RooTooth User Guide

Revision 1.20

Copyright 2006 – RoboDynamics Corp.

Introduction

RooTooth provides Bluetooth capabilities to your Roomba and allows you to connect and communicate with it using any Bluetooth enabled device over Bluetooth's Serial Port Profile (SPP). RooTooth is ideal for wirelessly interfacing your Roomba to common Bluetooth enabled devices such as PCs, laptops, PDAs, and cell phones.

Communication with Roomba occurs through a virtual COM Port created on your device through Bluetooth's SPP, which allows for Serial communications wirelessly. Any program that can talk to the Serial port is able to send commands to the Roomba as well as receive information from it. The RooTooth is designed to accommodate the Class1 or Class2 Bluetooth radio modem serial modules at 2.4GHz frequency.

1. The AT Commands

This section describes the protocol used to control and configure the RooTooth module. The protocol is similar to the industry standard Hayes AT protocol used in telephone modems. Appropriate AT commands have been provided to make the module perform the two core actions of a Bluetooth device, which are make/break connections and inquiry. Additional AT commands are also provided to perform ancillary functions.

The CSR (Cambridge Silicon Radio) BC02 chipset in the BlueTooth modules is memory resource limited, therefore it is NOT proposed that there be full implementation of the AT protocol characteristic of an AT modem. However, the protocol is similar enough such that the existing source code written for modems can be used with very little modification for this serial module.

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 actions. The command set is extensive enough to allow a host to make connections, which are authenticated and encrypted or not. The RooTooth can be configured, commanded, and controlled using simple ASCII strings sent through the hardware serial UART or over a remote Bluetooth RF connection.

1.1 IMPORTANT NOTES ABOUT THE AT COMMANDS

1. All commands obey the following format – "[command]<cr>", where "cr" represents carriage return (0x0D in ASCII Table).
2. Valid commands respond with "<cr,lf>OK<cr,lf>" or "<cr,lf>ERROR<cr,lf>". Notice that "lf" represents a linefeed character (0x0A). The exceptions to this rule are "ATSW20" (set Baud Rate) and "ATURST" (perform CPU/Factory Reset) which both have no response.
3. All response data following a command response obey the following format "<cr,lf>[data]<cr,lf>".

Copyright 2006 – RoboDynamics Corp.

2. Interface

The TX and RX of the Bluetooth module on RooTooth are connected to the RX and TX pins of Roomba respectively, such that two modules form a null modem. The user controls the TX and RX pins of the Bluetooth module through the controlling application. Additionally, there is a general purpose I/O pin on the Bluetooth module which user can control through the AT commands. This I/O pin is connected to the DD pin of the Roomba. A falling edge on this pin causes Roomba activate. Figure 1 shows the block diagram of the system.

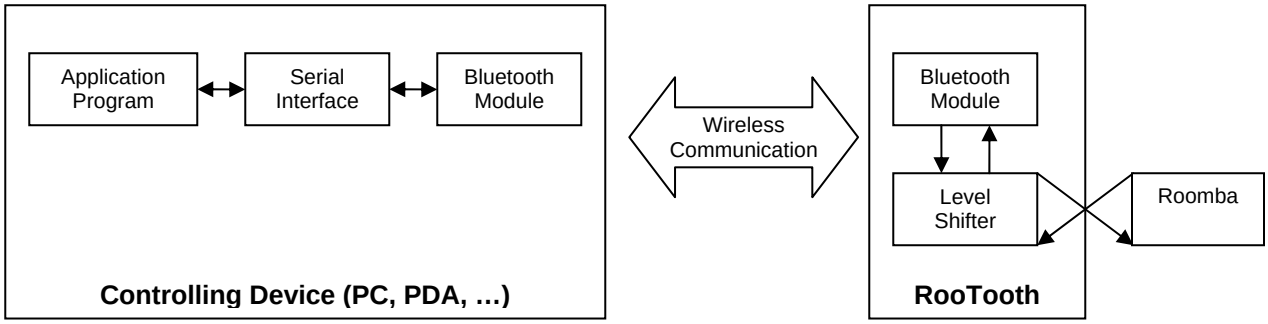

Figure 1. Block Diagram of the RooTooth System and Communication with the Controlling Application

The interface between the RooTooth and Roomba is through a 5 pin connector, which connects to the 5 pins of the DIN 7 connector on the Roomba. Table 1 shows the pin map of J1 connector on the RooTooth board.

Note: The pins with the same names on RooTooth and Roomba should be directly connected together. Do not switch the RXD and TXD signals. Pins 6 and 7 are unused by the RooTooth.

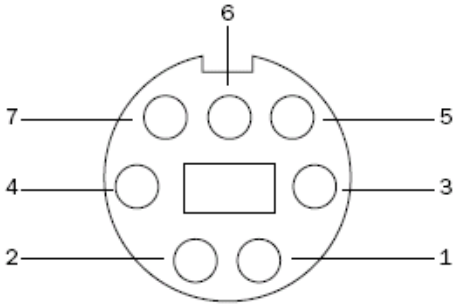

Figure 2. Pin-out connector of the RooTooth. Refer to the Roomba SCI Manual for more information.

Pin #	RooTooth Signal Name	IO Direction	Explanation
1	Vpwr (15.6V)	Power In (10 – 25 V)	
2	RXD	Input (0 – 5 V)	
3	TXD	Output (0 – 3.3 V)	
4	DD	Output (0 – 3.3 V)	Connected to PIO6 pin of the Bluetooth module
5	GND	Ground (0V)	

Table 1. Pin Map of J1 Connector on RooTooth

Note: While the RooTooth is plugged in the Roomba, it is always on. Leaving the RooTooth plugged for a prolonged period of time will slowly drain the battery of the Roomba if it is not docked at the charging station.

Copyright 2006 – RoboDynamics Corp.

3. Configuring the RooTooth

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 RooTooth must be in command mode and enabled to accept AT Commands. While in command mode, RooTooth will accept ASCII bytes as commands.

3.1 COMMUNICATION SETTINGS

In Windows when prompted by the Add Device Wizard, the passkey for pairing with the RooTooth is ***default***. The default communication settings are as follows:

9600 bps, 8 Data Bits, No Parity, 1 Stop Bit

Once you change these parameters, you have the option to store them permanently in non-volatile memory.

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 a <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 ms to cause a reset in case of electrical "brown-out" or poor input supplied VDD. Allow 1 second for module to fully reboot. Module will not initially boot-up reliably if the VDD ramp rate is in milliseconds.

3.2 RESETTING THE ROOTOOTH

To reset the RooTooth, just unplug it from the Roomba, wait one second and plug back in. ***Do not unplug during data transfer****.*

WARNING:

During prolonged use, the voltage regulator on the RooTooth board may become hot. Please use caution when unplugging the RooTooth.

Copyright 2006 – RoboDynamics Corp.

4. AT Command Description

As explained in the previous section, the BlueTooth module settings on RooTooth may be viewed and changed using AT commands.

If the radio is Bluetooth connected, the application program will need to send **+++<cr>** on the local UART (Virtual COM Port) to take the radio out of data mode and place it into command mode. The application program can then enter any of the AT commands described in the following sections, followed by <cr>. Commands will return an "OK", for valid response, and "ERROR" for invalid. To return to regular data mode while RF connected, the application program should send **ATMD** to pass or receive data from a remotely connected Bluetooth device.

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

In the next section, the details of each AT command are explained. The box below explains the AT Command Prefix which must precede all AT commands.

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

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 exceptions are ATVER, ATOP, and the radio's Personal Identification alphanumeric Number (PIN) which is caps sensitive.

ATVER,ver1 **Firmware Version**

Get Radios firmware version (***ver1*** is lower case).

EXAMPLE:
TYPE : **ATVER,ver1<cr>**
REPLY: **<cr_lf>OK<cr_lf>**
<cr_lf>Ver 2.8.1.0.0<cr_lf>

Make sure this version number matches this document version before proceeding.

Copyright 2006 – RoboDynamics Corp.

5. AT Command List

5.1 GET / SET RADIO INFORMATION

> 5.1.1 Get Status Information

Status Information (SI) 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.

ATSI,<n> **Status Information**

Get radio status information of type <n>. The following table lists the information types and their corresponding numbers <n>.

<n>	Information
0	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)"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) 003C,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 enabled. (HEX)01

See next page for examples.

Copyright 2006 – RoboDynamics Corp.

EXAMPLE(s):
TYPE : **ATSI,0<cr>**
REPLY: **<cr_lf>OK<cr_lf><cr_lf>RooTooth 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> RooTooth <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>0027,0003,0002<cr_lf> // Not set. Still defaults to 9600bps**

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

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**

Copyright 2006 – RoboDynamics Corp.

> 5.1.2 Set the Radio Name

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

Note: To detect the new name for the device after a name change, the computer may need to be restarted.

ATSN,<name> **Set Name**
Sets the radio's friendly name (16 alphanumeric characters maximum).

EXAMPLE:
TYPE : **ATSN,MYRADIOS_0123456<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

This example command renames the RooTooth to "MYRADIOS_0123456". By default, your RooTooth will have the name "RooTooth" until changed.

> 5.1.3 Write Memory Locations

ATSW,<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.

The table below shows the possible <n> values for the ATSW command. Each <n> command will be explained in further detail after the table. Please refer to their respective boxes for a more detailed description and example.

<n>	Description	Arguments
20	UART Settings	,<baudrate>,<parity>,<stopbits>,<store>
22	Set PIO State	,<PIO#>,<state>,<store>
23	Set PIO Logic	,<PIO#>,<value>,<store>
24	Power Up Default Settings	,<value>,<value>,<value>,<value>
26	Lock User Definable Settings	,<value>,<value>
29	Set PIN Lock Code	,<PIN>,<value>
30	Set Deep Sleep Mode	,<value>

Copyright 2006 – RoboDynamics Corp.

ATSW20,<baudrate>,<parity>,<stopbits>,<store> **UART Settings**

Configure the UART Settings. Flow control is always enabled and shorting CTS/RTS are shorted together.

Note: *Factory default setting is: 9600 8, N, 1 with hardware flow control RTS/CTS enabled. You cannot change the number of data bits from 8.*

Baud Rates: 1200 - 921600 bps (see table below).

Baud Rate	ASCII Value	Error
same	0	-
1200	5	1.73%
2400	10	1.73%
4800	20	1.73%
9600	39	-0.82%
192k	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%

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.

To reconfigure Radio back to default factory settings, apply 3.3Vdc on PIO#4 during initial power up for 1 second.

Copyright 2006 – RoboDynamics Corp.

ATSW22,<PIO#>,<state>,<store> **Set PIO Mode to Input or Output**

Set the direction of the PIO pin as either input or output. Below is a table of the pins and their possible <state> arguments.

Note: *Only PIO(6) is available on RooTooth, which is connected to the DD input of Roomba*

<PIO#>	PIO	<state>
2	= PIO(2)	This PIN IS NOT available
3	= PIO(3)	This PIN IS NOT available
4	= PIO(4)	This PIN IS NOT available
5	= PIO(5)	This PIN IS NOT available
6	**= PIO(6)**	**0 = Input (Default setting – DD of Roomba)** **1 = Output**
7	= PIO(7)	This PIN IS NOT available

Store Parameters: 0 = Do Not Store, 1 = Store Parameters in Flash

EXAMPLE*: To Wake Up the Roomba*

TYPE : **ATSW22,6,1,1<cr>** ***// Set PIO6 to output***
REPLY: **<cr_lf>OK<cr_lf>**
TYPE : **ATSW23,6,0,1<cr>** ***// Toggle PIO6 low***
REPLY: **<cr_lf>OK<cr_lf>**
TYPE : **ATSW23,6,1,1<cr>**
REPLY: **<cr_lf>OK<cr_lf>** ***// Toggle PIO6 high***

ATSW23,<PIO#>,<value>,<store> **Set PIO Logic**

Configure the logic of the PIO. Below is a table of the pins and their possible <value> arguments.

<PIO#>	PIO	<value>
2	= PIO(2)	This PIN IS NOT available
3	= PIO(3)	This PIN IS NOT available
4	= PIO(4)	This PIN IS NOT available
5	= PIO(5)	This PIN IS NOT available
6	**= PIO(6)**	**0 = Off / 0V** **1 = On / +V (user definable)**
7	= PIO(7)	This PIN IS NOT available

Store Parameters: 0 = Do Not Store, 1 = Store Parameters in Flash

EXAMPLE*: To Wake Up the Roomba*

TYPE : **ATSW22,6,1,1<cr>** ***// Set PIO6 to output***
REPLY: **<cr_lf>OK<cr_lf>**
TYPE : **ATSW23,6,0,1<cr>** ***// Toggle PIO6 low***
REPLY: **<cr_lf>OK<cr_lf>**
TYPE : **ATSW23,6,1,1<cr>**
REPLY: **<cr_lf>OK<cr_lf>** ***// Toggle PIO6 high***

Copyright 2006 – RoboDynamics Corp.

***ATSW24,<responseType>,<authMode>,<autoSCO>,<minorFilter>* Power Up Default Settings**

Change power up response type, security, audio, and filter modes.

Notes: *Factory default setting is* ***ATSW24,0,0,0,0***
Software or Hardware Reset required for change to take affect

responseType	authMode	autoSCO	minorFilter
0 : Long Response	**0** : No Authorization	**0** : No Automatic SCO Connect	**0** : No Minor Filter On COD
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.			

responseType = *AT Command Response*
Sets the format of the RooTooth reply to an AT command. Can be in LONG(0), SHORT(1), or NONE(2) form. The default is LONG FORM. See table at bottom for response types and their corresponding definitions.

authMode = *Authentication Security Mode.*
56bit encryption is automatically enabled when "**1**" is selected and default PIN = "**default**". UART will reply with **LINK,BTaddress** before the **CONNECT,BTaddress**.

minorfilter = *Filter Mode*
Default filter = 00000000 // no filter, finds all devices.

ATSW24,responseType, authMode, autoSCO, minorFilter // definitions
ATSW24,slave, fast data mode, allow data to pass // settings

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

Response Codes Table:
How the RooTooth will reply using the chosen responseType. Reference Appendix A

Long Form [0]	Short Form [1]	None [2]	Definition / Explanation
OK	00		Command correct and completed
CONNECT	01		Connection established
RING			Incoming ring detected
NO CARRIER	03		No connection or lost the carrier
ERROR	04		Bad command
NO DIALTONE			Dial tone not detected in S7 seconds
BUSY			Busy tone detected
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

Copyright 2006 – RoboDynamics Corp.

ATSW26,<PIN>,<lock/unlock> **Lock User Definable Settings**

Prevent unauthorized local and remote changing of settings.

<PIN> is the old PIN password. Default password is "default". Use it to access your RooTooth. To set a new PIN, see the next section.

<lock/unlock>	Setting
0	Unlocked
1	Locked

EXAMPLE: *Locks the RooTooth to prevent unauthorized access.*
TYPE : **ATSW26,default,1<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

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

ATSW29,<PIN>,<value> **Set PIN Lock Mode**

Maximum length of password is 16 alphanumeric characters, including spaces. Caps sensitive. This command also enables ATOP in the next section.

<PIN> is the PIN password. Use it to access your RooTooth.

<value>	Setting
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,1r>**
REPLY: **<cr_lf>OK<cr_lf>**

Copyright 2006 – RoboDynamics Corp.

ATSW30,<value> **Set Deep Sleep Mode**

Puts the RooTooth into deep sleep mode. See table below for settings.

Note: *This setting is stored in flash and does not require a reset to take affect. Takes approximately 1 second before the current will drop down to 90uA. Allow 5ms for the CPU unit to come out of deep sleep.*

<value>	Setting
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>**

Notes:

- *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.*

> 5.1.4 Security

ATSP,<new PIN>,<old PIN> **Set Personal Identification Number (PIN)**

Set PIN, Max alphanumeric characters (16) includes spaces (CAPS sensitive).

WARNING: *Be careful when entering a new PIN. There is no way to obtain PIN status after it is changed.*

EXAMPLE: *Change PIN from default to 1234567890123456*
TYPE : **ATSP,1234567890123456,default<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

Copyright 2006 – RoboDynamics Corp.

ATOP,<PIN> **Overwrite PIN**

Set Overwrite PIN so you do not need to enter old PIN to set a new PIN. Maximum of 16 alphanumeric characters, including spaces (CAPS sensitive).

Note: *This command is used in conjunction with ATSW29 command.*

EXAMPLE:
TYPE : **ATOP,1234<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

> 5.1.5 (COD) Class of Device

ATSC,<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 to.

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

EXAMPLE:
TYPE : **ATSC,00020114<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

> 5.1.6 Set Service Name

ATSSN,<value> **Set Service Name**

Set the *Bluetooth* Service Name. Maximum of 16 alphanumeric characters. Factory default is COM0.

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

EXAMPLE:
TYPE : **ATSSN,COM0<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

> 5.1.7 Read Memory Locations

ATSR<n> **Read an S Register**

The S registers refer to memory locations used for configuration

Copyright 2006 – RoboDynamics Corp.

ATSR21,<PIO#>

PIO#			
2 = PIO(3)	(0 = off,0v)	(1 = on,+V)	**(Reserved: Indicates BT connection)**
3 = PIO(3)	(0 = off,0v)	(1 = on,+V)	(user definable)
4 = PIO(4)	(0 = off,0v)	(1 = on,+V)	**(Reserved: Dual 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>**

5.2 INQUIRY / CONNECT / DISCONNECT COMMANDS

> 5.2.1 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.

ATUCL **Clear Unit**
Recommend executing this before performing an inquiry command. Places radio in idle mode. For more information, see section 5.4 Utilities.

ATDI,<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,Rootooth<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.*

Copyright 2006 – RoboDynamics Corp.

> 5.2.2 Set Master Default *Bluetooth* Address/Profile for Slave

ATSMA,<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.

Note: *Execute* ***ATSW25,1,0,0,0*** *"Master default . . ." before using the below command if still in slave mode.*

EXAMPLE (MASTER):
TYPE : **ATSMA,00A0961F904F,1101<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

Note: *Reset module for change to take effect.*

ATMACLR **Clears Stored Slave Address in Master**

EXAMPLE:
TYPE : **ATMACLR<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

> 5.2.3 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, thus when making a connection, if 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 less than 4 seconds.

ATDM,<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. Refer to UUID Table in following page.

EXAMPLE (MASTER):
TYPE : **ATDM,112233445566,UUID<cr>**
REPLY: **<cr_lf>OK<cr_lf>**
<cr_lf>CONNECT,00A0961D9C37<cr_lf> // successful connection to a slave
OR **<cr_lf>NO ANSWER<cr_lf>**
OR **<cr_lf>LINK,00A0961D9C37<cr_lf>** // if security is enabled

Copyright 2006 – RoboDynamics Corp.

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
PAN	**TB**
HID	**TB**

ATDL 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 **<cr_lf>?????<cr_lf>**

***ATLAST* Last Connected**

Displays the **LAST** connected Bluetooth device address.

EXAMPLE:
TYPE : **ATLAST<cr>**
REPLY: **<cr_lf>OK<cr_lf>**
<cr_lf>000000000000<cr_lf> // Nothing stored
OR **<cr_lf>123456789012<cr_lf>** // Last connected device

Note: *Unless the device was paired the address will be lost if power cycled.*

Copyright 2006 – RoboDynamics Corp.

> 5.2.4 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.

ATDS **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>**

> 5.2.5 Disconnect

ATDH **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 Rootooth Slave, the Slave will disconnect and go into Idle mode, not Slave mode by design.*

Copyright 2006 – RoboDynamics Corp.

5.3 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. Otherwise the **+++<cr>**will be accepted because the AT Command parser is still on and returns "**OK**"**.** In the previous 4Mbit legacy modules setup where the escape characters were fixed and set to "ATMC", is no longer implemented.

2) Operation Modes:

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. The different operation modes are listed below.

+++ 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.

EXAMPLE:
TYPE : **+++<cr>**
REPLY: **<cr_lf>OK<cr_lf>**

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.

***ATSESC ,<nn>* Set Escape ASCII Character**

Where 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>**

Note: *Do not need to perform reset and stores permanently.*

Need to add error constraint if not an integer value.

Copyright 2006 – RoboDynamics Corp.

ATMD **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 "**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>**

ATMF **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 are to reset power on the radio or strobe PIO(4) while connected.

If the command is successful, an "**OK"** is returned. Nothing will be returned if already in fast data mode and connected. "**No Carrier"** returned if the Bluetooth RF link is not established.

EXAMPLE:
TYPE : **ATMF<cr>**
REPLY: **<cr_lf>OK<cr_lf>** // if connected
OR
REPLY: **<cr_lf>OK<cr_lf>**
<cr_lf>NO CARRIER<cr_lf> // not connected

Note: Byte Gaps and Data Latency – Due to the way *Bluetooth* is designed and operates, random byte gaps of 5 ms to 20 ms 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. In regular data mode it is 60Kbps 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.

Copyright 2006 – RoboDynamics Corp.

5.4 UTILITIES

> 5.4.1 Cancel Command

ATUCL **Cancel Command**

The UCL command tells the radio to cancel inquiry or connect request commands and then places the radio in Idle Mode. This command is 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 cannot cancel a command while RF connected.

EXAMPLE:
TYPE : **ATUCL<cr>**
REPLY: **<cr_lf>OK<cr_lf>** *OR* **<cr_lf>ERROR<cr_lf>**

> 5.4.2 Reset

ATURST **Reset**

The URST command tells the radio to perform software reset on the CPU.

EXAMPLE:
TYPE : **ATURST<cr>**
REPLY:

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

ATFRST **Factory Reset**

The FRST command tells the radio to perform factory reset on the CPU.

EXAMPLE:
TYPE : **ATURST<cr>**
REPLY:

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

Note: You can send either above reset commands though the UART or over Bluetooth RF connection.

Copyright 2006 – RoboDynamics Corp.

> 5.4.3 Pairing

ATPAIR,<BT Address> **Pair Bluetooth Address**
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: *Pairing Timeout is 30 seconds. Need to be in Idle mode first. Security PINs are exchanged.*

ATUPAIR **Unpair Bluetooth Address**
The command tells the radio to un-pair from the Bluetooth stored address.

EXAMPLE:
TYPE : **ATUPAIR<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 or pair to a new device.*

ATAPAIR **Address of Last Paired**

EXAMPLE:
TYPE : **ATAPAIR<cr>**
REPLY: **<cr_lf>OK<cr_lf>**
<cr_lf>123456789012<cr_lf> *OR*
<cr_lf>000000000000<cr_lf> // Blank

Note: *The Slaves radio UART will output "LINK,BTaddress" for every secured connection. the RooTooth Masters UART will output the same if security flag is enabled.*

Copyright 2006 – RoboDynamics Corp.

> 5.4.4 Power Level

ATSPF,<value>,<sign> **Set Max TX Power Level**

This command sets the Radios maximum transmit power in (dBm) (Recommended values are 15, 12, 4, and 0 dBm). Factory default is 15dBm max for Class1 and 4 dBm for Class2 devices.

First value = integer from 20 to 0. Second value (sign) = (+) or (-)

Class 1 Power Table	Class 2 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
-6 dBm to +0 dBm Class3

Note: *The default value is 4dBm in a class2 BlueTooth 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>** *II* +4dBm
REPLY: **<cr_lf>OK<cr_lf>** *OR* if > (+15) or < (-20) **<cr_lf>ERROR<cr_lf>**

Copyright 2006 – RoboDynamics Corp.

Copyright 2006 – RoboDynamics Corp.