RF Application Command Reference for SAMR34

Microchip Technology, Inc.

REV	DATE	ORIGINATOR	DESCRIPTION OF CHANGE
0.1	03/01/18	WSG Team	Initial Draft
0.2	24/05/18	WSG Team	Added RSSI Get command and Sleep commands

Table of Contents

1 Introduction

This document summarizes RF Application Commands for SAMR34. This application is created from breaking parser application radio commands to a separate application.

2 Command Organization

There are two command categories as shown in below table.

Command Type	Keyword	Description
System	<sys>	Issues system level behavior actions – Sleep and Reset
Transceiver commands	<radio>	Issues radio specific configurations, directly accessing and updating the transceiver setup.

Note: Upon successful reception of commands, the module will respond with one of the following:

- `ok`
- `invalid_param`
- **Requested Information**
- **Descriptive Error Message**

3 System Commands

System commands begin with the system keyword <sys> and include the categories

Parameter	Description
sleep	Puts the system in Sleep for a finite number of milliseconds.
reset	Resets and restarts the SAMR34 module.
factoryRESET	Resets the SAMR34 module's configuration data and user EEPROM to factory default values and restarts the SAMR34 module.
sys get ver	Returns the version-related information

3.1.1 sys sleep <sleepmode>

3.1.1.1 <sleepmode>:

Type of sleep mode the SAMR34 will put to.

Sleepmode	Wakeup source
standby	Userbutton will wakeup the device from sleep
backup	Reset
off	Reset

3.1.1.2 Response:

```
System can sleep for <x> ms
Entering Standby Sleep
```

System will be in sleep for the maximum time it can. For example if any software timer is running /transaction is ongoing, the time will specify the sleep period.

3.1.1.3 Description:

This command puts the system to different sleep mode available in SAMR34

3.1.1.4 Example:

sys sleep standby // Puts the system to standby sleep

System can sleep for 4294967 ms
Entering Standby Sleep

3.1.2 sys reset

3.1.2.1 Response:

SAMR34 Xpro MLS_SDK_ MAJORNUM_MINORNUM_BUILDYPE_BUILDNUM MMM DD YYYY HH:MM:SS, where MMM is month, DD is the day, HH:MM:SS is hour, minutes, seconds (format: [HW] [FW] [Date] [Time]). [Date] and [Time] refer to the release of the firmware.

3.1.2.2 Description:

This command resets and restarts the SAMR34 module; stored internal configurations will be loaded automatically upon reboot.

3.1.2.3 Example:

`sys reset` // Resets and restarts the SAMR34 module.

3.1.3 sys factoryRESET

3.1.3.1 Response:

SAMR34 Xpro MLS_SDK_ MAJORNUM_MINORNUM_BUILDYPE_BUILDNUM MMM DD YYYY HH:MM:SS, where MMM is month, DD is the day, HH:MM:SS is hour, minutes, seconds (format: [HW] [FW] [Date] [Time]). [Date] and [Time] refer to the release of the firmware.

.

3.1.3.2 Description:

This command resets the module's configuration data and user EEPROM to factory default values and restarts the module. After `factoryRESET`, the RN2483 module will automatically reset and all configuration parameters are restored to factory default values.

3.1.3.3 Example:

`sys factoryRESET` // Restores factory default values

3.1.4 sys get ver

3.1.4.1 Response:

`SAMR34 Xpro MLS_SDK_X_Y_E/P_Z MMM DD YYYY HH:MM:SS`, where X.Y.Z is firmware version, E- Engineering and P-Public release , MMM is month, DD is day, HH:MM:SS is hour, minutes, seconds (format: [HW] [FW] [Date] [Time]). [Date] and [Time] refer to the release of the firmware.

3.1.4.2 Description:

This command returns the information related to the hardware platform, firmware version, release date and time stamp on firmware creation.

3.1.4.3 Example:

`sys get ver` // Returns version-related information.

4 Radio Commands

There are two command categories as shown in below table.

Parameter	Description
rx	This command configures the radio to receive simple radio packets continuously according to prior configuration settings.
tx	This command configures a simple radio packet transmission multiple times according to prior configuration settings.

cw	This command will put the module into a Continuous Wave (cw) Transmission for system tuning or certification use.
set	This command allows modification to the radio setting directly. This command allows for the user to change the method of radio operation within module type band limits.
get	This command grants the ability to read out radio settings as they are currently configured.

RADIO PARAMETERS AVAILABILITY FOR DIFFERENT OPERATIONS

Command	radio get	radio set	Availability for LoRa® Modulation	Availability for FSK Modulation
bt	√	√	—	√
mod	√	√	√	√
freq	√	√	√	√
pwr	√	√	√	√
sf	√	√	√	—
afcbw	√	√	—	√
rxbw	√	√	—	√
bitrate	√	√	—	√
fdev	√	√	—	√
prlen	√	√	—	√
crc	√	√	√	√
iqi	√	√	√	—
cr	√	√	√	—
wdt	√	√	√	√
sync	√	√	√	√
bw	√	√	√	—
snr	√	—	√	—

4.1 Radio Transmit/Receive Commands

4.1.1 radio rx <rxWindowSize>

4.1.1.1 <rxWindowSize>:

decimal number representing the number of symbols (for LoRa modulation) or time-out (in milliseconds, for FSK modulation) that the receiver will be opened, from 0 to 65535. Set `<rxWindowSize>` to '0' in order to enable the Continuous Reception mode. Continuous Reception mode will not be exited after a valid packet is received. Radio will be in Continuous reception mode always.

4.1.1.2 Description:

This command may reply with two responses. The first response will be received immediately after entering the command. If the command is valid (`ok` reply received), a second reply will be received no. of times after the no. of packets received or after the time-out occurred.
Response after entering the command:
• `ok` – if parameter is valid and the transceiver is configured in Receive mode
• `invalid_param` – if parameter is not valid
• `busy` – if the transceiver is currently busy
Response after the receive process:
• `radio_rx <data>` – if reception was successful, `<data>`: hexadecimal value that was received;
• `radio_err` – if reception was not successful, reception time-out occurred

4.1.1.3 Example:

`radio rx 0` // Puts the radio into continuous Receive mode always.
radio rx hello
radio rx hello
radio rx hello
radio rx hello
radio rx hello
radio rx hello

4.1.2 radio tx <data> <count>

4.1.2.1 <data>:

hexadecimal value representing the data to be transmitted, from 0 to 255 bytes for LoRa modulation and from 0 to 64 bytes for FSK modulation.

4.1.2.2 <count>:

decimal value representing the count of the data to transmitted multiple times from 0 to 65535 bytes for LoRa modulation and for FSK modulation.

4.1.2.3 Response:

this command may reply with two responses. The first response will be

received immediately after entering the command. If the command is valid (`ok` reply received), a second reply will be received no. of times as per the count value after the no. of effective transmission. If the count value is '0', a second reply will be received one time after the effective transmission, transmission will happen one time.

Response after entering the command:

- `ok` – if parameter is valid and the transceiver is configured in Transmit mode
- `invalid_param` – if parameter is not valid
- `busy` – if the transceiver is currently busy

Response after the effective transmission:

- `radio_tx_ok` – if transmission was successful and transmission will be repeated until it reaches the count value
- `radio_err` – if transmission was unsuccessful (interrupted by radio Watchdog Timer time-out)

This command transmits the `<data>` passed no. of times as per the count value given.

4.1.2.4 Example:

`radio tx 48656c6C6F 5`// Transmits a packet of

```
[0x48][0x65][0x6c][0x6C][0x6F];
[0x48][0x65][0x6c][0x6C][0x6F];
[0x48][0x65][0x6c][0x6C][0x6F];
[0x48][0x65][0x6c][0x6C][0x6F];
[0x48][0x65][0x6c][0x6C][0x6F];
[0x48][0x65][0x6c][0x6C][0x6F];
Hello.
Hello.
Hello.
Hello.
Hello.
Hello.
```

Count value as '0'

Example2: **`radio tx 48656c6C6F 0`**// Transmits a packet of

```
[0x48][0x65][0x6c][0x6C][0x6F];
Hello.
```

Note: In order to meet ETSI regulations in the given frequency bands, the radio has to use either Listen Before Talk (LBT) + Adaptive Frequency Agility (AFA) or duty cycle limitations. By issuing the `radio tx <data>` command the module does not perform LBT before transmission, thus the user has to make sure that duty cycle limits are not violated. For more information on duty cycle limits

Note: When transmitting FSK packets, the payload and the 2-byte CRC is whitened by being XORed with a pseudorandom sequence generated by an LFSR with the polynomial X9 + X5 + 1. This process is automatically reverted on reception so that it is transparent to the user.

4.1.3 radio cw <state>

4.1.3.1 <state>:

string representing the state of the Continuous Wave (CW) mode, either `on` or `off`.

4.1.3.2 Response:

`ok` if state is valid
`invalid_param` if state is not valid

4.1.3.3 Description:

This command will enable or disable the CW mode on the module. CW mode allows the user to put the transceiver into Transmission mode to observe the generated signal. By altering the settings for the radio the user can observe the changes in transmissions levels.

4.1.3.4 Example:

```
Radio cw on
```

Note: Please note that using `radio cw off` resets the module, this command being semantically identical to `sys reset`.

4.2 Radio Set Commands

There are the radio set commands as shown in below table.

Parameter	Description
bt	Set the data shaping for frequency shift keying (FSK) modulation type.
mod	Set the module Modulation mode.
freq	Set the current operation frequency for the radio.
pwr	Set the output power level used by the radio during transmission.
sf	Set the requested spreading factor (SF) to be used during transmission.
afcbw	Set the value used by the automatic frequency correction bandwidth.
rxbw	Set the operational receive bandwidth.
bitrate	Set the frequency shift keying (FSK) bit rate.
fdev	Set the frequency deviation allowed by the end device.

prlen	Set the preamble length used during transmissions.
crc	Set if a CRC header is to be used.
iqi	Set if IQ inversion is used.
cr	Set the coding rate used by the radio.
wdt	Set the time-out limit for the radio Watchdog Timer.
sync	Set the sync word used.
bw	Set the value used for the radio bandwidth.
reg	Set the register value to a particular address

4.2.1 radio set bt <gfBT>

4.2.1.1 <gfBT>:

string representing the Gaussian baseband data shaping, enabling GFSK modulation. Parameter values can be: `none`, `1.0`, `0.5`, `0.3`.

4.2.1.2 Response:

`ok` if the data shaping is valid
`invalid_param` if the data shaping is not valid

4.2.1.3 Description:

This command modifies the data shaping applied to FSK transmissions. Entering any `<gfBT>` other than `none` will result in a Gaussian Filter BT being applied to transmissions in FSK mode.

4.2.1.4 Example:

`radio set bt none` // Data shaping in FSK mode is disabled or null.

4.2.2 radio set mod <mode>

4.2.2.1 <mode>:

string representing the modulation method, either `lora` or `fsk`.

4.2.2.2 Response:

`ok` if the modulation is valid
`invalid_param` if the modulation is not valid

4.2.2.3 Description:

This command changes the modulation method being used by the module. Altering the mode of operation does not affect previously set parameters, variables or

registers. FSK mode also allows GFSK transmissions when data shaping is enabled.

4.2.2.4 Example:

```
radio set mod lora
```

4.2.3 radio set freq <frequency>

4.2.3.1 <frequency>:

decimal representing the frequency, from 137000000 to 175000000 or from 410000000 to 525000000 or from 862000000 to 1020000000, in Hz.

4.2.3.2 Response:

`ok` if the frequency is valid
`invalid_param` if the frequency is not valid

4.2.3.3 Description:

This command changes the communication frequency of the radio transceiver.

4.2.3.4 Example:

```
radio set freq 868000000
```

4.2.4 radio set pwr <pwrout>

4.2.4.1 <pwrout>:

signed decimal number representing the transceiver output power, from -4 to 17.

4.2.4.2 Response:

`ok` if the output power is valid
`invalid_param` if the output power is not valid

4.2.4.3 Description:

This command changes the transceiver output power. However, note that the transceiver is designed to transmit a maximum of +14 dBm. It is possible to set the output power above the regulatory limits. This power setting allows some compensation on the cable or transmission line loss.

4.2.4.4 Example:

```
radio set pwr 14
```

4.2.5 radio set sf <spreadingfactor>

4.2.5.1 <spreadingFactor>:

string representing the spreading factor. Parameter values can be: `sf7`, `sf8`, `sf9`, `sf10`, `sf11` or `sf12`

4.2.5.2 Response:

`ok` if the spreading factor is valid
`invalid_param` if the spreading factor is not valid

4.2.5.3 Description:

This command sets the spreading factor used during transmission.

4.2.5.4 Example:

```
radio set sf sf7
```

4.2.6 radio set afcbw <autoFreqBand >

4.2.6.1 <autoFreqBand>:

float representing the automatic frequency correction, in kHz. Parameter values can be: 250, 125, 62.5, 31.3, 15.6, 7.8, 3.9, 200,100, 50, 25, 12.5, 6.3, 3.1, 166.7, 83.3, 41.7, 20.8, 10.4, 5.2, 2.6.

4.2.6.2 Response:

`ok` if the automatic frequency correction is valid
`invalid_param` if the automatic frequency correction is not valid

4.2.6.3 Description:

This command modifies the automatic frequency correction bandwidth for receiving/transmitting.

4.2.6.4 Example:

```
radio set afcbw 125
```

4.2.7 radio set rxbw <rxbandwidth>

4.2.7.1 <rxbandwidth>:

float representing the signal bandwidth, in kHz. Parameter values can be: 250, 125, 62.5, 31.3, 15.6, 7.8, 3.9, 200, 100, 50, 25, 12.5, 6.3, 3.1, 166.7, 83.3, 41.7, 20.8, 10.4, 5.2, 2.6.

4.2.7.2 Response:

`ok` if the signal bandwidth is valid
`invalid_param` if signal bandwidth is not valid

4.2.7.3 Description:

This command sets the signal bandwidth when receiving.

4.2.7.4 Example:

`radio set rxbw 250` // Signal bandwidth for receiving is 250 kHz.

4.2.8 radio set bitrate <fskBitRate>

4.2.8.1 <fskBitRate>:

decimal number representing the FSK bit rate value, from 1 to 300000.

4.2.8.2 Response:

`ok` if the bit rate value is valid
`invalid_param` if the bit rate value is not valid

4.2.8.3 Description:

This command sets the FSK bit rate value.

4.2.8.4 Example:

`radio set bitrate 5000` // FSK bit rate is set to 5 kb/s.

4.2.9 radio set fdev <freqdev>

4.2.9.1 <freqdev>:

decimal number representing the frequency deviation, from 0 to 200000.

4.2.9.2 Response:

`ok` if the frequency deviation is valid
`invalid_param` if frequency deviation is not valid

4.2.9.3 Description:

This command sets the frequency deviation during operation.

4.2.9.4 Example:

`radio set fdev 5000` // Frequency deviation is 5 kHz.

4.2.10 radio set prlen <preamble>

4.2.10.1 <prlen>:

decimal number representing the preamble length, from 0 to 65535.

4.2.10.2 Response:

`ok` if the preamble length is valid
`invalid_param` if the preamble length is not valid

4.2.10.3 Description:

This command sets the preamble length for transmit/receive.

4.2.10.4 Example:

`radio set prlen 8` // Preamble length is 8.

4.2.11 radio set crc <crcHeader>

4.2.11.1 <crcHeader>:

string representing the state of the CRC header, either `on` or `off`.

4.2.11.2 Response:

`ok` if the state is valid
`invalid_param` if the state is not valid

4.2.11.3 Description:

This command enables or disables the CRC header for communications.

4.2.11.4 Example:

`radio set crc on` // Enables the CRC header.

4.2.12 radio set iqi <iqInvert>

4.2.12.1 <iqInvert>:

string representing the state of the invert IQ, either `on` or `off`.

4.2.12.2 Response:

`ok` if the state is valid
`invalid_param` if the state is not valid

4.2.12.3 Description:

This command enables or disables the Invert IQ for communications

4.2.12.4 Example:

`radio set iqi on` // Invert IQ is enabled.

4.2.13 radio set cf <codingRate>

4.2.13.1 <codingRate>:

string representing the coding rate. Parameter values can be: `4/5`, `4/6`, `4/7`, `4/8`.

4.2.13.2 Response:

`ok` if the coding rate is valid
`invalid_param` if the coding rate is not valid

4.2.13.3 Description:

This command modifies the coding rate currently being used by the radio.

4.2.13.4 Example:

`radio set cr 4/7` // The coding rate is set to 4/7.

4.2.14 radio set wdt <watchDog>

4.2.14.1 <watchDog>:

decimal number representing the time-out length for the Watchdog Timer, from 0 to 4294967295. Set to '0' to disable this functionality.

4.2.14.2 Response:

`ok` if the watchdog time-out is valid
`invalid_param` if the watchdog time-out is not valid

4.2.14.3 Description:

This command updates the time-out length, in milliseconds, applied to the radio Watchdog Timer. If this functionality is enabled, then the Watchdog Timer is started for every transceiver reception or transmission. The Watchdog Timer is stopped when the operation in progress in finished.

4.2.14.4 Example:

`radio set wdt 2000` // The Watchdog Timer is configured for 2000 ms.

Note: Ensure the value configured for the Watchdog Timer matches the radio configurations. For example, set the `<watchDog>` value to '0' in order to disable this functionality during the radio continuous reception.

4.2.15 radio set sync <syncWord>

4.2.15.1 <syncWord>:

hexadecimal value representing the Sync word used during communication. For LoRa modulation one byte is used, for FSK up to eight bytes can be entered.

4.2.15.2 Response:

`ok` if the sync word is valid
`invalid_param` if the sync word is not valid

4.2.15.3 Description:

This command configures the sync word used during communication.

4.2.15.4 Example:

`radio set sync 12` // Set the sync word to a single byte with the value 0x12.

4.2.16 radio set bw <bandWidth>

4.2.16.1 <bandWidth>:

decimal representing the operating radio bandwidth, in kHz.
Parameter values can be: 125, 250, 500.

4.2.16.2 Response:

`ok` if the bandwidth is valid
`invalid_param` if the bandwidth is not valid

4.2.16.3 Description:

This command sets the operating radio bandwidth for LoRa operation.

4.2.16.4 Example:

`radio set bw 250` // The operating bandwidth is 250 kHz.

4.2.17 radio set pa <paboost>

4.2.17.1 <paboost>:

string represents the state of the PABOOST, either on or off.

4.2.17.2 Response:

`ok` if the state is valid
`invalid_param` If the state is not valid

4.2.17.3 Description:

This command enables the PABOOST to use maximum power for radio operation

4.2.17.4 Example:

`radio set pa on` // Enables the PABOOST.

4.2.18 radio set reg <regAddr> <regValue>

4.2.18.1 <regAddr>:

hexadecimal value representing the address of radio register

4.2.18.2 <regValue>:

hexadecimal value representing the value to be written to **regAddr**

4.2.18.3 Response:

`ok` if the parameters are valid
`invalid_param` If the regaddr & regvalue is not valid

4.2.18.4 Description:

This command writes the given value to a radio register

Example**:**
`radio set reg 02 05` // Sets the value 0x05 to a radio register 0x02

4.3 Radio Get Commands

There are the radio set commands as shown in below table.

Parameter	Description
bt	Get the data shaping for frequency shift keying (FSK) modulation type.
mod	Get the module Modulation mode.
freq	Get the current operation frequency for the radio.
pwr	Get the output power level used by the radio during transmission.
sf	Get the requested spreading factor (SF) to be used during transmission.
afcbw	Get the value used by the automatic frequency correction bandwidth.
rxbw	Get the operational receive bandwidth.
bitrate	Get the frequency shift keying (FSK) bit rate.
fdev	Get the frequency deviation allowed by the end device.
prlen	Get the preamble length used during transmissions.
crc	Get if a CRC header is to be used.
iqi	Get if IQ inversion is used.
cr	Get the coding rate used by the radio.
wdt	Get the time-out limit for the Watchdog Timer.
bw	Get the value used for the radio bandwidth.

snr	Get the signal noise ratio (SNR) of the last received packet.
sync	Get the synchronization word used for communication.
reg	Get the register value of a particular address if reg is to be used
regdump	Get the register value from starting address to ending address used by the radio
pktrssi	Get the rssi value of last received packet (Value in dBm)

4.3.1 radio get bt

4.3.1.1 Response:

string representing the configuration for data shaping. Parameter values can be: `none`, `1.0`, `0.5`, `0.3`.

4.3.1.2 Description:

This command reads back the current configuration for data shaping applied to FSK transmissions.
Default: `0.5`

4.3.1.3 Example:

`radio get bt` // Reads the current data shaping FSK configuration.

4.3.2 radio get mod

4.3.2.1 Response:

string representing the current mode of operation of the module, either `lora` or `fsk`.

4.3.2.2 Description:

This command reads back the current mode of operation of the module.
Default: `lora`

4.3.2.3 Example:

`radio get mod` // Reads if module is modulating in LoRa or FSK.

.

4.3.3 radio get freq

4.3.3.1 Response:

decimal representing the frequency, from 137000000 to 175000000 or from 410000000 to 525000000 or from 862000000 to 1020000000, in Hz.

4.3.3.2 Description:

This command reads back the current operation frequency of the module.
Default: 868100000

4.3.3.3 Example:

`radio get freq` // Reads back the current frequency the transceiver communicates on.

4.3.4 radio get pwr

4.3.4.1 Response:

signed decimal representing the current power level, from -4to 17.

4.3.4.2 Description:

This command reads back the current power level settings used in operation.
Default: 1

4.3.4.3 Example:

`radio get pwr` // Reads back the current transmit output power.

4.3.5 radio get sf

4.3.5.1 Response:

string representing the current spreading factor.

4.3.5.2 Description:

This command reads back the current spreading factor being used by the transceiver. Parameter values can be: `sf7`, `sf8`, `sf9`, `sf10`, `sf11`, `sf12`
Default: `sf7`

4.3.5.3 Example:

`radio get sf` // Reads back the current spreading factor settings.

4.3.6 radio get afcbw

4.3.6.1 Response:

float representing the automatic frequency correction band, in kHz.
Parameter values can be: 250, 125, 62.5, 31.3, 15.6, 7.8, 3.9, 200, 100, 50, 25, 12.5, 6.3, 3.1, 166.7, 83.3, 41.7, 20.8, 10.4, 5.2, 2.6.

4.3.6.2 Description:

This command reads back the status of the Automatic Frequency Correction Bandwidth.
Default: 41.7

4.3.6.3 Example:

`radio get afcbw` // Reads back the current automatic frequency correction bandwidth.

4.3.7 radio get rxbw

4.3.7.1 Response:

float representing the signal bandwidth, in kHz. Parameter values can be: 250, 125, 62.5, 31.3, 15.6, 7.8, 3.9, 200, 100, 50, 25, 12.5, 6.3, 3.1, 166.7, 83.3, 41.7, 20.8, 10.4, 5.2, 2.6.

4.3.7.2 Description:

This command reads back the signal bandwidth used for receiving.
Default: 25

4.3.7.3 Example:

`radio get rxbw` // Reads back the receive signal bandwidth.

4.3.8 radio get bitrate

4.3.8.1 Response:

signed decimal representing the configured bit rate, from 1 to 300000.

4.3.8.2 Description:

This command reads back the configured bit rate for FSK communications.
Default: 50000

4.3.8.3 Example:

`radio get bitrate` // Reads back the current FSK bit rate setting.

4.3.9 radio get fdev

4.3.9.1 Response:

signed decimal representing the frequency deviation setting, from 0 to 200000.

4.3.9.2 Description:

This command reads frequency deviation setting on the transceiver.
Default: 25000

4.3.9.3 Example:

`radio get fdev` // Reads back current configured frequency deviation setting.

4.3.10 radio get prlen

4.3.10.1 Response:

signed decimal representing the preamble length, from 0 to 65535.

4.3.10.2 Description:

This command reads the current preamble length used for communication.
Default: 8

4.3.10.3 Example:

`radio get prlen` // Reads back the preamble length used by the transceiver.

4.3.11 radio get crc

4.3.11.1 Response:

string representing the status of the CRC header, either `on` or `off`

4.3.11.2 Description:

This command reads back the status of the CRC header, to determine if it is to be included during operation.
Default: `on`

4.3.11.3 Example:

`radio get crc` // Reads back if the CRC header is enabled for use.

4.3.12 radio get iqi

4.3.12.1 Response:

string representing the status of the Invert IQ functionality, either `on` or `off`.

4.3.12.2 Description:

This command reads back the status of the Invert IQ functionality.
Default: `off`

4.3.12.3 Example:

`radio get iqi` // Reads back the status of the Invert IQ functionality.

4.3.13 radio get cr

4.3.13.1 Response:

string representing the current value settings used for the coding rate.
Parameter values can be: `4/5`, `4/6`, `4/7`, `4/8`.

4.3.13.2 Description:

This command reads back the current value settings used for the coding rate during communication.
Default: `4/5`

4.3.13.3 Example:

`radio get cr` // Reads back the current coding rate transceiver settings.

4.3.14 radio get wdt

4.3.14.1 Response:

decimal number representing the length used for the watchdog time-out, from 0 to 4294967295.

4.3.14.2 Description:

This command reads back, in milliseconds, the length used for the watchdog time-out.
Default: 15000

4.3.14.3 Example:

`radio get wdt` // Reads back the current time-out value applied to the Watchdog Timer

4.3.15 radio get bw

4.3.15.1 Response:

decimal representing the current operating radio bandwidth, in kHz.
Parameter values can be: 125, 250 or 500.

4.3.15.2 Description:

This command reads back the current operating radio bandwidth used by the transceiver.
Default: 125

4.3.15.3 Example:

`radio get bw` // Reads back the current operational bandwidth applied to transmissions.

4.3.16 radio get snr

4.3.16.1 Response:

signed decimal number representing the signal to noise ratio (SNR), from -128 to 127.

4.3.16.2 Description:

This command reads back the Signal Noise Radio (SNR) for the last received packet.
Default: -128

4.3.16.3 Example:

`radio get snr` // Reads back the measured SNR for the previously packet reception.

4.3.17 radio get sync

4.3.17.1 Response:

hexadecimal number representing the synchronization word used for radio communication.

4.3.17.2 Description:

This command reads back the configured synchronization word used for radio communication. One byte long synchronization word is used for the LoRa modulation while up to eight bytes can be entered for FSK.
Default: 34

4.3.17.3 Example:

```
radio get sync
```

4.3.18 radio get pa

4.3.18.1 Response:

string representing the status of the PABOOST, either `on` or `off`

4.3.18.2 Description:

This command reads back the status of the PABOOST, to determine if it is to be included during operation.
Default: `off`

4.3.18.3 Example:

`radio get pa` // Reads back if the PABOOST is enabled for use.

4.3.19 radio get reg <regAddr>

4.3.19.1 Response:

Hexadecimal value representing the address of radio register

4.3.19.2 Description:

This command returns the data from the particular radio register.

4.3.19.3 Example:

`radio get reg 10` // Reads back content of the radio register which address is 10

4.3.20 radio get regdump <regAddrStart> <regAddrEnd>

4.3.20.1 <regAddrStart>:

Hexadecimal value representing the starting address of radio register to be read

4.3.20.2 < regAddrEnd >:

Hexadecimal value representing the End address of radio register to be read

4.3.20.3 Response:

Sequence of Hexadecimal values read from the radio registers from start value to end

4.3.20.4 Description:

This command returns the set of register value from starting address mentioned in regAddrStart to ending address mentioned in regAddrEnd

4.3.20.5 Example1:

```
radio get regdump 00 05 // Returns the register content
from address 0x00 to 0x05
```

Register Address: 0x0 Value: 0x0
Register Address: 0x1 Value: 0x88
Register Address: 0x2 Value: 0x1a
Register Address: 0x3 Value: 0xb
Register Address: 0x4 Value: 0x0
Register Address: 0x5 Value: 0x52

Note: Register address should be given as Hex value

4.3.21 radio get pktrssi

4.3.21.1 Response:

signed decimal number representing the rssi value of last received packet using lora modulation.

4.3.21.2 Description:

This command reads back the RSSI value of last received packet while in loRa modulation
Default: `0 – If transceiver is in FSK modulation, this value will be 0`

4.3.21.3 Example:

```
radio get pktrssi  // Reads back rssi value of last received packet
```

5 References

-LORAWAN 1.0.2 specification

-LORAWAN 1.1 specification

-Requirements from PLC RF validation team.

6 Glossary