User Guide

For:

Serialio.com BlueSnap Smart Dongle
Serial Module Command Set

Revision 4
October 21, 2015

1. Introduction

Features

- Bluetooth Smart (Bluetooth Low Energy or BLE) modules are fully certified to meet Bluetooth version 4.1
- Integrated 8 Mbit SPI-serial flash supports firmware updates and multiple on-board applications
- On-board PCB-style antenna
- ARM® Cortex™ M3-based microprocessor core
- Programmable RF transmit power control
- Operation directly from a battery or mains supply
- Small keyfob size fits many use cases
- Secure Over-the-Air (OTA) software update using industry=standard AES-128 security

Interfaces

- A/D converter: 7 channels with 4 modes providing variable resolution (10-13 ENOBs), sampling rate (5.9-187 kHz), and conversion latency (5-171 µs)
- USART: 1 x 4-wire up to 1.5 Mbit/s for general use, 1 x 2-wire for in-circuit programming
- SPI: 1 x master, 1 x slave at ≤ 12 Mbit/s
- I^2C: 1 x I^2C master interface up to 1Mhz
- GPIO: Up to 12 GPIOs (overlaid with peripherals), programmable pull-up/pull-down resistors with 16 mA drive strength at 3.3V (2 mA standard)
- 4 x PWM channels each with a 10-bit counter clocked at 128 kHz
- Wake-up: Wake from GPIO for ultra-low power operation

Operational & Radio

- Operational voltage: 1.8V – 3.6V
- Operational Temperate Range: -30°C to +85°C
- Dimensions
- Weight
- Current consumption @ 3.0V, 25°C
    - Deep Sleep: 1.65 µA (typ.)
    - Sleep: 12 µA
    - Active Receive: 26.6 mA
    - Active Transmit: 22.0 mA
- RF Transmit Power: +4 dBm (max.)
- Receive Sensitivity: -94 dBm

2. General Information

The BlueSnap Smart RS232 adapter provides the lowest cost solution suitable for price sensitive applications. It is based on an ultra-low power Bluetooth Low Energy SoC from Broadcom Corporation. All modules incorporate serial-flash memory and a printed antenna to provide a state-of-the-art fully-certified Bluetooth 4.1 solution.

The Bluetooth Smart SoC from Broadcom is purpose-designed to support the entire spectrum of Bluetooth Smart use cases for medical, home automation, accessory, sensor, retail, and wearable market sectors. Superior receive sensitivity, an integrated RF transmit power amplifier and transmit/receive switch, along with the on-board antenna, provide extended range and full compatibility with all Bluetooth 4.1 devices.

The Bluetooth Smart dongle runs on a high-performance ARM® 32-bit Cortex™-M3 core operation at a frequency of 24 MHz. The SoC included is configured to boot applications stored in the 8 Mbit serial flash memory available on-board.

The unit provides an extensive array of I/O and peripheral interfaces. The following list of interfaces are available, many of which are accessible with I/O multi-plexing and alternate function capabilities:

- 7 x A/D converter channels
- 2 x UART interfaces: 1 x 4-wire, 1 x 2-wire
- 1 x SPI master, 1 x SPI slave bus
- 1 x I^2C interface
- Up to 12 x ultra-low power wake input
- 32kHz crystal interface

The BlueSnap Smart dongle can be powered directly from a battery supply in the range 1.8-3.6V including 2xAAA, 2xAA, or a single Lithium coin cell. The BlueSnap Smart dongle is also available in an eternally-powered version.

Internal power domains are automatically adjusted to minimize power dissipation based on user activity. Various power modes, including an ultra-low power deep sleep mode, are provided to minimize total average power consumption and maximize battery life.

Custom factory settings may be saved to a non-erasable area of static memory to enable out-of-the-box custom product configurations and the ability to return products to a known factory reset state.

The BlueSnap Smart dongle may be woken from deep sleep mode by a level transition on the GPIO assigned for stream mode selection. Other GPIOs may be dynamically allocated for control and status including a BLE connection indicator, serial bus mode control and status indicator, audible alerts, and factory resets.

An internal 32kHz low-power oscillator is available by default for non-critical timing requirements. Applications requiring an accurate real-time clock may connect an optional external 32kHz crystal.

All versions have Bluetooth BQB SIG certification and FCC & IC modular approval use in the United States and Canada, as well as CE approval for use in Europe and other countries.

3. Electrical Specifications

3.1 Absolute Maximum Ratings

The absolute maximum ratings in Table 1 and Table 2 indicate levels where permanent damage to the device can occur, even if these limits are exceeded for only a brief duration. Functional operation is not guaranteed under these conditions. Operation at absolute maximum conditions for extended periods can adversely affect long-term reliability of the device.

Table 1. Absolute Maximum Voltage Ratings

Symbol	Ratings	Min.	Max.	Unit
VDD_{max}	DC supply voltage supplied to VDD pin	GND – 0.3	3.8	V
V_{IO}	Voltage on input or output pin	GND – 0.3	VDD_{max} + 0.3	V

Table 2. Absolute Maximum Environmental Ratings

Characteristic	Note	Min.	Max.	Unit
Storage Temperature	-	-40	+125	°C
Relative Humidity	Non-condensing (storage)	-	65	%

3.2 Recommended Operating Conditions

Functional operation is not guaranteed outside the limits shown in Table 3 and Table 4, and operation outside these limits for extended periods can adversely affect long-term reliability of the device.

Table 3. Recommended DC Operating Conditions

Symbol	Ratings[1]	Min.	Typical	Max.	Unit
VDD	DC supply voltage applied to VDD pin				
	On-board serial flash read only	1.8	-	3.63	V
	On-board serial flash read and write	2.1	-	3.63	V
	On-board serial flash at power-on	2.25[2]	-	3.63	V

Notes:

[1]Overall performance degrades beyond minimum and maximum power voltages.

[2]The module may fail to boot if this condition is not met when power is initially applied.

Table 4. Recommended Environmental Conditions

Characteristic	Note	Min.	Max.	Unit
Ambient Temperature		-30	+85	°C
Relative Humidity	Non-condensing (operating)	-	85	%

3.3 Power Consumption

The unit automatically adjusts power dissipation based on user activity to minimize power usage. The power consumption in each state, and for each version of the module, is specified in Table 5.

Table 5. Power Consumption (3.0V, 25°C)

Operation State	Note		Typical	Unit
Deep Sleep[1]	BLE Chip (0.65µA) + Serial Flash (1 µA)		1.65	µA
Sleep	Wake in <5ms		12.0	µA
	Active Receive	Receiver enabled and operating at 100% duty cycle	26.6	mA
	Active Transmit[2]	Transmitter enabled and operating at 100% duty cycle	22.0	mA

Notes:

[1]Requires 10K pullup resistor on module pin 6. See 6.5.1, UART Application for Configuration and Streaming.

[2]RF transmit power = 0dBm

3.4 Input/Output Pins

Table 6. Digital I/O Pin Levels

Symbol	Description	Min.	Typical	Max.	Unit
V_{IL}	Input low voltage	-	-	0.4	V
V_{IH}	Input high voltage	0.75 x VDD	-	-	V
V_{OL}	Output low voltage[1]	-	-	0.4	V
V_{OH}	Output high voltage[1]	VDD – 0.4	-	-	V

Notes:

[1]At the specified drive current for the pin.

3.5 ADC Specifications

Table 7. ADC Specifications (@ 25°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Channels	-		-	8	-	-
Channel Switching Rate	f_{ch}		-	133.33	-	kch/s
Input signal range	V_{inp}		0	-	3.63	V
Reference settling time	-		7.5	-	-	µS
Input resistance	R_{in}	Effective, signal ended	-	500	-	kΩ
Input capacitance	C_{in}		-	-	5	pF
Conversion rate	R_c		5.859	-	187	kHz
Conversion time	Tc		5.35	-	170.7	µS
Effective Number of Bits	ENOB		10	-	13	bits
Abs. Voltage Meas. Error	-		-	±2	-	%
Current	I		-	-	1	mA
Leakage Current	$I_{leakage}$		-	-	100	nA
Power-up time	T_{pup}		-	-	200	µS
Integral nonlinearity	I_{NL}	LSBs at the 10-bit level	-1	-	1	LSB
Differential nonlinearity	D_{NL}	LSBs at the t0-bit level	-1	-	1	LSB

4. Bluetooth RF Specifications

Unless otherwise stated, the specifications in this section apply when operating conditions are within the limits specified in Section 3.2, Recommended Operating Conditions. Functional operation outside these limits is not guaranteed.

All specifications are measured with a coax pigtail soldered to the PCB antenna feed point, with VDD = 3.0V and at a room temperature of 25°C.

4.1 Summary RF Specifications

Table 8. Summary RF Specifications

Feature Supported	Description
Bluetooth Standard	Version 4.1
Frequency Band	2.400 GHz – 2.500 GHz
Channels (2MHz spacing)	3 x Advertising channels @ 2402 / 2426 / 2480 MHz 36 x Data channels
Maximum Raw Data Rate	1 Mbit/s
Maximum Application Data Rate	0.27 Mbit/s
Modulation Type	DSSS: GFSK (modulation index = 0.5)
Maximum RF Input	-10 dBm
Typical Receive Sensitivity	-94 dBm
Maximum RF Tx Output Power	+4 dMb
Carrier Frequency Accuracy	±20 ppm (24 MHz crystal, ±20 ppm stability @ 25°C)

4.2 Receiver Specifications

Table 9. Receiver Performance Specifications

Parameter	Condition/Notes	Min.	Typical	Max.	Unit
Frequency Range	-	2402	-	2480	MHz
Operating Temperature	-	-30	-	+85	°C
Receive Sensitivity	200 x 37-byte packets with 30.8% PER	-	-94	-	dBm
Maximum Receive Level	-	-10	-	-	dBm

4.3 Transmitter Specifications

Table 10. Transmitter Performance Specifications

Parameter	Condition/Notes	Min.	Typical	Max.	Unit
Frequency Range	-	2402	-	2480	MHz
Operating Temperature	-	-30	-	+85	°C
Transmit Power	-	-	4.0	-	dBm
Transmit Power Variation	Process dependent	-	2.0	-	-
Transmit Power Range	Adjustable via software control	-20	-	4	dBm
Channel Spacing	-	-	2	-	MHz

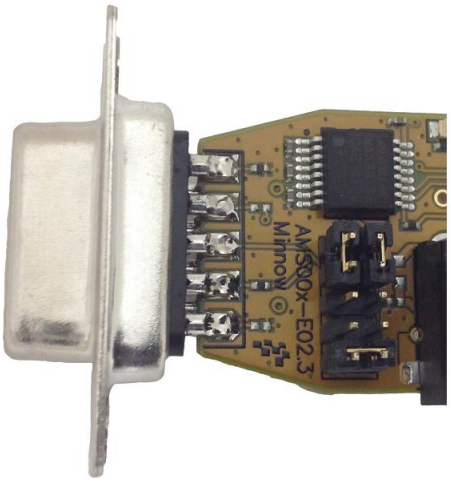
4.4 Serial Pin Configurations and Jumper Configuration

Table 11. DB9 Pinout

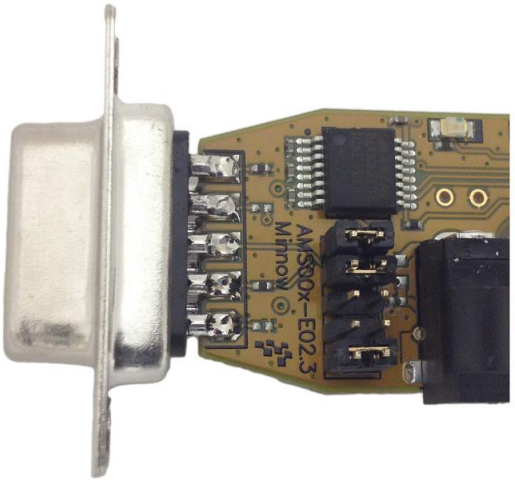
DB9 Female	IO DIR
1 – NC	
2 – DB9 RX	IN <-
3 – DB9 TX	OUT ->
4 – NC	
5 - GND	<->
6 – NC	
7 – DB9 RTS	OUT ->
8 – DB9 CTS	IN <-
9 – V+	IN <-

Table 12. Jumper Settings

DCE with No Flow Control (default)



DTE with No Flow Control



5. Description of Commands

adc - read an ADC value

Description – Get value of ADC in mV. Valid only for GPIOs that support ADC. The `adc` command can be used regardless of the GPIO function configuration.

Syntax – `adc <GPIO>`

Example

```
> adc 2
R000006
1404
```

```
> adc 3
R000006
2563
```

adv – Advertise as a peripheral

Description – Turn on advertising as a peripheral at the specified rate. The command `adv off` turns advertising off. If no argument is supplied, the default is `adv high`.

On reset, advertising defaults to high for a duration specified by *bl v h d* (default: 30 seconds), then switches to low for a duration specified by *bl v l d* (default: 300 seconds), then turns off.

The advertising settings correspond to the following advertising modes:

- `high` – High Duty Cycle Undirected Advertising
- `low` – Low Duty Cycle Undirected Advertising
- `off` – No Advertising

For more information, see the variables used to control advertising:

- *bl v h d*
- *bl v h i*
- *bl v l d*
- *bl v l i*

Syntax – `adv [<low/high/off>]`

Example

```
> adv high
Success
```

beep – Send a beep to a speaker

Description – Emit a short beep from a speaker. The speaker must be connected to a GPIO that is configured with the GPIO alternate function: speaker.

Syntax – `beep <duration>`
where: `<duration>` is expressed in milliseconds, ranging from 50 to 1000.

Example

```
>beep 200
Success
```

con – Connect to a peripheral

Description – Connect to a peripheral with the specified index number. The index number is obtained from the output of the *scan* command.

Syntax – `con <index>`

Example

```
> con 1
```

```
Success
```

dct – Disconnect from a peripheral

Description – Disconnect from a peripheral.

Syntax – `dct`

Example

```
> dct
Success
```

fac – Factory reset

Description – Factory reset. Return variables to factory default settings by deleting user configuration (if present). See *save*.

To avoid accidental factory reset, the BD address of the module must be provided as an argument. Obtain the BD address with the *get bl a* command.

Note: The default bus mode may change after a factory reset. If you are unable to communicate with the module with serial commands, it may be necessary to toggle from `STREAM` mode to `COMMAND` mode.

Syntax – `fac <BD_ADDRESS>`

Example

```
> get bl a
4C55CC129A42
> fac 4C55CC129A42
TruConnect-1.0.0.14, Built: Nov 10 2014 17:07:33, Module:AMS002.5,
Board:AMS001-E01.2
[COMMAND_MODE]
```

gdi – GPIO direction

Description – Set the direction and initial state of a general purpose I/O pin (configured as `stdio`). See *Peripherals*.

Syntax – `gdi <GPIO#> <direction>`
where:

- `<GPIO#>` - The GPIO number
- `<direction>` - may be one of the following types
  - `in` : Input high impedance
  - `ipd` : Input, pull-down
  - `ipu` : Input, pull-up
  - `olo` : Output initialized to low value
  - `ohi` : Output initialized to high value
  - `hiz` : High impedance

Example

```
> gdi 12 in
R000009
Success
```

get – Get the value of a variable

Description – Get the value of the specified variable.

Syntax – `get <variable>`

Example

```
> get ua b
115200
```

gfu – GPIO function

Description – Configure a GPIO with the specified function. A function may only be assigned to a pin that has a function set to `none` i.e the pin is not already assigned.

Factory reset is an exception: it may be moved to any pin, but not unassigned by setting to `none`.
A list of available functions is shown in the following table.

Note: A *save* and *reboot* is required after configuring a function, with the exception of the `none`, `stdio`, and `pwm` functions.

activity	BLE and UART activity indicator. Can act as a system power off, if the activity GPIO pin is connected to the enable of a power regulator. See *Power Management*.
ble_blink	GPIO toggles regularly when there is activity on the wireless BLE interface. Connect this GPIO to an LED to indicate wireless activity.
conn_gpio[1]	Connection indication GPIO (output) - Logic 0 : Not connected to any other BLE device - Logic 1 : At least one connection is established
factory[1]	Factory reset pin. May be moved but NOT unassigned.
mode_sel[1]	Selects bus serial mode (input) Depending on settings of variable: *bu s c* (bus serial control) as edge or level, mode_sel works as follows: edge: - mode toggles on rising edge level: - low level – COMMAND_MODE - high level – STREAM_MODE
none	GPIO is not assigned to any function (high impedance)
pwm	GPIO configured for use with the PWM command. Available only for a GPIO with a PWM connected.
reserved	GPIO not available to user. Do not connect to this pin.
shutdown	GPIO has to possible functions. - If activity GPIO is not configured, acts as edge triggered sleep/wake toggle - If activity GPIO is configured as described, and shutdown is asserted for more than 1 second, shuts down module. See *Power Management*.
sleepwake	GPIO level controls sleep or wake – sleep low, wake high. See *Power Management*.
speaker[1]	GPIO configured for use with beep command. This works only with a GPIO connected to a speaker.
status_led[1]	GPIO is configured to operate as general status indicator (output) to show the connection status. The blink pattern is controlled with the *sy l s* variable.
stdio	GPIO is configured as a standard IO (input/output/others). Use the following commands to control pins configured as stdio: - *gdi* : configure direction & configure initialization value - *gse* : set the output level - *gge* : set the input level
stream_gpio[1]	GPIO to indicate the serial bus is set to STREAM mode (output) - Logic 1 : STREAM mode - Logic 0 : COMMAND mode
user_cts[2]	The user UART CTS function (input) is assigned automatically to GPIO 4 when flow control is enabled by the *set ua f 1* command. Any other function configured for this pin is overridden.
user_rts[2]	The user UART RTS function (output) is assigned automatically to GPIO 3 when flow control is enabled by the *set ua f 1* command. Any other function configured for this pin is overridden.
user_rx	GPIO will be used as User UART RX (input).
user_tx	GPIO will be used as User UART TX (output).

NOTES:

- [1] This function may be assigned to only one pin at a time

- [2] This function is auto-assigned and cannot be manually controlled

See also: *gdi, gge, gse, bu s c, gp u, Peripherals – GPIOs*

Syntax – `gfu <GPIO#> <function>`

Example 1

```
> gfu 6 none
Success
> gfu 6 mode_sel
Success
```

Example 2

```
> gfu 13 speaker
Success
```

gge – Get GPIO value

Description – Get the current value of a general purpose I/O pin configured for the `stdio` function. See *Peripherals*.

Syntax – `gge <GPIO#>`

Example

```
> gge 12
1
```

gse – Set GPIO value

Description – Immediately set the value of a general purpose I/O pin. When setting a GPIO, the GPIO direction must be set correctly, using the GPIO direction command `gdi`, or the command will fail. See *Peripherals*.

Syntax – `gse <GPIO#> <value>`

Example

```
> gse 12 0
Success
> gge 12
0
```

pwm – Control a Pulse Width Modulator

Description – The `pwm` command controls a pulse width modulator GPIO. See *Peripherals*. The TPIO must be set to the `pwm` function. See *gfu*.

Syntax – `pwm <gpio> <[low_count high_count]|[stop]>`

The internal PWM clock rate is 128*1024=131072 Hz. The PWM signal is high for `high_count` clock cycles and low for `low_count` clock cycles. The maximum value of high_count + low_count is 1023, so the minimum frequency is about 128Hz and the maximum frequency is 65536Hz.

Note: It is not possible to achieve a duty cycle of 100%. To set a GPIO high or low, use *gfu* to set the GPIO to the `stdio` function, use *gdi* to set the direction to `ohi` or `olo`, then use *gse* to set the value to 1 or 0.

Duty cycle is expressed as a fraction of 1.0 in the equations below.

$$\text{high\_count} = \frac{131072 \times \text{duty\_cycle}}{\text{frequency}}$$

$$\text{low\_count} = \frac{131072 \times (1.0 - \text{duty\_cycle})}{\text{frequency}}$$

For a 0.5 (50%) duty_cycle:

$$\text{high\_count} = \text{low\_count} = \frac{131072 \times 0.5}{\text{frequency}}$$

Example

```
> gfu 13 none          ← Unassign any existing function on GPIO 13 (may not be needed)
Success
> gfu 13 pwm           ← Configure GPIO 13 for use with the PWM
Success
> pwm 13 250 250       ← Start PWM on GPIO 13 with 50% duty cycle since high = low = 250
Success
>pwm 13 stop           ← Stop PWM
Success
```

rbmode – Change remote device bus mode

Description – Change the bus mode of a remote device operating as a peripheral. The rbmode command enables a device operating as a central (and connected to a remote peripheral) to:

- Read the bus mode of the remote peripheral
- Set the bus mode of the remote peripheral by changing the BLE mode characteristic of the remote

If the bus mode of the remote peripheral is set to remote COMMAND mode, the remote peripheral device can be controlled as if it was a local device. To control the remote peripheral, the controlling module connects as a central to the (remote) peripheral and then:

- Issues `rbmode remote` to set the bus mode of the remote peripheral to COMMAND mode
- Switches itself to `stream` mode, either with the `mode_sel` GPIO (see *gfu*), or by issuing the *str* command
- Issues one or more commands to the remote peripheral as a stream, then reads the response(s)

For a detailed description of bus modes, see *Serial Bus Modes*, *Serial Interface*.

For a demonstration of a remote control of a device, see the *Bus Mode Selection and Remote Control* application note.

Notes:

- The bus mode of the remote peripheral device cannot be changed to `local COMMAND` mode using `rbmode`.
- The remote device must have remote access enabled. See *sy r e*.

Syntax – `rbmode [stream | remote]`

Example 1

```
> rbmode
stream
```

Example 2

```
> rbmode remote
Success
```

reboot – Reboot

Description – Reboot the application. After reboot, the bus serial mode is displayed between square brackets.

Syntax – reboot

Example

```
> reboot
[COMMAND MODE]
> set bu i stream
Success
> save
Success
> reboot
[STREAM MODE]
```

save – Save variables

Description – Save the current user configuration value of all variables to non-volatile flash memory. After save completes, user configuration variable settings are automatically loaded on reboot.

The `save factory <BD_ADDR>` command sequence writes the current value of all variables to a unique factory configuration.

Factory settings may be saved repeatedly in the lifetime of the module.

Before factory settings are saved, double check the module configuration is correct. To avoid accidentally running the factory save sequence, `BD_ADDR` must be passed as an argument.

If factory settings are saved with the `lock` option, factory settings cannot be overwritten. After running the `save` command once with the `lock` option, issuing the save factory command again results in the response `Command failed`.

The `BD_ADDR` may be obtained using the *get bl a* command. When factory reset is activated, user settings are discarded and factory settings are restored. See *fac*.

Syntax – `save [factory <BD_ADDR>] [lock]`

Example 1 – save user config

```
> save
```

```
Success
```

Example 2 – save factory settings

```
> get bl a
4C55CC012345
> save factory 4C55CC012345
Success
```

Example 3 – save factory settings and lock. Locked settings cannot be overwritten.

```
> save factory 4C55CC012345 lock
Success
```

scan – Scan for nearby peripherals

Description – Scan for nearby BLE peripherals. Scan mode may be `low` or `high`, which determines the scan rate. If no scan mode argument is supplied, the default is `high`.

Scanning continues for a fixed period which is 300 seconds for `low` and 30 seconds for `high`. For peripherals in range, the scan details are listed with an index number and an address. The index number is used with the *con* command to connect to the peripheral.

By default, scan is restricted to peripherals with the service UUID specified in the variable *bl s u*. Use the scan argument `all` to remove this restriction.

Issue `scan off` to turn off scanning immediately.

The scan command asynchronously sends scan results to the serial interface. If the system print level *sy p* >=3, asynchronous messages are shown and response indicating a device is detected may be interleaved with subsequent commands and responses.

To prevent asynchronous scan results appearing, set *sy p* < 3 and issue `scan results` to view results.

Peripherals with the service UUID specified by *bl s u* are returned in scan results by default. To scan for peripherals advertising with a specific `service UUID`, provide the service UUID as an argument to the scan command. Or, to scan for all BLE peripherals, use the `all` parameter.

Each peripheral detected during scanning is listed only once in scan results. To enable duplicate result listing, use the dup parameter. This parameter is useful for obtaining real=time information about the signal strength (proximity) of a peripheral.

Syntax – `scan [<low / high]> <all / service_uuid> <dup>] | [<off / results>]`

Examples

```
scan
scan low
scan high
scan off
scan dup
scan all
```

```
scan dup high
scan high 175f8f23-a570-49bd-9627-815a6a27de2a
scan low all dup
scan results
```

Example

```
> scan high
R000038
! # RSSI BD_ADDR Device Name
# 1 -46 4C:55:CC:1a:30:1f AMS-3DDF
# 2 -46 4C:55:CC:1a:30:1f AMS-301F
```

set

Description – Set the value of a variable. See the *variable* documentation for details of valid arguments.

Syntax – `set <variable> <args>`

Example

```
> set sy c e 0
Success
```

sleep – Sleep

Description – Put the module into the lowest power sleep state. The module sleeps until a wakeup event occurs such as an interrupt on the `mode_sel` GPIO.

Syntax – `sleep`

Example

```
> sleep
R000009
Success
```

str – Stream mode

Description – Switch to serial bus STREAM mode. See Serial Interface.

Syntax – `str`

Example

```
> str
STREAM_MODE
```

ver – Version

Description – Returns the app firmware version. This is the command equivalent of the sy v variable.

Syntax – `ver`

Example

```
> ver
TruConnect-1.0.0.1, Built: Nov 9 2014 12:58:29, Module: AMS001.4,
Board: AMS001-E01.2
```

Variable Reference

This section provides a list of variables with a full description of the function of each variable together with example usage.

Variables are cached in volatile RAM and must be saved to non-volatile flash memory to persist between reboots. To save variables to flash, use the *save* command. Some variables impact the operation of the entire system. A *save* and *reboot* is required before new settings for these types of variables take effect.

Documentation Format

Many of the responses shown in the examples in this section were captured with system print level (*sy p*) = all, and system command header enabled (*sy c h*) = 1. These settings are provided to make it easy for a host microcontroller to parse response headers. The response headers appear first in the response output and are similar to `R000009`.

Documentation for each variable will be in the format shown below.

variable

Brief Description	A one-line variable description
Access	get/set

Description – A description of variable function.

Arguments:

- A full list of mandatory and optional arguments.

Default

- The factory reset default value.

Get example

An example of how to read the variable, including response codes.

Set example

An example of how to write the variable, including response codes (for writeable variables)

Note: Do not forget to check out *command navigation tips* to make it easier to find and type specific variable names.

List of Variables

- **All Variables**
  - *al* – all variables
- **BLE Peripheral**
  - *bl a* – BD address
  - *bl c c* – connection count
  - *bl s u* – service UUID
  - *bl t a* – transmit power (advertising)
  - *bl t c* – transmit power (connection)
  - *bl v m* – advertising mode
  - *bl v h d* – advertising high duration
  - *bl v l d* – advertising low duration
  - *bl v h i* – advertising high interval
  - *bl v l i* – advertising low interval
- **Bus**
  - *bu I* – bus initialization mode
  - *bu s c* – bus serial control level or edge sensitivity
- **BLE Central**
  - *ce c c* – connection count
  - *ce c m* – connection mode
  - *ce s h d* – central high scan duration
  - *ce s h i* – central high scan interval
  - *ce s l d* – central low scan duration
  - *ce s l i* – central low scan interval
  - *ce s m* – scan mode
- **GPIO**
  - *gp u* – GPIO usage
- **System**
  - *sy a t* – system activity timeout
  - *sy b n* – board name
  - *sy c e* – command echo
  - *sy c h* – command header
  - *sy c m* – command mode
  - *sy c p* – command prompt enable
  - *sy d n* – device name
  - *sy l s* – system indicator status LED blink behavior
  - *sy o e* – OTA enabled
  - *sy p* – print level
  - *sy r e* – remote command enabled
  - *sy s t* – system go-to-sleep timeout
  - *sy u* – system UUID
  - *sy v* – firmware version
  - *sy w t* – system wakeup timeout
- **UART**
  - *ua b* – UART baud
  - *ua f* – UART flow
- **User Storage**

- *us v* – user storage

Variable Description

All

al

Brief Description	All variables
Access	get

Description – Returns a list of all variables. This includes the list of GPIO settings returned by gp u.

Arguments
- N/A

Default
- N/A

Get example

```
> get al
bl a        : 20737A129A42
bl c c      : 0
bl s u      : 175f8f23-a570-49bd-9627-815a6a27de2a
bl t a      : 0
bl t c      : 4
bl v m      : off
bl v h d    : 30
bl v h i    : 32
bl v l d    : 300
bl v l i    : 1024
bu i        : stream
bu s c      : edge
ce c c      : 0
ce c m      : none
ce s h d    : 30
ce s h i    : 96
ce s l d    : 300
ce s l i    : 2048
ce s m      : off
gp u  :
! # Description
# 0 i2c_sda
# 1 i2c_scl
# 2 user_tx
# 3 none
# 4 none
# 5 user_rx
# 6 none, factory
# 7 reserved
# 8 none
# 9 mode_sel
# 10 status_led
```

```
# 11 none
# 12 none
# 13 speaker
# 14 none
sy a t      : 300
sy b n      : AMS001-E01.2
sy c e      : 1
sy c h      : 0
sy c p      : 1
sy d n      : AMS-94A2
sy o e      : 1
sy p        : 4
sy r e      : 1
sy u        : 0D06EA434C55CC01B10B1411081309007E615F0A
sy v        : TruConnect-1.5.0.18, Built: Mar 19 2015 14:03:19, Module:
N/A, Board: N/A
ua b        : 115200
ua f        : 0
us v        :
```

BLE Peripheral

bl a

Brief Description	BD address
Access	get

Description – Get BLE device address

Arguments
- N/A

Default
- N/A

Get example

```
> get bl a
R000014
4C55CC129A42
```

bl c c

Brief Description	Connection count
Access	get/set

Description – Returns the number of centrals connected when acting as a peripheral.

Arguments
- N/A

Default

- N/A

Get example

```
> get bl c c
R000003
0
```

bl s u

Brief Description	Service UUID
Access	get/set

Description – Configures the peripheral service UUID. Also used as the default service UUID for the *scan* command.

Arguments
- <UUID>

Default
- N/A

Get example

```
> get bl s u
R000038
175f8f23-a570-49bd-9627-815a6a27de2a
```

Set example

```
> set bl s u 175f8f23-a570-49bd-9627-815a6a27de2a
R000009
Success
```

bl t a

Brief Description	Transmit power (advertising)
Access	get/set

Description – Configures the RF transmit power when advertising. Units: dBm, range -25 to 4.
Used in conjunction with *bl t c*. Connecting at higher power than advertising helps to provide a more stable connection.

Arguments
- <advertising_power_dmb>

Default
- 0

Get example

```
> get bl t a
R000003
```

```
4
```

Set example

```
> set bl t a 0
R000009
Success
```

bl t c

Brief Description	Transmit power (connection)
Access	get/set

Description – Configures the TF transmit power when connected. Units: dBm, range -25 to 4.
Used in conjunction with *bl t a*. Connecting at higher power than advertising helps to provide a more stable connection.

Arguments
- `<connection_power-dBm>`

Default
- `4`

Get example

```
> get bl t c
R000003
0
```

Set example

```
> set bl t c 4
R000009
Success
```

bl v m

Brief Description	Advertising mode
Access	get

Description – Get peripheral advertising mode, as set by the *adv* command.

Arguments
- N/A

Default
- N/A

Get example

```
> get bl v m
R000006
high
```

bl v h d

Brief Description	Advertising high duration
Access	get/set

Description – Configures high mode advertising duration. This is the duration in seconds for which advertising continues after issuing the *adv high* command. Valid range: 0 – 1000 seconds. A value of 0 means advertise forever.

Arguments
- `<duration>`

Default
- `30`

Get example

```
> get bl v h d
R000004
30
```

Set example

```
> set bl v h d 40
R000009
Success
```

bl v h i

Brief Description	Advertising high interval
Access	get/set

Description – Configures high mode advertising interval, used for the *adv* command `high` option. The interval is measured in slots. For example, an interval of 32 means advertise in slot 0, then 32, then 64, and so on. Valid range: 32-8000 slots inclusive.

Arguments
- `<interval>`

Default
- `32`

Get example

```
> get bl v h i
R000004
32
```

Set example

```
> set bl v h i 50
R000009
```

```
Success
```

bl v l d

Brief Description	Advertising low duration
Access	get/set

Description – Configures low mode advertising duration. This is the duration in seconds for which advertising continues after issuing the *adv low* command. Valid range: 0 – 1000 seconds inclusive. A value of 0 means advertise forever.

Arguments
- `<duration>`

Default
- `300`

Get example

```
> get bl v l d
R000004
300
```

Set example

```
> set bl v l d 350
R000004
Success
```

bl v l i

Brief Description	Advertising low interval
Access	get/set

Description – Configures low mode advertising interval, used for the *adv* command `low` option. The interval is measured in slots. For example, an interval of 32 means advertise in slot 0, then 32, then 64, and so on. Valid range: 32-8000 slots inclusive.

Arguments
- `<interval>`

Default
- `1024`

Get example

```
> get bl v l i
R000004
1024
```

Set example

```
> set bl v l i 64
```

```
R000009
Success
```

Bus

bu i

Brief Description	Bus initialization mode
Access	get/set

Description – Serial bus initialization mode. The mode in which the bus initializes on boot up.

Note: Save before reboot, or changes will be lost. See *Serial Interface*.

To control the bus mode of a remote peripheral module, see *rbmode*.

Arguments

- `<stream/command>`

bu i value	Description	Interface in Control
`command`	Initialize to Local COMMAND mode	Serial interface (remote locked out)
`stream`	Initialize to STREAM mode	Serial or Remote interface

Default

- `stream`

Note: The serial bus mode depends on the `mode_sel` GPIO configuration. See *Serial Interface*, *Manual and Automatic Bus Mode Selection*.

Get example

```
> get bu i
R000008
stream
```

Set example

```
> set bu i command
R000009
Success
> save
R000009
Success
```

bu s c

Brief Description	Bus serial control level or edge sensitivity
Access	get/set

Description – The serial bus control variable determines how the serial bus is switched between STREAM mode and COMMAND mode. If `bu s c` is set to `edge`, a rising edge on the `mode_sel` pin toggles

modes. If, however, `bu s c` is set to `level`, the serial bus mode is selected by driving a constant high or low logic level onto the `mode_sel` pin.

The `mode_sel` pin is configured with the *GPIO function* command.

Arguments
- `<edge/level>`

Default
- `level`

Get example

```
> get bu s c
R000006
edge
```

Set example

```
> set bu s c edge
R000009
Success
```

BLE Central

ce c c

Brief Description	Central connection count
Access	get

Description – Returns the number of peripherals connected when acting as a central.

Arguments
- N/A

Default
- N/A

Get example

```
> get ce c c
R000003
0
```

ce c m

Brief Description	Central connection mode
Access	get

Description – Returns the connection mode when connected with a peripheral.

Connection modes are:
- `none`: no connection

- `low`: connected with low connection interval
- `high`: connected with high connection interval

Arguments
- N/A

Default
- N/A

Get example

```
> get ce c m
R000006
none
```

ce s h d

Brief Description	Central scan high duration
Access	get/set

Description – Configures central scan high mode duration. This is the duration in seconds for which scan continues after issuing the *scan high* command. Valid range: 0 – 1000 seconds.

If `<duration>` = 0, scan forever at the specified time.

Arguments
- `<duration>`

Default
- `300`

Get example

```
> get ce s h d
R000004
30
```

Set example

```
> set ce s h d 40
R000009
Success
```

ce s h i

Brief Description	Central scan high interval
Access	get/set

Description – Configures central high mode scanning interval, used for the *scan* command `high` option. Valid range: 96 – 8000.

Arguments

- <interval>

Default
- 96

Get example

```
> get ce s h i
R000004
96
```

Set example

```
> set ce s h i 1000
R000009
Success
```

ce s l d

Brief Description	Advertising low duration
Access	get/set

Description – Configures central low mode scanning duration. This is the duration in seconds for which scanning continues after issuing the *scan low* command. Valid range: 0 – 1000 seconds inclusive.

If <duration> = 0, scan forever at the specified rate.

Arguments
- <duration>

Default
- 300

Get example

```
> get ce s l d
R000004
300
```

Set example

```
> set ce s l d 350
R000009
Success
```

ce s l i

Brief Description	Advertising low interval
Access	get/set

Description – Configures low mode scanning interval, used for the *scan* command low option. Valid range: 96 – 8000.

Arguments
- `<interval>`

Default
- `2048`

Get example

```
> get ce s l i
R000004
2048
```

Set example

```
> set ce s l i 1024
R000009
Success
```

ce s m

Brief Description	Central scan mode
Access	get

Description – With this device acting as a Central, returns scan mode, as set by the *scan* command.

Values:
- `off`: no scan
- `low`: low scan
- `high`: high scan

Arguments
- N/A

Default
- N/A

Get example

```
> get ce s m
R000005
off
```

GPIO

gp u

Brief Description	GPIO usage
Access	get

Description – Get GPIO usage. See the *GPIO function* command for a description of configurable and automatically assigned GPIO functions.

Arguments

- N/A

Default

- N/A

Get example

```
! # Description
# 0 i2c_sda
# 1 i2c_scl
# 2 user_tx
# 3 none
# 4 none
# 5 user_rx
# 6 none, factory
# 7 reserved
# 8 none
# 9 mode_sel
# 10 status_led
# 11 none
# 12 none
# 13 speaker
# 14 none
```

System

sy a t

Brief Description	System activity timeout
Access	get/set

Description – Specify a period of inactivity on the BLE and UART interfaces. After the `system activity timeout` elapses it may trigger power management features as follows.

The `system activity timeout` triggers entry to low power advertising mode. See *Power Management, Low Power Advertising Mode*.

The activity GPIO may be configured to respond to the `system activity timeout`. See the *gfu (GPIO function)* command. If configured, the `activity` GPIO goes high on PoR, and on any UART or BLE activity, and remains high for the duration specified by `system activity timeout`. After no interference activity for the specified duration, the `activity` GPIO goes low. See *Power Management, activity GPIO and system activity timeout*.

Arguments

- `<duration in seconds>`

Default

- `300`

Get example

```
> get sy a t
R000005
300
```

Set example

```
> set sy a t 400
R000009
Success
```

sy b n

Brief Description	System board name
Access	get/set

Description – Configure board name. Maximum length: 16 characters.

Arguments
- <name>

Default
- <board name>

Get example

```
> get sy b n
R000014
AMS001-E01.2
```

Set example

```
> set sy b n LG0051
R000009
Success
```

sy c e

Brief Description	System command echo
Access	get/set

Description – Enable/disable character echo.

Note: If character echo is turned off, keystrokes that are subsequently types are not echoed to the serial interface (or terminal). This mode is primarily intended for machine control.

Arguments
- <0/1>

Default
- 0

Get example

```
> get sy c e
R000003
1
```

Set example

```
> set sy c e 0
R000009
Success
```

sy c h

Brief Description	System command header
Access	get/set

Description – Enable/disable a response header for commands. Only applies to command mode. Response headers make it easy to parse responses with a host MCU.

Arguments
- <0/1>

Default
- 0

Get example

```
> get sy c h
R000003
0
```

Set example

```
> set sy c h 1
R000009
Success
```

sy c m

Brief Description	System command mode
Access	set

Description – Puts the setup app command interface into human or machine mode. See *Serial Interface, Configuration*.

Note: This variable is NOT readable.

Arguments
- <human/machine>

Default
- human

Set example

```
> set sy c m machine
R000009
Success
```

sy c p

Brief Description	System command prompt enabled
Access	get/set

Description – Enable/disable terminal command prompt. Only applies to command mode. A prompt makes it easy for humans to interact with the setup app.

Arguments
- <0/1>

Default
- 1

Get example

```
>get sy c p
R000003
1
```

Set example

```
> set sy c p 1
R000009
Success
```

sy d n

Brief Description	System device name
Access	get/set

Description – Bluetooth device name, up to 8 characters in length. The last 2 to 6 characters from the *BD_ADDR* may be substituted for # wildcards supplied in the final characters of the name.

Arguments
- <name [#]>

Default
- AMS-####

Get example

```
> get bl d
4C55CCABCDEF
> get sy d n
R000010
AMS-CDEF
```

Set example – In the following examples the BD_ADDR address is 4C55CCABCDEF (see *bl a*)

```
> set sy d n ACK-##
R000009
Success
> get sy d n
R000008
ACK-EF
> set sy d n ACKme###
R000009
Success
> get sy d n
R000006
ACKmeDEF
> set sy d n my######
R000009
Success
> get sy d n
R000010
myABCDEF
```

sy i s

Brief Description	System indicator status LED blink behavior
Access	get/set

Description – Set the blink pattern of the system indication status LED. This can be used in power management. See *Power Management*.

Arguments

- `<AABBCCDD>`

where AA, BB, CC, and DD are hex numbers in the range `00` to `7F`.

AABB sets the LED blink pattern when not connected. CCDD sets the LED blink pattern when connected.

The low part of the blink pattern is set with AA when not connected (or CC when connected). The high part of the blink pattern is set with BB when not connected (or DD when connected).

The minimum high or low period is 0.125s (125ms).

Some examples:

Low = AA/CC	High = BB/DD	Period	Duty Cycle	Notes
01	01	0.25s	50%	Fastest possible blink. Duty cycle = 50%
04	04	1.00s	50%	1 Hz – default blink pattern when not connected
00	7f	-	100%	Always on – default blink pattern when connected
1c	04	4.00s	12.5%	period = (28 + 4 = 32) x 0.125, duty cycle = 100*4/(28+4)
7f	7f	31.75s	50%	Slowest possible blink. Duty cycle = 50%

Default
- 0404007f

Get example

```
> get sy i s
0404007f
```

Set example – set to fastest blink when not connected, slowest blink when connected.

```
> set sy i s 01017f7f
Success
```

sy o e

Brief Description	System OTA enabled
Access	get/set

Description – Enable/disable OTA upgrades. 1: enabled; 0: disabled.

Arguments
- <1/0>

Default
- 1

Get example

```
> get sy o e
R000003
1
```

Set example

```
> set sy o e 1
Success
```

sy p

Brief Description	System print level
Access	get/set

Description – System print level. Pint levels:
- 0 – None
- 1 – Synchronous system msgs
- 2 – Synchronous logging msgs
- 3 – Asynchronous system msgs
- 4/all – Asynchronous logging msgs

Arguments
- <[0-4]/[all]>

Default
- all

Get example

```
> get sy p
R000003
0
```

Set example

```
> set sy p 0
R000009
Success
```

sy r e

Brief Description	System remote command enabled
Access	get/set

Description – Enables/disables access to the command interface from a remote terminal via the BLE interface (remote COMMAND mode). If *sy r e* is set to 0, access to the command interface is restricted to the UART interface (local COMMAND mode). See *Serial Interface*.

Arguments
- <1/0>

Default
- 1

Get example

```
> get sy r e
R000003
1
```

Set example

```
> set sy r e 1
Success
```

sy s t

Brief Description	System go-to-sleep timeout
Access	get/set

Description – The module automatically goes to sleep after a timeout period of the specified <seconds>. The timeout countdown restarts when a wake event occurs.

Module go-to-sleep is delayed while data is available on the connection.

The minimum timeout is 10 seconds. This provides time, while the module is awake, to issue a command to set sy s t to 0, which prevents the module cycling back to sleep.

A save and reboot is required before the sleep timeout is enabled.

See *Power Management, Sleep or Wake on Timeout*.

Arguments
- <seconds>

Default
- 0

Get example

```
> get sy s t
R000003
0
```

Set example

```
> set sy s t 10
Success
```

sy u

Brief Description	System UUID
Access	get

Description – Returns the hardware UUID of the module.

Get example

```
> get sy u
R000042
0D06EA434C55CC01B10B1411081309007E615F0A
```

sy v

Brief Description	Firmware version
Access	get

Description – Returns the app firmware version. This is the variable equivalent of the *ver* command.

Arguments
- N/A

Default
- N/A

Get example

```
> get sy v
TruConnect-1.5.0.1, Built: Feb 23 2015 16:50:00, Module: N/A, Board:
AMS001-E01.2
```

sy w t

Brief Description	System wakeup timeout
Access	get/set

Description – The module automatically wakes from sleep after timeout <seconds> from the moment of going to sleep. `0` disables auto-wakeup on a timer.

A save a reboot is required before the sleep timeout is enabled.

See *Power Management, Sleep or Wake on Timeout*.

Arguments
- `<seconds>`
- Range: `0` to `86400`

Set to 0 to disable wakeup.

Default
- `0`

Get example

```
> get sy w t
R000003
0
```

Set example

```
> set sy w t 3600
Success
```

UART

ua b

Brief Description	UART baud
Access	get/set

Description – Sets the UART baud rate. The `<baud rate>` argument must be one of the following standard rates: `9600, 19200, 28800, 38400, 56000, 57600, 115200, 128000, 153600, 230400, 256000, 460800, 921600, 1000000, 1500000.`

Arguments
- `<baud rate>`

Default
- `115200`

Get example

```
> get ua b
```

```
R000008
115200
```

Set example

```
> set ua b 115200
R000009
Success
```

ua f

Brief Description	UART flow
Access	get/set

Description – Turn on/off UART hardware flow control.

Arguments
- <1/0>

Default
- 0

Get example

```
> get ua f
R000003
0
```

Set example

```
> set ua f 0
R000009
Success
```

User Storage

us v

Brief Description	User storage
Access	get/set

Description – Allows storage and retrieval of up to 32 bytes (256 bits) of arbitrary user information.

Arguments
- <user_info>

Default
- N/A

Set example

```
> set us v 32_bytes_of_my_user_information!
```

```
Success
```

Get example

```
> get us v
32_bytes_of_my_user_information!
```