Piksi Settings

Piksi Firmware version v0.15

1 Introduction

Piksi Firmware has a number of settings that can be controlled by the end user via the provided Piksi Console or through the SBP binary message protocol. This Document serves to enumerate these settings with an explanation and any relevant notes.

2 Settings Table

Grouping	Name	Description
base station mode		
	enable	Output surveyed base station location
	surveyed alt	Surveyed altitude of the Piksi's antenna
	surveyed lat	Surveyed latitude of the Piksi's antenna
	surveyed lon	Surveyed longitude of the Piksi's antenna
float kf		
	phase var	Assumed variance of a satellite's phase measurement
	code var	Assumed variance of a satellite's pseudorange measurement
	amb init var	Initial integer ambiguity variance at filter initialization
	new amb var	Variance for new ambiguity measurements
frontend		
	antenna selection	Determines which antenna to use.
iar		
	phase var	Determines the measured carrier phase variance for use in the integer ambiguity resolution test loop
	code var	Determines the pseudocode variance for the integer ambiguity resolution subroutine
sbp		
	obs msg max size	Determines the maximum message length for raw observation sbp messages.
simulator		
	mode mask	Determines the types of position outputs for the simulator.
	radius	Radius of the circle around which the simulated Piksi will move
	base ecef x	Simulated base station position
	base ecef y	Simulated base station position
	base ecef z	Simulated base station position
	speed	Simulated tangential speed of Piksi
	phase sigma	Standard deviation of noise added to the simulated carrier phase
	pseudorange sigma	Standard deviation of noise added to the simulated pseudo range
	cn0 sigma	Standard deviation of noise added to the simulated signal to noise ratio
	speed sigma	Standard deviation of noise addition to simulated tangential speed.
	pos sigma	Standard deviation of simulated single point position
	num sats	The number of satellites for the simulator.
	enabled	Toggles the Piksi internal simulator on and off
solution		
	known baseline d	Determines the baseline vector for the "init known baseline" feature.
	known baseline e	Determines the baseline vector for the "init known baseline" feature.
	known baseline n	Determines the baseline vector for the "init known baseline" feature.
	dgnss solution mode	Determines the type of RTK solution which will be output.
	dgnss filter	Determines the type of carrier phase ambiguity resolution that the Piksi will attempt to achieve.
	output every n obs	Integer divisor of solution frequency for which the observations will be output.
	soln freq	The rate at which a solution is generated internally to the Piksi.
system info		
	firmware built	Date of firmware build
	firmware version	Indicates the firmware version for the Local Piksi

	nap fft index bits	Number of bits to represent the result of fast fourier transform in SwiftNAP firmware
	nap channels	Number of tracking channels in the SwiftNAP firmware
	serial number	The serial number of the Piksi
	nap version	Version of the SwiftNAP FPGA firmware.
	hw revision	hardware revision for Piksi
system monitor		
	heartbeat period milliseconds	Period for sending the SBP HEARTBEAT messages
telemetry radio		
	configuration string	Configuration string to send radio modem over UART when detected
uart ftdi		
	mode	Configure mode for USB serial port on Piksi
	sbp message mask	Configure the message mask for SBP messages on the UART for the USB port on Piksi
	baudrate	The baudrate for the UART for the USB port on Piksi
uart uarta		
	mode	Configure mode for UART
	sbp message mask	Configure the message mask for SBP messages on UART
	configure telemetry radio on boot	Determines whether this UART will attempt to configure a telemetry radio upon boot
	baudrate	The baudrate for the UART
uart uartb		
	mode	Configure mode for UART
	sbp message mask	Configure the message mask for SBP messages on UART
	configure telemetry radio on boot	Determines whether this UART will attempt to configure a telemetry radio upon boot
	baudrate	The baudrate for the uart

Table 2.0.1: Summary of message types

3 Settings Detail

3.1 base station mode

3.1.1 enable

Description: Output surveyed base station location

Label	Value
group	*base station mode*
enumerated possible values	*true, false*
name	*enable*
units	*None*
default value	*false*
type	*boolean*

Table 3.1.1: enable

Notes: This flag ultimately determines whether the SBP message with identifier MSG_BASE_POS will be calculated and sent. Logically, setting this attribute to "true" sets the Local Piksi as a base station and configures the unit to send its surveyed location coordinates to the other Piksi(s) with which the base station is communicating. If "true", the Piksi that receives the base station's surveyed coordinates will also calculate and communicate its own pseudo absolute RTK position based upon the received position of the Base Station.

3.1.2 surveyed alt

Description: Surveyed altitude of the Piksi's antenna

Label	Value
group	*base station mode*
enumerated possible values	*None*
name	*surveyedalt*
units	*meters*
default value	0
type	*Double*

Table 3.1.2: surveyed alt

Notes: This setting represents the altitude of the antenna connected to the Piksi above the WGS84 ellipsoid. This coordinate will be communicated to the Rover against which to calculate a pseudo-absolute position. This value should be precise to 1 cm. Any errors in the surveyed position will directly affect the pseudo-absolute RTK position measurement reported by the Rover.

3.1.3 surveyed lat

Description: Surveyed latitude of the Piksi's antenna

Label	Value
group	*base station mode*
enumerated possible values	*None*
name	*surveyedlat*
units	*degrees*
default value	0
type	*Double*

Table 3.1.3: surveyed lat

Notes: This setting represents the latitude of the connected Piksi's antenna. If "base station mode" is "true", the coordinate will be communicated to the Rover with which to calculate the Rover's pseudo-absolute RTK position. The value should be as accurate as possible and should have precision to at least 7 digits following the decimal point. For reference, 1e-7 degrees of latitude is about 1.1cm on the surface of the earth. Any errors in the surveyed position will directly affect the pseudo-absolute RTK position measurement reported by the Rover.

3.1.4 surveyed lon

Description: Surveyed longitude of the Piksi's antenna

Label	Value
group	*base station mode*
enumerated possible values	*None*
name	*surveyedlon*
units	*degrees*
default value	0
type	*Double*

Table 3.1.4: surveyed lon

Notes: This setting represents the longitude of the connected Piksi's antenna. If "base station mode" is "true", the coordinate will be communicated to the Rover unit with which to calculate the Rover's pseudo-absolute RTK position. This value should be as accurate as possible and should have precision to at least 7 digits following the decimal point. For reference, 1e-7 degrees of longitude at 35 degrees latitude is about 1 cm. Any errors in the surveyed position will directly affect the pseudo-absolute RTK position measurement reported by the Rover.

3.2 float kf

3.2.1 phase var

Description: Assumed variance of a satellite's phase measurement

Label	Value
group	*float kf*
enumerated possible values	*None*
name	*phase var*
units	$cycles^2$
default value	0.0144
type	*Double*

Table 3.2.1: phase var

Notes: This setting adjusts variance estimates in the Swift Kalman filter which aids in integer ambiguity resolution (IAR). Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.2.2 code var

Description: Assumed variance of a satellite's pseudorange measurement

Label	Value
group	*float kf*
enumerated possible values	*None*
name	*code var*
units	$meters^2$
default value	40000
type	*Double*

Table 3.2.2: code var

Notes: This setting adjusts variance estimates in the Swift Kalman filter which aids in integer ambiguity resolution (IAR). Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.2.3 amb init var

Description: Initial integer ambiguity variance at filter initialization

Label	Value
group	*float kf*
enumerated possible values	*None*
name	*amb init var*
units	*nondimensional*
default value	$1.00E+08$
type	*Double*

Table 3.2.3: amb init var

Notes: This setting adjusts variance estimates in the Swift Kalman filter which aids in integer ambiguity resolution (IAR). Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.2.4 new amb var

Description: Variance for new ambiguity measurements

Label	Value
group	*float kf*
enumerated possible values	*None*
name	*new amb var*
units	*nondimensional*
default value	$1.00E+10$
type	*Double*

Table 3.2.4: new amb var

Notes: This setting adjusts variance estimates in the Swift Kalman filter which aids in integer ambiguity resolution (IAR). Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.3 frontend

3.3.1 antenna selection

Description: Determines which antenna to use.

Label	Value
group	*frontend*
enumerated possible values	*Auto, Patch, External*
name	*antenna selection*
units	*None*
default value	*Auto*
type	*enum*

Table 3.3.1: antenna selection

Notes: This setting selects the antenna input that should be used by the Piksi. When set to "Auto", if the unit senses an external antenna attached to the Piksi from a load placed on the antenna output DC bias, it will use the external antenna. If no external antenna is attached (or a passive antenna is attached), it will use the integrated patch antenna. Selecting "Patch" or "External" for this setting can override the automatic antenna selection and force the external or patch antenna to be used.

3.4 iar

3.4.1 phase var

Description: Determines the measured carrier phase variance for use in the integer ambiguity resolution test loop

Label	Value
group	*iar*
enumerated possible values	*None*
name	*phase var*
units	*cycles*^{2}
default value	0.0144
type	*double*

Table 3.4.1: phase var

Notes: This setting adjusts variance estimates in the integer ambiguity resolution (IAR) subroutine. Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.4.2 code var

Description: Determines the pseudocode variance for the integer ambiguity resolution subroutine

Label	Value
group	*iar*
enumerated possible values	*None*
name	*code var*
units	*meters*^{2}
default value	40000
type	*double*

Table 3.4.2: code var

Notes: This setting adjusts variance estimates in the integer ambiguity resolution (IAR) subroutine. Increasing this value can reduce the occurrence of false carrier phase locks but can also increase the time required to achieve an IAR fixed solution. This setting should not be adjusted by end users.

3.5 sbp

3.5.1 obs msg max size

Description: Determines the maximum message length for raw observation sbp messages.

Label	Value
group	*sbp*
enumerated possible values	*None*
name	*obs msg max size*
units	*bytes*
default value	104
type	*integer*

Table 3.5.1: obs msg max size

Notes: This parameter is useful for tuning observation messages for compatibility with radio modems. Some serial modems will internally split serial packets for their protocol and this parameter allows the size of the message to be reduced as to prevent the modem from sending multiple packets. If the parameter exceeds 255 bytes (the maximum size of an SBP message), the Piksi firmware will ignore the parameter and use 255 bytes. If the parameter is set smaller than the size of one observation, the Piksi firmware will ignore the parameter and use the size of one observation as the maximum message size.

3.6 simulator

3.6.1 mode mask

Description: Determines the types of position outputs for the simulator.

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*mode mask*
units	*None*
default value	$15(decimal), 0xF(hexadecimal)$
type	*packedbitfield*

Table 3.6.1: mode mask

Notes: bit 0 (decimal value 1) turns on single point position PVT simulated outputs
bit 1 (decimal value 2) turns on the satellite tracking simulated outputs
bit 2 (decimal value 4) turns on Float IAR simulated RTK outputs
bit 3 (decimal value 8) turns on Fixed IAR simulated RTK outputs

3.6.2 radius

Description: Radius of the circle around which the simulated Piksi will move

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*radius*
units	*meters*
default value	100
type	*double*

Table 3.6.2: radius

Notes: None

3.6.3 base ecef x

Description: Simulated base station position

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*base ecef x*
units	*meters*
default value	*None*
type	*double*

Table 3.6.3: base ecef x

Notes: Earth centered earth fixed (ECEF) x position of the simulated base station.

3.6.4 base ecef y

Description: Simulated base station position

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*base ecef y*
units	*meters*
default value	*None*
type	*double*

Table 3.6.4: base ecef y

Notes: Earth centered earth fixed (ECEF) y position of the simulated base station.

3.6.5 base ecef z

Description: Simulated base station position

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*base ecef z*
units	*meters*
default value	*None*
type	*double*

Table 3.6.5: base ecef z

Notes: Earth centered earth fixed (ECEF) z position of the simulated base station.

3.6.6 speed

Description: Simulated tangential speed of Piksi

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*speed*
units	*meters/s*
default value	4
type	*double*

Table 3.6.6: speed

Notes: None

3.6.7 phase sigma

Description: Standard deviation of noise added to the simulated carrier phase

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*phase sigma*
units	*cycles*
default value	0.0009
type	*double*

Table 3.6.7: phase sigma

Notes: None

3.6.8 pseudorange sigma

Description: Standard deviation of noise added to the simulated pseudo range

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*pseudorange sigma*
units	*meters*
default value	16
type	*double*

Table 3.6.8: pseudorange sigma

Notes: None

3.6.9 cn0 sigma

Description: Standard deviation of noise added to the simulated signal to noise ratio

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*cn0 sigma*
units	*dbmhz*
default value	0.1
type	*double*

Table 3.6.9: cn0 sigma

Notes: None

3.6.10 speed sigma

Description: Standard deviation of noise addition to simulated tangential speed.

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*speed sigma*
units	$meters^2/s^2$
default value	0.02
type	*double*

Table 3.6.10: speed sigma

Notes: None

3.6.11 pos sigma

Description: Standard deviation of simulated single point position

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*pos sigma*
units	$meters^2$
default value	2
type	*double*

Table 3.6.11: pos sigma

Notes: None

3.6.12 num sats

Description: The number of satellites for the simulator.

Label	Value
group	*simulator*
enumerated possible values	*None*
name	*num sats*
units	*None*
default value	9
type	*integer*

Table 3.6.12: num sats

Notes: None

3.6.13 enabled

Description: Toggles the Piksi internal simulator on and off

Label	Value
group	*simulator*
enumerated possible values	*true*, *false*
name	*enabled*
units	*None*
default value	*false*
type	*boolean*

Table 3.6.13: enabled

Notes: The Piksi simulator will provide simulated outputs of a stationary base station and the Local Piksi moving in a circle around the base station The simulator is intended to aid in system integration by providing realistic looking outputs but does not faithfully simulate every aspect of device operation.

3.7 solution

3.7.1 known baseline d

Description: Determines the baseline vector for the "init known baseline" feature.

Label	Value
group	*solution*
enumerated possible values	*None*
name	*known baseline d*
units	*meters(down)*
default value	0
type	*double*

Table 3.7.1: known baseline d

Notes: This sets the number of meters that the rover is Down from the base station when the "init known baseline" feature is used.

3.7.2 known baseline e

Description: Determines the baseline vector for the "init known baseline" feature.

Label	Value
group	*solution*
enumerated possible values	*None*
name	*known baseline e*
units	*meters*(*east*)
default value	0
type	*double*

Table 3.7.2: known baseline e

Notes: This sets the number of meters that the rover is East from the base station when the "init known baseline" feature is used.

3.7.3 known baseline n

Description: Determines the baseline vector for the "init known baseline" feature.

Label	Value
group	*solution*
enumerated possible values	*None*
name	*known baseline n*
units	*meters*(*north*)
default value	0
type	*double*

Table 3.7.3: known baseline n

Notes: This sets the number of meters that the rover is North from the base station when the "init known baseline" feature is used.

3.7.4 dgnss solution mode

Description: Determines the type of RTK solution which will be output.

Label	Value
group	*solution*
enumerated possible values	*LowLatency*, *TimeMatched*
name	*dgnss solution mode*
units	*None*
default value	*None*
type	*enum*

Table 3.7.4: dgnss solution mode

Notes: A "Low Latency" solution uses an internal model of anticipated satellite observations to provide RTK output with minimal latency but slightly reduced accuracy. "Low Latency" mode assumes that the base station is stationary. For applications where accuracy is desired over timelieness or when both Piksi's are moving, "Time matched" mode can be chosen. This means that the RTK output will require a corresponding set of correction observations for each timestamp.

3.7.5 dgnss filter

Description: Determines the type of carrier phase ambiguity resolution that the Piksi will attempt to achieve.

Label	Value
group	*solution*
enumerated possible values	*Fixed, Float*
name	*dgnss filter*
units	*None*
default value	*Fixed*
type	*enum*

Table 3.7.5: dgnss filter

Notes: If "fixed", the Piksi will output a integer fixed ambiguity estimate. If no fixed solution is available, it will revert to the float solution. If "float", the device will only output the float ambiguity estimate.

3.7.6 output every n obs

Description: Integer divisor of solution frequency for which the observations will be output.

Label	Value
group	*solution*
enumerated possible values	*None*
name	*output every n obs*
units	*None*
default value	2
type	*integer*

Table 3.7.6: output every n obs

Notes: For instance, if the solution frequency is 10 hz, and the "output every n obs" parameter is 2, it means that the observation output will occur at a rate of 5hz. Since the observations are the information used by the Piksi receiving corrections from the connected Piksi, this determines the rate of information sharing for RTK solution output. This parameter is designed to tune the rate at which correction information is passed from one Piksi to the other as to efficiently use radio modem bandwidth and fit with user applications.

3.7.7 soln freq

Description: The rate at which a solution is generated internally to the Piksi.

Label	Value
group	*solution*
enumerated possible values	*None*
name	*soln freq*
units	*hz*
default value	10
type	*integer*

Table 3.7.7: soln freq

Notes: None

3.8 system info

3.8.1 firmware built

Description: Date of firmware build

Label	Value
group	*system info*
enumerated possible values	*None*
name	*firmware built*
units	*None*
default value	*None*
type	*string*

Table 3.8.1: firmware built

Notes: None

3.8.2 firmware version

Description: Indicates the firmware version for the Local Piksi

Label	Value
group	*system info*
enumerated possible values	*None*
name	*firmware version*
units	*None*
default value	*None*
type	*string*

Table 3.8.2: firmware version

Notes: For user generated firmware, this information will appear the same as the git command: "git describe –dirty"

3.8.3 nap fft index bits

Description: Number of bits to represent the result of fast fourier transform in SwiftNAP firmware

Label	Value
group	*system info*
enumerated possible values	*None*
name	*nap fft index bits*
units	*None*
default value	*None*
type	*None*

Table 3.8.3: nap fft index bits

Notes: None

3.8.4 nap channels

Description: Number of tracking channels in the SwiftNAP firmware

Label	Value
group	*system info*
enumerated possible values	*None*
name	*nap channels*
units	*None*
default value	*None*
type	*integer*

Table 3.8.4: nap channels

Notes: None

3.8.5 serial number

Description: The serial number of the Piksi

Label	Value
group	*system info*
enumerated possible values	*None*
name	*serial number*
units	*None*
default value	*None*
type	*integer*

Table 3.8.5: serial number

Notes: This number should match the number on the barcode and cannot be modified

3.8.6 nap version

Description: Version of the SwiftNAP FPGA firmware.

Label	Value
group	*system info*
enumerated possible values	*None*
name	*nap version*
units	*None*
default value	*None*
type	*integer*

Table 3.8.6: nap version

Notes: None

3.8.7 hw revision

Description: hardware revision for Piksi

Label	Value
group	*system info*
enumerated possible values	*None*
name	*hw revision*
units	*None*
default value	*None*
type	*string*

Table 3.8.7: hw revision

Notes: None

3.9 system monitor

3.9.1 heartbeat period milliseconds

Description: Period for sending the SBP_HEARTBEAT messages

Label	Value
group	*system monitor*
enumerated possible values	*None*
name	*heartbeat period milliseconds*
units	*None*
default value	*None*
type	*integer*

Table 3.9.1: heartbeat period milliseconds

Notes: None

3.10 telemetry radio

3.10.1 configuration string

Description: Configuration string to send radio modem over UART when detected

Label	Value
group	*telemetry radio*
enumerated possible values	*None*
name	*configuration string*
units	*None*
default value	*None*
type	*string*

Table 3.10.1: configuration string

Notes: This configuration string is intended for radios that use AT style commands

3.11 uart ftdi

3.11.1 mode

Description: Configure mode for USB serial port on Piksi

Label	Value
group	*uart ftdi*
enumerated possible values	*SBP, NMEA, RTCM*
name	*mode*
units	*None*
default value	*SBP*
type	*enum*

Table 3.11.1: mode

Notes: None

3.11.2 sbp message mask

Description: Configure the message mask for SBP messages on the UART for the USB port on Piksi

Label	Value
group	*uart ftdi*
enumerated possible values	*None*
name	*sbp message mask*
units	*None*
default value	65535(*decimal*), 0x*FFFF*(*hex*)
type	*integer*

Table 3.11.2: sbp message mask

Notes: The message mask is bitwise anded to the message identifier for a particular message. If the result is non-zero, the message will be sent over this UART. For example, consider the Piksi firmware sending an SBP message with ID 0x0041. If UART A has mask "64" (0x0040), The SBP subsystem bitwise-ands the message id with the UART A mask giving the result of 0x0040. Since the result is non-zero, the message is valid for UART A and is sent. Practically, the UART with mask 64 (0x0040) transmits only RTK observation data and the USART with mask 65280 (0xFF00) transmits most messages of interest to the host system (such as position and velocity). A mask of 0xFFFF will transmit all messages at the expense of bandwidth.

3.11.3 baudrate

Description: The baudrate for the UART for the USB port on Piksi

Label	Value
group	*uart ftdi*
enumerated possible values	*None*
name	*baudrate*
units	*baud*
default value	1000000
type	*integer*

Table 3.11.3: baudrate

Notes: None

3.12 uart uarta

3.12.1 mode

Description: Configure mode for UART

Label	Value
group	*uart uarta*
enumerated possible values	*SBP, NMEA, RTCM*
name	*mode*
units	*None*
default value	*SBP*
type	*enum*

Table 3.12.1: mode

Notes: None

3.12.2 sbp message mask

Description: Configure the message mask for SBP messages on UART

Label	Value
group	*uart uarta*
enumerated possible values	*None*
name	*sbp message mask*
units	*None*
default value	64(*decimal*), 0x0040(*hex*)
type	*integer*

Table 3.12.2: sbp message mask

Notes: The default message mask on this UART (0x0040) is appropriate for a radio to communicate observation messages to another Piksi. The out-of-the box configuration uses UART A for Piksi to Piksi communication.

3.12.3 configure telemetry radio on boot

Description: Determines whether this UART will attempt to configure a telemetry radio upon boot

Label	Value
group	*uart uarta*
enumerated possible values	*true*, *false*
name	*configure telemetry radio on boot*
units	*None*
default value	*TRUE*
type	*boolean*

Table 3.12.3: configure telemetry radio on boot

Notes: If a telemetry radio is connected to this UART, this should be set to true in order to send the configuration string to the radio.

3.12.4 baudrate

Description: The baudrate for the UART

Label	Value
group	*uart uarta*
enumerated possible values	*None*
name	*baudrate*
units	*baud*
default value	115200
type	*integer*

Table 3.12.4: baudrate

Notes: The radio baudrate may be constrained by the particular RF equipment used for the telemetry radio.

3.13 uart uartb

3.13.1 mode

Description: Configure mode for UART

Label	Value
group	*uart uartb*
enumerated possible values	*SBP, NMEA, RTCM*
name	*mode*
units	*None*
default value	*SBP*
type	*enum*

Table 3.13.1: mode

3.13.2 sbp message mask

Description: Configure the message mask for SBP messages on UART

Label	Value
group	*uart uartb*
enumerated possible values	*None*
name	*sbp message mask*
units	*None*
default value	655280(*decimal*), 0x*FF*00(*hex*)
type	*integer*

Table 3.13.2: sbp message mask

Notes: The default message mask on this uart (0xFF00) is appropriate for a general purpose interface to the Piksi.

3.13.3 configure telemetry radio on boot

Description: Determines whether this UART will attempt to configure a telemetry radio upon boot

Label	Value
group	*uart uartb*
enumerated possible values	*true, false*
name	*configure telemetry radio on boot*
units	*None*
default value	*TRUE*
type	*boolean*

Table 3.13.3: configure telemetry radio on boot

Notes: If a telemetry radio is connected to this UART, this should be set to true in order to send the configuration string to the radio.

3.13.4 baudrate

Description: The baudrate for the uart

Label	Value
group	*uart uartb*
enumerated possible values	*None*
name	*baudrate*
units	*baud*
default value	115200
type	*integer*

Table 3.13.4: baudrate

Notes: None