on,+V) **(Strobes 1/sec. for Slave indication)** 6 = PIO(6) (0 = off,0v) (1 = on,+V) (user definable) 7 = PIO(7) (0 = off,0v) (1 = on,+V) (user definable) **EXAMPLE:** *TYPE :* **ATSR21,3<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>1<cr_lf>** or **<cr_lf>0<cr_lf>**

2.2 Inquiry/Connect/Disconnect Commands

2.2.1 Inquiry

This command is used to discover all *Bluetooth* radios (within range) that match the Class of Device (COD). If the COD is not known it is best to use 00000000 which allows discovery of all devices. You can not be in the default slave mode and perform the inquiry command. Only a Master or Radios in idle mode can perform an inquiry.

UCL	***{Clear Unit}, when not Bluetooth connected*** Recommend executing this before performing an inquiry command, places the radio in idle mode. See 3.4 Utilities

DI	***,<number>,<cod>*** ***Inquire*** This command signals the radio that the numbers, letters, and commas that follow are all part of a radio number that it should connect to. The number represents how many devices to find. An "**OK**" is returned immediately following this command. "**DONE**" will appear after all devices have been found, or a *20 second timeout* has occurred while searching for the number of devices specified. Returns the following: <bd_Address 12chars>,<cod 8chars>,<name up to 16chars> **EXAMPLE MASTER**: *TYPE :* **ATDI,1,00000000<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>00A0961F2023,00000000,BlueRadios<cr_lf>DONE<cr_lf>** *Note: The request for name is a separate command string from the other two so depending if the RF link is marginal the name may come back blank.*
IL	***{Repeat last inquiry performed}*** **EXAMPLE MASTER**: *TYPE :* **ATIL<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>00A0961F2023,00000000,BlueRadios<cr_lf>DONE<cr_lf>** **or** *REPLY:* **<cr_lf>ERROR<cr_lf>** // If inquiry not previously performed. Note: ATDI command string previously used is stored in Flash

2.2.2 Set Master Default *Bluetooth* Address/Profile for Slave

SMA	***,<bd address>,<profile>*** ***Set Master*** This command will set a specific Bluetooth Slave address and service profile into the Master device so on power up the Master will automatically search and connect to a unique Slave device in Fast data mode. The below example is for SPP profile service which is 1101. **EXAMPLE MASTER**: *TYPE :* **ATSMA,00A0961F904F,1101<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Reset module for change to take affect. Execute **ATSW25,1,0,0,0** "Master default . . ." before using the above command if still in slave mode.
ATMACLR	***{Clears stored Slave address in Master}*** **EXAMPLE**: *TYPE :* **ATMACLR<cr>**

	REPLY: **<cr_lf>OK<cr_lf>**

2.2.3 Connect as Master

This command is used to connect one radio module to another. Doing this will enable data transmission bi-directionally. When performing this command the reply is critical so as to understand where the connection process is. A connection can take several seconds, so when making a connection, if it is not already connected, an **"OK"** will be sent back immediately. Don't mistake this for a connection being complete. A completed connection will return **"CONNECT,123456789012"** some time after the command was sent typically <4 seconds. PIO(2) will go high and stay high or the Blue LED on the ***Blue*Radios** evaluation board will turn on and stay on while a *Bluetooth* connection is established.

DM	***,<BT_Address>,<UUID>*** ***{Dial as Master}*** This command gives the Slave address, and the type of profile that it will connect to/with. When connecting as a **"MASTER"** use valid *Bluetooth* Slave address and ending with a valid UUID from the table below. You must be in Master mode to connect to a remote Slave device. Using this command while still in Slave mode will not work. **EXAMPLE MASTER:** *TYPE :* **ATDM,112233445566,UUID<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>CONNECT,00A0961D9C37<cr_lf>** // example of a successful connection to a slave or **<cr_lf>NO ANSWER<cr_lf>** **or** **<cr_lf>LINK,00A0961D9C37<cr_lf>** // if security is enabled **<cr_lf>CONNECT,00A0961D9C37<cr_lf>**
DL	**{Dials Last}** Connects to last successful Slave Bluetooth address connection over SPP unless ATDM command was executed then the UUID from the ATDM command will be used. **EXAMPLE:** *TYPE :* **ATDL<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>CONNECT,123456789012<cr_lf>** **or** *REPLY:* **<cr_lf>?????<cr_lf>** *Note: Execute the command while in idle mode.* Connection information is stored permanently in flash. Factory reset will clear and set to back 000000000000 for Bluetooth address. To verify use the ATLAST command below.
LAST	Displays the **LAST** connected Bluetooth device address. **EXAMPLE:** *TYPE :* **ATLAST<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>000000000000<cr_lf>** **or** // Nothing stored *REPLY:* **<cr_lf>123456789012<cr_lf>** // Last connected BT address

UUID Table:

The Highlighted Universal Unique Identifiers (UUID's) have been tested. Others may be activated, but the results are unknown.

Profile Name	*UUID*
Serial Port (SPP)	1101
LAN Accessing PPP	1102
Dial-up Network (DUN)	1103
IrMC Sync	1104
OBEX Object Push	1105
OBEX File Transfer	1106
IrMC Sync Command	1107
Headset	1108
Cordless Telephone (CTP)	1109
Intercom	1110
Fax	1111
Audio Gateway	1112
WAP	1113
WAP_CLIENT	1114
BNEP/PAN (Client)	0000

2.2.4 Connect as Slave

This command is used to connect one radio module to another. Doing this will enable data transmission bi-directionally. When performing this command the reply is critical so as to understand where the connection process is. A connection can take several seconds, so when making a connection, if it is not already connected, an **"OK"** will be sent back immediately. Don't mistake this for a connection being complete. A completed connection will return **"CONNECT"** some time after the command was sent < 10 seconds typically 2 seconds. PIO(2) will go high and stay high or the Blue LED on the ***Blue*Radios** evaluation board will turn on and stay on while connected.

DS	***{Dial as Slave}*** This command places the Radio in Slave mode where it is waiting for a connection to occur from a Master. **EXAMPLE SLAVE:** *TYPE :* **ATDS<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>CARRIER<cr_lf>** or **<cr_lf>CONNECT,00A0961F008F<cr_lf>**

2.2.5 Disconnect

DH	***Dial {Hang up}*** This command will issue disconnect to the radio. **EXAMPLE:** *TYPE :* **ATDH<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>NO CARRIER<cr_lf>** Note: If you send this command over the RF link to a remote BlueRadios Slave the Slave will disconnect and go into Idle mode not Slave mode by design.

2.3 Command/Data Modes

1) Fast Data Mode:

The drawback to this is that once in fast data mode there is limited ways to get out of it like CPU reset, power down, or strobe PIO(4) on the radio. The advantage of this mode is that the data being sent does not have to be evaluated or processed for AT commands and will allow for a faster effective data throughput of around 200Kbps. To verify you are in fast data mode type **+++<cr>** it will pass directly through the UART because the AT Command parser is off. Else the **+++<cr>** will be accepted because the AT Command parser is still on and returns "**OK**". On the previous 4Mbit legacy modules the escape characters was fixed and set to "ATMC", is no longer implemented.

2) The different operation modes are explained in the next table below:

The Command Mode or Slow Data Mode will slow down the throughput when the radio looks for the AT commands. One way to allow minimal overhead transmission burden is to perform all configuration commands, and then place the radio into the fast data mode.

+++	***{Put Radio into Command Mode when Bluetooth connected}*** This sequence is used to force the radio into command mode state. If the Radio has been place in Fast Data Mode this command will have no affect and the typed command will be treated as data. When using this command allow at least **100ms** delay before sending next AT command. If successful an "**OK**" is returned, or nothing will be returned if already in data or fast data mode, and connected. A "**NO CARRIER**" occurs when the Bluetooth connection has been lost. **EXAMPLE:** *TYPE :* **+++<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>NO CARRIER<cr_lf>**
SESC	***,<nn>* {Set Escape ASCII Character} *where integer nn , is the ASCII decimal character <256 or non extended ascii characters.*** **EXAMPLE:** *TYPE :* **ATSESC,43<cr>** // 43 = 2B(hex) = "+" ASCII character *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>ERROR<cr_lf>** Do not need to perform reset and stores permanently.
MD	***{Put Radio into Data Mode}*** This sequence is used to force the radio into Data Mode. In order for this to have an effect the Radio has to be CONNECTED.

	If successful an **"OK"** is returned else a **"NO CARRIER".** Also a **"NO CARRIER"** occurs when the connection has been lost. **EXAMPLE:** *TYPE :* **ATMD<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>NO CARRIER<cr_lf>**
MF	***{Put Radio into Fast Data Mode}*** This sequence is used to force the radio into fast data mode. Once in Fast Data Mode all commands are treated as data, ways to get out of this mode it to reset power on the radio, or strobe PIO(4) while connected. If successful an **"OK"** is returned, or nothing will be returned if already in fast data mode and connected else **"No Carrier"** if the Bluetooth RF link is not established. **EXAMPLE:** *TYPE :* **ATMF<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** // if connected command is accepted **or** **<cr_lf>OK<cr_lf>** **<cr_lf>NO CARRIER<cr_lf>** // not connected command is rejected

Note: Byte Gaps and Data Latency – The way *Bluetooth* is designed and operates random byte gaps of 5 msec to 20 mesec are common. Packet size will vary from transmission to transmission. The faster the UART speed the smaller the byte gap delay.

Effective data payload throughput in fast streaming mode is approximately 200Kbps and 60Kbps in regular data mode when the AT parser looks at each character for ASCII valid command scripts in the data stream. The radio RX has very limited buffering so if you do not use hardware flow control and are transmitting further distances you will quickly overflow the buffer because of RF retransmissions, etc.

When a *Bluetooth* connection is made the radio modem goes into regular data mode per the power-up factory default settings. This enables the user to remotely configure the radio settings via a remote RF *Bluetooth* connection. Basically you can setup the radio so no commands are required to be sent from the embedded side of the radios UART. This will prevent any software embedded firmware development or testing for legacy systems.

2.4 Utilities

2.4.1 Cancel Command

UCL	***{Cancel Command}*** The UCL command tells the radio to cancel inquiry or connect requests commands and then places the radio in Idle Mode. This command can come in handy for a quick exit from commands like inquiry mode if there are no devices in the area and you do not want to wait 60 seconds for an automatic timeout. You can not cancel a command while RF connected. **EXAMPLE:** *TYPE :* **ATUCL<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **or** **<cr_lf>ERROR<cr_lf>**

2.4.2 Reset

URST	***{Reset}*** The RST command tells the radio to perform software reset on the CPU. **EXAMPLE:** *TYPE :* **ATURST<cr>** *REPLY:* **This unique Command does not reply with "OK" or "ERROR" because of internal UART data processing limitations and response timing.** ***Blue***Radios evaluation board has a convenient manual pushbutton software reset switch on the PCB labeled SW1.
FRST	***{Factory Reset}*** **EXAMPLE:** *TYPE :* **ATFRST<cr>** *REPLY:* **This unique Command does not reply with "OK" or "ERROR" because of internal UART data processing limitations and response timing.** ***Blue***Radios evaluation board has a convenient manual pushbutton software reset switch on the PCB labeled PIO(4). Resets the radio back to factory defaults. Allow 5 seconds for the Radio to read and write to FLASH. ***Blue***Radios evaluation board has a convenient manual pushbutton factory reset switch on the PCB labeled PIO4 as well. ***Note:*** You can send either above reset commands though the UART or over Bluetooth RF connection.
ATSSW,	***0,<index> {Sets bypass for hardware factory configuration reset PIO(4)}*** 1 = bypass 0 = PIO(4) factory reset enabled **EXAMPLE:** *TYPE :* **ATRSW,0,1<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Use this command in replace of physically connecting PIO(4) to ground to prevent and inadvertent factory configuration reset.
ATRSW,	***0 {read above setting}*** **EXAMPLE:** *TYPE :* **ATRSW,0<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>0<cr_lf>**

2.4.3 Variable Storage

STORE	***,<index>,<string) {Store registers for users}*** The Store command is permanent storage for the user for ID's, address, etc. 0 : Location 0 Up to 16 alphanumeric characters 1 : Location 1 Up to 16 alphanumeric characters **EXAMPLE:** *TYPE :* **ATSTORE,0,1234567890123456<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>ERROR<cr_lf>**
READ	***,<index> {Read contents from ATSTORE index}*** The READ from index locations:

	0 : Location 0 Up to 16 alphanumeric characters 1 : Location 1 Up to 16 alphanumeric characters **EXAMPLE:** *TYPE :* **ATREAD,0<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>1234567890123456<cr_lf> or** **<cr_lf>OK<cr_lf><cr_lf>OK<cr_lf>** ***Note:*** Nothing stored, will return blank

2.4.4 RSSI and Link Quality

RSSI	***{Returns RSSI value}*** This command is used to obtain the RSSI value for a given open connection. This is a parameter associated with the ACL connection to a peer device. Hex character typical value from -10 to +31 in integer dB value. **EXAMPLE:** *TYPE :* **ATRRSI<cr>** *REPLY:* **<cr_lf>OK<cr_lf> or <cr_lf>NO CARRIER<cr_lf>** **<cr_lf>+00<cr_lf>** ***Note:*** The RSSI value will be +00 if the signal is within the Golden Range. The Golden Range min and max value is 1 and 12 respectively for the modules. *This value is the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range (see below for definition). Any positive RSSI value returned by the Host side indicates how many dB the RSSI is above the upper limit. Any negative value indicates how many dB the RSSI is below the lower limit. A value of zero indicates that the RSSI is inside the Golden Receive Power Range. Note: how accurate the dB values will be depends on the Bluetooth hardware. The only requirements for the hardware are that the Bluetooth device is able to tell whether the RSSI is inside, above, or below the Golden Device Power Range.*
LQ	***{Returns Link Quality}*** Hex value from 0 to 255 decimal which is the measure of Bit Error Rate (BER) **EXAMPLE:** *TYPE :* **ATLQ<cr>** *REPLY:* **<cr_lf>OK<cr_lf> or <cr_lf>NO CARRIER<cr_lf>** **<cr_lf>250<cr_lf>** *Link_Quality is a value from 0-255, which represents the quality of the link between two Bluetooth devices. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality. In the case for CSR, this value is a measure of BER (Bit Error Rate).*

GOLDEN RECEIVE POWER RANGE

The lower threshold level of the Golden Receive Power Range corresponds to a receive power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/-6 dB.

2.4.5 Audio (SCO)

DSCO	***{Dial SCO}*** The DSCO tells the radio to dial and connect audio channel. **EXAMPLE:** *TYPE :* **ATDSCO<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>SCO CONNECT<cr_lf>**
DHSCO	***{Dial Hang up SCO}*** The DHSCO tells the radio disconnect audio channel. **EXAMPLE:** *TYPE :* **ATDHSCO<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>SCO DISCONNECT<cr_lf>**

2.4.6 Park and Sniff (Connected Slave)

PARK	**,<integer value>,<integer value> {Park}** Master and Slave ATPARK,maxIntval,minIntval **EXAMPLE:** *TYPE :* **ATPARK,1000,11<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Allow 7 seconds for the slave to drop its average current draw from 45mA to 3mA and go into PARK. Any UART traffic will cause the Slave radio go back to full operation within 5msec of the first character for 7 seconds before going back into PARK mode without subsequent data. The Slave will remain connected to the Master because the BlueRadios Module only supports one connection. Note: Both units are required to support park. Park request is sent from both Master and Slave to go into affect only during an active Bluetooth connection. The order does not matter.
XPARK	**{Exit Park Mode}** Initiated from Master to Remote Slave The ATXPARK command tells the radio to exit the park mode. **EXAMPLE:** *TYPE :* **ATXPARK<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Unit will immediately exit park.
SNIFF	**,<integer value>,<integer value>,<integer value>,<integer value> {Sniff}** for Slave device connected in time slots (N). Time = N * 0.625msec, ATSNIFF,maxIntval,minIntval,attempt,timeout **EXAMPLE:** *TYPE :* **ATSNIFF,1600,160,10,160<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Manually enable sniff results in an always connected slave using only 2mA average current when no data is sent. *Takes about 7 seconds before the*

	current drops after the connection is established. Parameters are lost after connection is dropped.
SSNIFF	**,<integer value>,<integer value>,<integer value>,<integer value> {Set/Store Sniff}** for Slave device connected in time slots (N). Time = N * 0.625msec, ATSSNIFF,maxIntval,minIntval,attempt,timeout **EXAMPLE:** *TYPE :* **ATSSNIFF,1600,160,10,160<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** *Stores parameters permanently in Flash and auto starts Sniff after connection is established.*
CSNIFF	**{Clear Sniff stored values above}** **EXAMPLE:** *TYPE :* **ATCSNIFF<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**
XSNIFF	**{Exit Sniff mode}** The ATXSNIFF tells the radio to exit and stop "sniffing" RF signals. **EXAMPLE:** *TYPE :* **ATXSNIFF<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**

2.4.7 Pairing

PAIR	**,<BdAddr>** The command tells the radio in Master mode to Pair to a specific Bluetooth address. **EXAMPLE:** *TYPE :* **ATPAIR,00A0961F008F<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** or **<cr_lf>NO ANSWER<cr_lf>** or **<cr_lf>PAIRED,FAILED<cr_lf>** or **<cr_lf>PAIRED,123456789012<cr_lf>** // Successful *Note: Paring Timeout is 30 seconds. Need to be in Idle mode first. Security PINs are exchanged. The Slaves radio UART will output "LINK,BTaddress" for every secured connection. Also the BlueRadios Masters UART will output the same if security flag is enabled.*
UPAIR	**,<index> {Un Pair by index number}** The command tells the radio to Un Pair from the Bluetooth stored address in index locations 00, 01, 02, and 03. **EXAMPLE:** *TYPE :* **ATUPAIR,00<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** Use the command ATLAST to view the stored address after pairing. Note: paring is not the same as a connection so you will not see inquiry or connect indication. Can be in Master, Slave, or Idle mode to un pair. Will

	need to reset or cycle power to clear paired address.
UPAIRB	**,<bdAddr> {Un Pair by Bluetooth address}** **EXAMPLE:** *TYPE :* **ATUPAIR,123456789012<cr>** *REPLY:* **<cr_lf>OK<cr_lf> or <cr_lf>ERROR<cr_lf>**

2.4.8 Max TX Power

SPF	***,<value>,<sign>* {Set Max TX Power Level}** **First value = integer from 20 to 0** **Second value (sign) = (+) or (-)** This command sets the Radios maximum transmit power in (dBm) (15, 12, 4, and 0 dBm) recommended values. Factory default is 15dBm max for Class1 and 4 dBm for Class2 devices. Class 1 Power Table / Class2 Power Table 15 dBm / 4 dBm 11 dBm / 0 dBm 7 dBm / -4 dBm 3 dBm / -8 dBm -1 dBm / -12 dBm -5 dBm / -16 dBm -9 dBm / -20 dBm 0 dBm to +20 dBm Class1 -6 dBm to +4 dBm Class2 up to 0 dBm Class3 Note: The default value is 4dBm in a class2 BlueRadios module the max performance is still 4dBm output gain for the class2 radio if set above this. This value does not include gains associated with the external antenna (2 dBm). The firmware uses the highest value in the power table that is less than or equal to the requested max transmit power number above. **EXAMPLE** *TYPE :* **ATSPF,4,+<cr>** // +4dBm *REPLY:* **<cr_lf>OK<cr_lf>** or if > (+15) or < (-20) **<cr_lf>EROOR<cr_lf>** Note: Does not require a reset for change to take affect.

2.4.9 Link Supervisory Timeout

LSTO	***,<integer value>, {Link Supervisory Timeout}*** The command tells the radio to drop the connection if the units can not handshake for X amount of time in seconds. Factory default is ~4 seconds. Integer value from 2 to 41 **EXAMPLE** *TYPE :* **ATLSTO,20<cr>** // Sets timeout to ~20 seconds *REPLY:* **<cr_lf>OK<cr_lf> or <cr_lf>ERROR<cr_lf>**

	Note: If timeout is set too short of a duration you may inadvertently drop the connection prematurely if the RF link margin is poor.

BlueRadios Class1 Module Power Consumptions:

Deep Sleep Mode enabled = 90uA (Idle Mode) or if in (Sniff Mode), will go momentarily as low as 90uA in between processes.

Idle Mode ≈ 1.6mA average, neither Slave or Master
Slave connected ≈ 25mA average
or
Use the command ATSNIFF,1600,160,10,160 results ≈ 2mA average, slave connected when no data sent.

Slave not connected ≈ 45mA average (Factory default)
or
ATSW21,4096,18,4096,18 settings will result in the Slave not connected ≈ 1mA average

This can even go as low as 0.6mA if you make the Slave not discoverable but connectable.

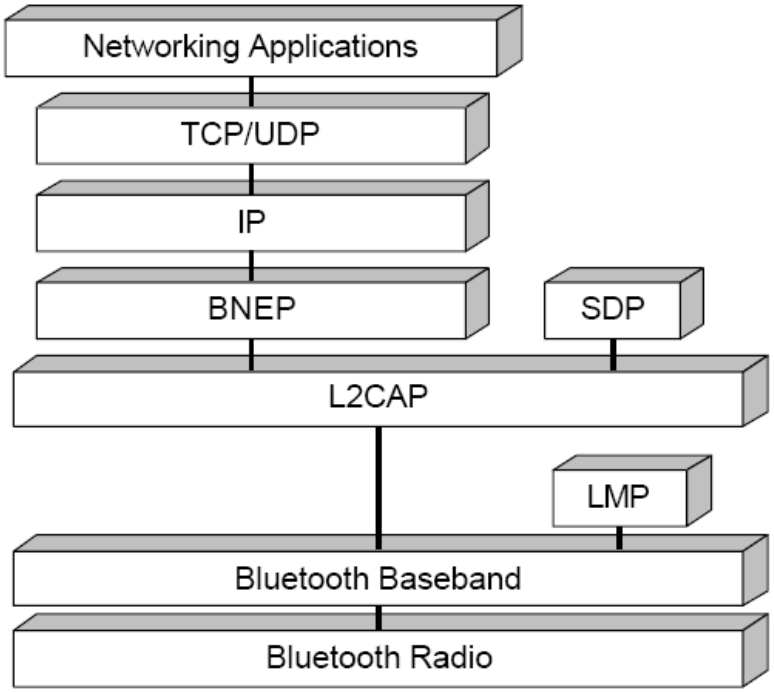
Master not connected ≈ 1.6mA. However, a Master radio not in use it is better to just turn off the power completely to the radio and draw zero current (0mA).
Master inquiry ≈ 60mA for the first couple of seconds before it connects to the Slave
Master Connected ≈ 25mA? (too difficult to measure)

If you perform a remote *Bluetooth* RF "ATDH" disconnect command from any Master to the BlueRadios Slave radio, the Slave radio will go into idle mode. You will have to either send a command to the Slave radio through its local UART or cycle power on the radios to have it come back up in Slave mode. This was designed as if you were communicating to the local UART on the Slave radio. Typically a Slave never sends a commanded disconnect only the Master does. In this mode you can devise some clever power saving features like leaving the Slave in idle mode until another event triggers the radio to go back into Slave discoverable mode. It is important to remember a device in Idle Mode is not discoverable which has some security advantages. The same sort of benefits can be achieved by just controlling when the radio is turned off/on again.

3 PAN Client {BNEP and IP}

PAN – Personnel Area Networking Profile (client side only). Directly connect to an IP address on the Internet without embedding a TCP/IP stack on your application. Using PAN profile is a two step process. 1) Open a BNEP session and 2) Dial IP address and port. Negotiating the DHCP session takes approx. 10 seconds, once this is established the IP connect is less than a second.

Stack Overview



The use of the BNEP for transporting an Ethernet packet is shown in the above Figure. BNEP removes and replaces the Ethernet Header with the BNEP Header. The Ethernet Payload remains unchanged. Finally, both the BNEP Header and the Ethernet Payload is encapsulated by L2CAP and is sent over the Bluetooth media. The maximum payload that BNEP will accept from the higher layer is equal to the negotiated L2CAP MTU (minimum value: 1691), minus 191 (reserved for BNEP headers). This way it can be assured that enough frame buffer space is reserved to transmit all BNEP.

DBNEP	***,<BT_Address>,<n> {Dial BNEP}*** ***1 = DHCP*** ***0 = no DHCP*** This command dials any Bluetooth PAN AP Server box or cell phone. When connecting the BlueRadios as a **"MASTER"**, make sure the radio is in idle mode atucl<cr>. You must be in Master mode to connect to a Remote Slave PAN Server device. Using this command while still in Slave mode will not work. **EXAMPLE MASTER:** *TYPE :* **ATDBNEP,112233445566,1<cr>** //Dial BNEP Request with DHCP

	REPLY: **<cr_lf>OK<cr_lf> or <cr_lf>BNEP:FAILED<cr_lf>** **<cr_lf>BNEP:OPEN,00A0961D9C37<cr_lf>** //BNEP Open to AP **<cr_lf>LINK,00A0961D9C37<cr_lf>** // Security PIN requested from Server **<cr_lf>DHCP:DONE<cr_lf> or <cr_lf>DHCP:FAILED<cr_lf>** Note: Make sure you set the PIN in the radio to match what the AP requires. If the Server request security the LINK reply comes after the BNEP reply. If the BlueRadios unit requests security the LINK reply happens first. If power is not cycled on the device and AP the link keys are no longer exchanged on future connections. After BNEP radio is in command mode.
HBNEP	***{Hang-Up BNEP}*** **EXAMPLE MASTER:** *TYPE :* **ATHBNEP<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>BNEP:CLOSED<cr_lf>**
DIP	***,<IP_Address>,<port> {Dial IP v4 Address}*** **EXAMPLE MASTER:** *TYPE :* **ATDIP,84a30465,13<cr>** // IP is in Hex format "132.163.4.101" *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>TCP:CONNECT,84a30465<cr_lf>** Note: PIO(2) will go high indicating you can TX/RX data to remote IP Web Server. If the above BNEP connection is lost PIO(2) will go low after the AP lost Bluetooth connection timeout is reached ~20 seconds and reply BNEP:CLOSED. The above IP address automatically connects sends current date and time and disconnects from the UTC (NIST) time server.
HIP	***{Hang Up IP Connection}*** **EXAMPLE:** *TYPE :* **ATHIP<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>TCP:DISCONNECT<cr_lf>**
AIP	***{Abort IP Connection}*** **EXAMPLE:** *TYPE :* **ATAIP<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>TCP:DISCONNECT<cr_lf>**
ATLTCP	***{Listen TCP}*** The BlueRadios device will listen for a TCP Request from Remote Web Server. **EXAMPLE:** *TYPE :* **ATLTCP<cr>** *REPLY:* **<cr_lf>OK<cr_lf>**
ATSIP	***,<IP Address(hex)>,<port#)> {Set IP} in radio for static or pre-assigned IP v4 address*** **EXAMPLE:** *TYPE :* **ATSIP,c0a80202,13<cr>** // IP is in Hex format. IP 192.168.2.2

	REPLY: **<cr_lf>OK<cr_lf>** Note: Port number (decimal) up to 4 digits.
SPAN	***{Status PAN} for BNEP and IP.*** This command returns x,y where x and y are 0 or 1 x: bnep open = 1 closed = 0 and y: ip connected = 1 disconnected = 0 **x,y** 0,0 BNEP closed, IP disconnected. Normal factory default state. 1,0 BNEP open, IP disconnected. 1,1 BNEP open, IP connected. Only state you can pass data. **EXAMPLE:** *TYPE :* **ATSPAN<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>0,0<cr_lf>** // BNEP Closed, and IP disconnected. Note: You need to be in command mode to execute this command
SSW	***,<integer>,<integer> {set switch #2, connect IP}*** **EXAMPLE:** *TYPE :* **ATSSW,2,1<cr>** // auto connect IP (ON) *REPLY:* **<cr_lf>OK<cr_lf>** **EXAMPLE:** *TYPE :* **ATSSW,2,0<cr>** // auto connect IP (OFF) factory default *REPLY:* **<cr_lf>OK<cr_lf>**
RSW	***,<integer> {read switch #2, connect IP Setting}*** **EXAMPLE:** *TYPE :* **ATRSW,2<cr>** *REPLY:* **<cr_lf>OK<cr_lf>** **<cr_lf>0<cr_lf>** // no auto connect IP

4 *Blue*Radios Factory Default Power up Settings

AT command Response Form = Long Form
Bluetooth Service Profile = Serial Port Profile (SPP)
Device Role = Slave
Baud Rate = 9600bps
Data bits = 8 bits
Parity = None
Stop bits = 1 bit
Hardware flow control RTS/CTS = Enabled
Power Mode = Never go into deep sleep mode
Country Code = North America and Europe
Module Type = BlueRadios AT
Name of Device (local name) = BlueRadios
My radio status = 1,0 {slave disconnected}
Service Name = COM0
Power up default (ATSW24) settings = 0,0,0,0 {if security flag is enabled a factory reset of parameters does not disable security}
Power up default (ATSW25) settings = 0,1,0,0
Lock user definable settings (ATSW26) = False
LED pulses on/off (ATSW27) = 1200msec
Set password lock bit (ATSW29) = 00 (HEX)
Deep sleep enabled (ATSW30) = False
    Radio will go back to deep sleep if enabled after 1 second of no activity on UART or PIO(3)
Major & Minor Class Of Device (COD) = 00000000 {undefined}
Security PIN and Encryption Disabled
    Default PIN = "default" *caps sensitive*
Page Scan Interval = 0x400 {1024 time slots}
Page Scan Window = 0x200 {512 time slots}
Inquiry Scan Interval = 0x400 {1024 time slots}
Inquiry Scan Window = 0x200 {512 time slots}
Timeout Connection Parameters
    ATDI Dial Inquiry timeout = 20 seconds
    ATDM Master Connect Request = 60 seconds
    (need to perform **ATUCL** to cancel last command to halt sooner or set ATSW28 to a different value)
Pairing timeout = 30 seconds
Timeout for lost of *Bluetooth* connection = 4 seconds (Link Supervisory Timeout)
Auto connect IP (OFF)
Class1 Radio Max transmit power = 15dbm, Class2 module will still have a max of 4dbm output performance even though you can set it to 15.

Note: Apply 3.3Vdc on PIO(4) for 1 sec. during initial power, up will change any of these back that are variables to the factory settings above. The other option is to use the software command **ATFRST<cr>**. The only exception for these two options is the name of device (friendly name) BlueRadios and PIN will not change back if you had changed this already. Allow approximately 5 seconds for the radio to reconfigure. On the BlueRadios evaluation boards we included a push button for PIO(4) factory reset which can be held down for 1 second while power is applied to the radio.

Misc. items:
Over the air RF guaranteed data packet retries is set for indefinite.

5 Example of a Master Discovery/Connection Sequence

From power up and no connection:

1) Perform an Inquiry to obtain ***BT_Address's*** (unless it is already known).
   Sent : ATUCL**<cr>** // Clears radio state and places in Idle Mode
   Reply:<cr_lf>OK<cr_lf>
   Sent : ATDI,1,00000000 {Class of Device}**<cr>** // Looks for only one Bluetooth device
   Reply:<cr_lf>00A0961F2023,00000104,BLUERADIOS<cr_lf>
   <cr_lf>DONE<cr_lf>

2) Perform a Master Connect over SPP using the ***BT_Address.***
   Sent : ATDM, 00A0961F2023,1101**<cr>** // SPP connection
   Reply:<cr_lf>CONNECT**,**00A0961F008F <cr_lf> // Returns Slave BT address radios is in Data Mode

3) Place radio into Fast Data Mode.
   Sent : ATMF**<cr>** // Places radio in Fast Data Mode
   Reply:<cr_lf>OK<cr_lf>

4) Send Data.

Note: When sending commands from the Slave when the Slave connects in Fast Data Mode (ATSW25/or issuing ATMF). All valid AT commands are sent through the Slaves UART will be interpreted and responded by the Master radio as if it was the local Slave radio. Basically in this configuration from the Slave end you can obtain status and configure from the remote Master radio. This is a unique feature that may be useful in some applications but can confuse the user if you think you are talking to the local Slave UART.

To get out of Data Mode and check status:

1) Delay at least 50 milliseconds; this could be less or more.
2) Get into Command Mode.
   Sent : +++**<cr>** // Default escape sequence of characters
   Reply:<cr_lf>OK<cr_lf>
3) Check Status, perform a Disconnect …
   Sent : AT**<cr>**
   Reply:<cr_lf>OK<cr_lf>
4) Or send any AT Command example:
   Sent : ATSI,0**<cr>**
   Reply:<cr_lf>BlueRadios AT<cr_lf>

6 Example of a Slave Command Sequence

From power up:

1) Perform an inquiry and search for Slave Bluetooth device with a PC or other embedded unit.
2) Send a connection request from PC or embedded device to the Slave.
3) Wait for a connection
   Reply:<cr_lf>CONNECT,{SLAVE ADDRESS}<cr_lf> // SPP Connected
4) Send Data.

Note: This command sequence assumes the radio is in factory default in which it automatically comes up and is connectable as a Slave from a Master request.

To get out of Data Mode and check status:

1) Delay at least 50 milliseconds; this could be less or more.
2) Check Status, perform a Disconnect …etc.
   Sent : AT<cr>
   Reply:<cr_lf>OK<cr_lf>

7 Example of Audio and Data Connection (From Master to Remote Slave)

Master Radio

Sent : ATSW25,1,1,0,0<cr> // Set ConnectMode Master & data mode
Reply:<cr_lf>OK<cr_lf>
Perform hardware or software reset for the above changes to take effect on Master radio
Sent : ATDM,{SLAVE ADDRESS},1101<cr> // Serial Port Profile
Reply:<cr_lf>OK<cr_lf>
Reply:<cr_lf>CONNECT,{SLAVE ADDRESS}<cr_lf> // SPP Connected
"You can now send data back and fourth between the two units"
Sent : +++<cr> // Default escape sequence of characters
Reply:<cr_lf>OK<cr_lf>
Sent : ATDSCO<cr_lf> // Dial Audio SCO
Reply:<cr_lf>SCO CONNECT<cr_lf> // Audio channel connected
Sent : +++<cr>
Reply:<cr_lf>OK<cr_lf>
Sent : ATMD<cr> // Place back into Data mode
Reply:<cr_lf>OK<cr_lf>
"You can now send data over SPP and simultaneously talk over the SCO audio channel"
Sent : ATDH<cr> // Hang up Audio SCO & SPP
Reply:<cr_lf>OK<cr_lf>
Reply:<cr_lf>SCO DISCONNECT<cr_lf> // audio channel disconnected
Reply:<cr_lf>NO CARRIER<cr_lf> // SPP disconnected

Using ATDSCO will disconnect audio channel but you will need to type ATMD to go back into data mode to send characters

8 Acronyms/Abbreviations

API - Application Protocol Interface
AT - Automatic Tone
ASCII - American Standard Code for Information Interchange
BNEP - Bluetooth Networking Encapsulation Protocol
BR - BlueRadios
BT - Bluetooth
BTW - Bluetooth Windows Stack
COD - Class Of Device
COM - Communications
CR - Carriage Return
CTS - Clear To Send
DSR - Data Sent Receive
GND - Ground
IP – Internet Protocol
LF - Line Feed
MCU - Microcontroller Unit
MISO -
MOSI -
NC - Not Connected
PC - Personal Computer
PCB - Printed Circuit Board
PAN - Personnel Area Networking
RF - Radio Frequenc
PIO - Pin Input/Output
RST - Reset
RTS - Ready To Send
RX - Receive
SCO - Synchronous Connection-Oriented: the links used by BT to send voice.
SMT - Surface Mount Technology
SPI - Serial Protocol Interface
SPICK - SPI Clock
SPICS - SPI Chip Select
TTL - Transistor Transistor Logic
TX - Transmit
UART - Universal Asynchronous Receiver/Transmitter
USB - Universal Serial Bus
UUID - Universal Unique Identifier – maintain by Bluetooth SIG.
VCC - DC Power
VDD - DC Power

Go to www.blueradios.com on the left hand column to download *Bluetooth* Glossary of Terms PDF and other materials.

Appendix A: Verbalization Responses

Preliminary

No.	Command	Verbalization Long Response	Verbalization Short Response
1	AT	<cr_lf>OK<cr_lf>	<cr_lf>20,00<cr_lf>
2	ATVER,ver1	<cr_lf>OK<cr_lf><cr_lf>Ver 2.8.0.0.12<cr_lf>	<cr_lf>52,00<cr_lf>Ver 2.8.1.4.0<cr_lf>
3	ATSI,0	<cr_lf>OK<cr_lf><cr_lf>BlueRadios AT<cr_lf>	<cr_lf>16,BlueRadios AT<cr_lf>
4	ATSI,1	<cr_lf>OK<cr_lf><cr_lf>00A09608F513<cr_lf>	<cr_lf>14,00A09608F513<cr_lf>
5	ATSI,2	<cr_lf>OK<cr_lf><cr_lf>BlueRadios<cr_lf>	<cr_lf>53,10,BlueRadios<cr_lf>
6	ATSI,3	<cr_lf>OK<cr_lf><cr_lf>1,0<cr_lf>	<cr_lf>17,1,0 <cr_lf>
7	ATSI,4	<cr_lf>OK<cr_lf><cr_lf>COM0<cr_lf>	<cr_lf>18,COM0<cr_lf>
8	ATSI,5	<cr_lf>OK<cr_lf><cr_lf>00000000<cr_lf>	<cr_lf>19,00000000<cr_lf>
9	ATSI,6	<cr_lf>OK<cr_lf><cr_lf>0,0,0,0<cr_lf>	<cr_lf>20,0,0,0,0<cr_lf>
10	ATSI,7	<cr_lf>OK<cr_lf><cr_lf>0,1,0,0<cr_lf>	<cr_lf>21,0,1,0,0<cr_lf>
11	ATSI,8	<cr_lf>OK<cr_lf><cr_lf>0027<cr_lf>	<cr_lf>22,0027<cr_lf>
12	ATSI,9	<cr_lf>OK<cr_lf><cr_lf>Not Set!<cr_lf>	<cr_lf>23,NOT SET!<cr_lf>
13	ATSI,10	<cr_lf>OK<cr_lf><cr_lf>0400, 0400, 0200<cr_lf>	<cr_lf>24,0400, 0400, 0200<cr_lf>
14	ATSI,11	<cr_lf>OK<cr_lf><cr_lf>04B0<cr_lf>	<cr_lf>25,04B0<cr_lf>
15	ATSI,12	<cr_lf>OK<cr_lf><cr_lf>2B<cr_lf>	<cr_lf>68,2B<cr_lf>
16	ATSI,13	<cr_lf>OK<cr_lf><cr_lf>003C,003C<cr_lf>	<cr_lf>69,003C,003C<cr_lf>
17	ATSI,14	<cr_lf>OK<cr_lf><cr_lf>default<cr_lf>	<cr_lf>71,default<cr_lf>
18	ATSI,15	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>73,00<cr_lf>
19	ATSI,16	<cr_lf>OK<cr_lf><cr_lf>00<cr_lf>	<cr_lf>76,00<cr_lf>
20	ATSI,17	<cr_lf>OK<cr_lf>	<cr_lf>92,Not Set!<cr_lf>
21	ATSI,18	<cr_lf>OK<cr_lf>	<cr_lf>93,04<cr_lf>
22	ATSI,19	<cr_lf>OK<cr_lf>	<cr_lf>94,00,BDaddrs<cr_lf> 01,BDaddrs<cr_lf> 02,BDaddrs<cr_lf> 03,BDaddrs<cr_lf>
23	ATSI,20	<cr_lf>OK<cr_lf>	
24	ATSW20,5,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
25	ATSW20,10,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
26	ATSW20,20,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
27	ATSW20,39,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
28	ATSW20,79,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
29	ATSW20,157,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
30	ATSW20,236,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
31	ATSW20,472,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
32	ATSW20,944,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
33	ATSW20,1887,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
34	ATSW20,3775,0	<cr_lf>OK<cr_lf>	<cr_lf>45,00<cr_lf>
35	ATSW21,X,X,X,X	<cr_lf>OK<cr_lf>	<cr_lf>47,00<cr_lf>
36	ATSW22,2	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
37	ATSW22,3	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
38	ATSW22,4	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
39	ATSW22,5	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
40	ATSW22,6	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>
41	ATSW22,7	<cr_lf>OK<cr_lf>	<cr_lf>48,00<cr_lf>

42	ATSW23,3,1,0	<cr_lf>OK<cr_lf>	<cr_lf>49,00<cr_lf>
43	ATSW23,6,1,0	<cr_lf>OK<cr_lf>	<cr_lf>49,00<cr_lf>
44	ATSW24,0,0,0,0	<cr_lf>OK<cr_lf>	
45	ATSW24,0,0,0,1	<cr_lf>OK<cr_lf>	
46	ATSW24,0,0,1,0	<cr_lf>OK<cr_lf>	
47	ATSW24,0,0,1,1	<cr_lf>OK<cr_lf>	
48	ATSW24,0,1,0,0	<cr_lf>OK<cr_lf>	
49	ATSW24,0,1,0,1	<cr_lf>OK<cr_lf>	
50	ATSW24,0,1,1,0	<cr_lf>OK<cr_lf>	
51	ATSW24,0,1,1,1	<cr_lf>OK<cr_lf>	
52	ATSW24,1,0,0,0	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
53	ATSW24,1,0,0,1	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
54	ATSW24,1,0,1,0	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
55	ATSW24,1,0,1,1	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
56	ATSW24,1,1,0,0	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
57	ATSW24,1,1,0,1	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
58	ATSW24,1,1,1,0	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
59	ATSW24,1,1,1,1	<cr_lf>OK<cr_lf>	<cr_lf>34,00<cr_lf>
60	ATSW25,0,0,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
61	ATSW25,0,0,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
62	ATSW25,0,0,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
63	ATSW25,0,0,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
64	ATSW25,0,1,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
65	ATSW25,0,1,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
66	ATSW25,0,1,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
67	ATSW25,0,1,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
68	ATSW25,1,0,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
69	ATSW25,1,0,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
70	ATSW25,1,0,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
71	ATSW25,1,0,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
72	ATSW25,1,1,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
73	ATSW25,1,1,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
74	ATSW25,1,1,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
75	ATSW25,1,1,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
76	ATSW25,0,2,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
77	ATSW25,0,2,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
78	ATSW25,0,2,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
79	ATSW25,0,2,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
80	ATSW25,1,2,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
81	ATSW25,1,2,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
82	ATSW25,1,2,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
83	ATSW25,1,2,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
84	ATSW25,2,2,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
86	ATSW25,2,2,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
87	ATSW25,2,2,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
88	ATSW25,2,2,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
89	ATSW25,2,0,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
90	ATSW25,2,0,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
91	ATSW25,2,0,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
92	ATSW25,2,0,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
93	ATSW25,2,1,0,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
94	ATSW25,2,1,0,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>

95	ATSW25,2,1,1,0	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
96	ATSW25,2,1,1,1	<cr_lf>OK<cr_lf>	<cr_lf>35,00<cr_lf>
97	ATSW25,4,0,0,0		
98	ATSW25,4,1,0,0		
99	ATSW25,4,0,1,0		
100	ATSW25,4,0,0,0		
101	ATSW25,4,2,0,0		
102	ATSW25,4,2,1,0		
103	ATSW26,default,1	<cr_lf>OK<cr_lf>	<cr_lf>36,00<cr_lf>
104	ATSW26,default,0	<cr_lf>OK<cr_lf>	<cr_lf>36,00<cr_lf>
105	ATSW27,1200	<cr_lf>OK<cr_lf>	<cr_lf>46,00<cr_lf>
106	ATSW28,60,60	<cr_lf>OK<cr_lf>	<cr_lf>67,00<cr_lf>
107	ATSW29,default,0	<cr_lf>OK<cr_lf>	<cr_lf>72,00<cr_lf>
108	ATSW29,default,1	<cr_lf>OK<cr_lf>	<cr_lf>72,00<cr_lf>
109	ATSW29,default,2	<cr_lf>OK<cr_lf>	<cr_lf>72,00<cr_lf>
110	ATSW30,0	<cr_lf>OK<cr_lf>	<cr_lf>74,00<cr_lf>
111	ATSW30,1	<cr_lf>OK<cr_lf>	<cr_lf>74,00<cr_lf>
112	ATSP,…,default	<cr_lf>OK<cr_lf>	<cr_lf>39,00<cr_lf>
113	ATOP	<cr_lf>OK<cr_lf>	<cr_lf>78,00<cr_lf>
114	ATSC,00000000	<cr_lf>OK<cr_lf>	<cr_lf>40,00<cr_lf>
115	ATSSN,COM0	<cr_lf>OK<cr_lf>	<cr_lf>38,00<cr_lf>
116	ATSR21,2	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
117	ATSR21,3	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
118	ATSR21,4	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
119	ATSR21,5	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
120	ATSR21,6	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
121	ATSR21,7	<cr_lf>OK<cr_lf><cr_lf>0 or 1<cr_lf>	<cr_lf>44,00 <cr_lf>0 or 1<cr_lf>
122	ATDI,4,00000000	<cr_lf>OK<cr_lf>addresses,COD,name, DONE<cr_lf>	<cr_lf>12,00<cr_lf>addresses,COD,# of characters in name,name <cr_lf>12,03<cr_lf>
123	ATIL	<cr_lf>OK<cr_lf>	87,00<cr_lf> // Same ATDI<cr_lf>
124	ATSMA,< >,1101	<cr_lf>OK<cr_lf>	<cr_lf>42,00<cr_lf>
125	ATMACLR	<cr_lf>OK<cr_lf>	<cr_lf>43,00<cr_lf>
126	ATDM,	<cr_lf>OK<cr_lf><cr_lf>connect ,address<cr_lf>	<cr_lf>21,00<cr_lf><cr_lf>01,address
127	ATDL	<cr_lf>OK<cr_lf><cr_lf>connect ,address<cr_lf>	<cr_lf>77,00<cr_lf><cr_lf>01,address
128	ATLAST	<cr_lf>OK<cr_lf><cr_lf>address<cr_lf>	<cr_lf>64,00<cr_lf><cr_lf>address
129	ATDS	<cr_lf>OK<cr_lf>	<cr_lf>22,00<cr_lf>
130	ATDH	<cr_lf>OK<cr_lf>	<cr_lf>23,01<cr_lf><cr_lf>23,03
131	+++	<cr_lf>OK<cr_lf>	<cr_lf>32,00<cr_lf>
132	ATSESC	<cr_lf>OK<cr_lf>	<cr_lf>65,00<cr_lf>
133	ATMD	<cr_lf>OK<cr_lf>	<cr_lf>31,00<cr_lf>
134	ATMF	<cr_lf>OK<cr_lf>	<cr_lf>33,00<cr_lf>
135	ATUCL	<cr_lf>OK<cr_lf>	<cr_lf>51,00<cr_lf>
136	ATURST	<cr_lf>OK<cr_lf>	<cr_lf>50,00<cr_lf>
137	ATFRST	<cr_lf>OK<cr_lf>	<cr_lf>75,00<cr_lf><cr_lf>15,00<cr_lf>
138	ATSSW,0,X	<cr_lf>OK<cr_lf>	
139	ATRSW	<cr_lf>OK<cr_lf><cr_lf>0<cr_lf>	
140	ATDSCO	<cr_lf>SCO CONNECT<cr_lf>	<cr_lf>24,09<cr_lf>
141	ATDHSCO	<cr_lf>SCO DISCONNECT<cr_lf>	<cr_lf>24,11<cr_lf>
142	ATPARK,X,X	<cr_lf>NO CARRIER<cr_lf><cr_lf>OK<cr_lf>	<cr_lf>26,03<cr_lf><cr_lf>26,00<cr_lf>
143	ATXPARK	<cr_lf>NO CARRIER<cr_lf><cr_lf>OK<cr_lf>	<cr_lf>29,03<cr_lf><cr_lf>29,00<cr_lf>
144	ATSNIFF,X,X,X,X	<cr_lf>NO CARRIER<cr_lf><cr_lf>OK<cr_lf>	<cr_lf>27,03<cr_lf><cr_lf>27,00<cr_lf>

145	ATSSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>38,00<cr_lf>
146	ATCSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>95,00<cr_lf>
147	ATXSNIFF	<cr_lf>NO CARRIER<cr_lf><cr_lf>OK<cr_lf>	<cr_lf>28,03<cr_lf><cr_lf>28,00<cr_lf>
148	ATPAIR,address	<cr_lf>OK<cr_lf><cr_lf>address	<cr_lf>70,00<cr_lf><cr_lf>70,address
149	ATUPAIR,index	<cr_lf>OK<cr_lf>	<cr_lf>80,00<cr_lf>
150	ATUPAIRB,	<cr_lf>OK<cr_lf>	<cr_lf>95,01<cr_lf>
151	ATCPAIR	<cr_lf>OK<cr_lf>	<cr_lf>95,00<cr_lf>
152	ATCSNIFF	<cr_lf>OK<cr_lf>	<cr_lf>95,00<cr_lf>
153	ATSPF,15,+	<cr_lf>OK<cr_lf>	<cr_lf>79,00<cr_lf>
154	ATSTORE	<cr_lf>OK<cr_lf>	<cr_lf>90,00<cr_lf>
155	ATREAD	<cr_lf>OK<cr_lf>	<cr_lf>91,<cr_lf>
156	ATLSTO	<cr_lf>OK<cr_lf>	<cr_lf>88,00<cr_lf>
157	ATRSSI	<cr_lf>OK<cr_lf><cr_lf>+00<cr_lf>	
158	ATLQ	<cr_lf>OK<cr_lf><cr_lf>250<cr_lf>	
	BNEP/PAN		
159	ATDBNEP	<cr_lf>OK<cr_lf>	
160	ATHBNEP	<cr_lf>OK<cr_lf>	
161	ATDIP	<cr_lf>OK<cr_lf>	
162	ATHIP	<cr_lf>OK<cr_lf>	
163	ATAIP	<cr_lf>OK<cr_lf>	
164	ATLTCP	<cr_lf>OK<cr_lf>	
165	ATSIP	<cr_lf>OK<cr_lf>	
166	ATSPAN	<cr_lf>OK<cr_lf>	
167	ATSSW,2	<cr_lf>OK<cr_lf>	
168	ATRSW,2	<cr_lf>OK<cr_lf>	
		<cr_lf>OK<cr_lf>	,00
		<cr_lf>CONNECT<cr_lf>	,01
		<cr_lf>NO CARRIER<cr_lf>	,03
		<cr_lf>ERROR<cr_lf>	,04

APPENDIX B: AT Command Summary Table

Preliminary

BlueRadios AT Command	Description	Requires Reset (Y/N)	Stores Permanently (Y/N)
ATVER,ver1	Module Firmware Version	N/A	Yes
ATSI,0	Module Type	N/A	Yes
ATSI,1	Bluetooth Address ID	N/A	Yes
ATSI,2	Friendly Name	N/A	Yes
ATSI,3	Connection Status	N/A	Yes
ATSI,4	Service Name	N/A	Yes
ATSI,5	Class of Device (COD)	N/A	Yes
ATSI,6	ATSW24 Status	N/A	Yes
ATSI,7	ATSW25 Status	N/A	Yes
ATSI,8	UART Settings	N/A	Yes
ATSI,9	Auto Connect Masters Stored Slaves Address	N/A	Yes
ATSI,10	Slaves Scan Interval and Window	N/A	Yes
ATSI,11	PIO(5) Pulse Rate	N/A	Yes
ATSI,12	Escape Character Setting	N/A	Yes
ATSI,13	Inquiry and Masters Timeout	N/A	Yes
ATSI,14	Maximum Transmit Power Level	N/A	Yes
ATSI,15	Password Lock Code Status	N/A	Yes
ATSI,16	Deep Sleep Status	N/A	Yes
ATSI,17	Sniff Settings	N/A	Yes
ATSI,18	Link Supervisory Timeout setting	N/A	Yes
ATSI,19	Paired or secured connection index table	N/A	Yes
ATSI,20	Read IP auto connect address and port	N/A	Yes
ATSW,20	Switch 20: UART Settings	No	Optional
ATSW,21	Switch 21: Page Scan Interval & Window	Yes	Yes
ATSW,22	Switch 22: Set PIO State		
ATSW,23	Switch 23: Set PIO Logic	No	Optional
ATSW,24	Switch 24: Set Power Up Default Settings	No	Yes
ATSW,25	Switch 25: Set Auto Connecting Modes	No	Yes
ATSW,26	Switch 26: Lock User Definable Settings	No	Yes
ATSW,27	Switch 27: Set LED Pulse Rate on PIO(5)	No	Yes
ATSW,28	Switch 28: Set Inquiry Timeouts	No	Yes
ATSW,29	Switch 29: Set Password Lock Code	No	Yes
ATSW,30	Switch 30: Set Deep Sleep Mode	No	Yes
ATSC	Set COD (Class of Device)	No	Yes
ATSSN	Set Service Name	Yes	Yes
ATSP	Set PIN	Yes	Yes
ATOP	Override Password	No	Yes
ATSR,21	Set Register 21		??
ATDI	Dial Inquiry	N/A	N/A
ATIL	Inquiry Last	No	Yes
ATSMA	Set Master Address	Yes	Yes
ATMACLR	Master Address Clear	No	Yes

ATDM	Dial Master	N/A	N/A
ATDL	Dial Last	No	Yes
ATLAST	Last Connected Device Address	No	Yes
ATDS	Dial Slave	N/A	N/A
ATDH	Dial Hang-up	N/A	N/A
+++	Default Escape Characters	N/A	
ATSESC	Set Escape Characters	No	Yes
ATMD	Data Mode	N/A	
ATMF	Fast Data Mode	N/A	
Utilities			
ATUCL	Unit Cancel (idle mode)	No	No
ATURST	Unit Reset	N/A	N/A
ATFRST	Factory Reset	No	N/A
ATSSW	Bypass PIO(4) factory reconfiguration	??	Yes
ATRSW	Read above status of switch setting	No	Yes
ATSTORE	Store contents in memory register	No	Yes
ATREAD	Read contents from memory register	N/A	N/A
ATRSSI	Receive Signal Strength Indication	No	N/A
ATLQ	Link Quality	No	N/A
ATDSCO	Dial SCO	N/A	
ATDHSCO	Dial Hang-up SCO	N/A	
ATPARK	Park	No	No
ATXPARK	Exit Park	No	No
ATSNIFF	Sniff	No	No
ATSSNIFF	Set/Store Sniff	No	Yes
ATCSNIFF	Clear Sniff	No	Yes
ATXSNIFF	Exit Sniff	No	
ATPAIR	Pair	No	Yes?
ATUPAIR	Un Pair	No	N/A
ATUPAIRB	Un Pair by Bluetooth address		
ATCPAIR	Clear all paired or secured connections		
ATSPF	Set Power Factor	No	Yes
ATLSTO	Link Supervisory Timeout	No	Yes
BNEP/PAN			
ATDBNEP	Dial BNEP	N/A	N/A
ATHBNEP	Hang-up BNEP	N/A	N/A
ATDIP	Dial Internet Protocol address	N/A	N/A
ATHIP	Hang-up Internet Protocol	N/A	N/A
ATAIP	Abort Internet Protocol	N/A	N/A
ATLTCP	Listen TCP	N/A	??
ATSIP	Set Internet Protocol Address and Port Number	N/A	??
ATSPAN	Status Personal Area Network State	N/A	N/A
ATSSW,2	Set Switch #2 Auto Connect IP (Yes/No)		
ATRSW,2	Read Switch #2 Auto Connect IP (Yes/No)		