

# Wi-SUN - SoC CLI

The Wi-SUN CLI (Command-Line Interface) sample application allows developers to easily evaluate the Wi-SUN stack APIs. The Wi-SUN command-line interface provides a serial interface to a number of the Wi-SUN stack functions. For example, it can be used to connect the Wi-SUN device to a Wi-SUN border router and exchange IP packets or perform RF tests.

## Getting Started

To get started with Wi-SUN and Simplicity Studio, see [QSG181: Wi-SUN SDK Quick Start Guide](#).

NB: The Wi-SUN CLI application, as opposed to most other Wi-SUN sample applications, is not in 'auto-connect' mode by default. This means that the user has to set the network parameters and trigger the connection manually following a reset. This is visible when calling `wisun get wisun.join_state` in the console (the result will be `wisun.join_state = Disconnected (0)`). To change this behavior, use `wisun set app.autoconnect 1` followed by `wisun save` (when 'auto-connect' is active, a `Connecting...` message will be visible following a reset, with `Join_state: Select PAN...`).

This example exposes a command-line interface to interact with the Wi-SUN stack. To get started with the example, follow the steps below:

- Flash the "Wi-SUN Border Router" demonstration to a device and start the Border Router.
- Create and build the Wi-SUN CLI project.
- Flash the Wi-SUN CLI project to a second device.
- Using Simplicity Studio, open a console on the device running the Wi-SUN CLI project.

See the associated sections in [QSG181: Wi-SUN SDK Quick Start Guide](#) for step-by-step guidelines for each operation. To fully evaluate the Wi-SUN CLI features, another device running the Wi-SUN CLI application might be required. The Wi-SUN CLI application can also interact with the other Wi-SUN examples (Wi-SUN Ping, Wi-SUN TCP/UDP Server/Client...).

The Wi-SUN CLI example can be used to evaluate and test the Wi-SUN stack but should not be used to create production applications (due to its large footprint related to all PHYs being allowed). Developers should implement their own C application running in the EFR32 and using the Wi-SUN stack API, starting from simpler example applications.

## Wi-SUN CLI Commands

To see the available commands, enter the following command in the console.

```
wisun help
```

The list of available commands is output on the console with the associated help. Following is an extended description and examples of how to use each command.

Command	Description	Example
wisun get <section>.[setting]	Get a setting variable. Specifying only the section retrieves all settings of the section.	> wisun get wisun.ip_addresses > wisun get wisun > wisun get wisun.join_state > wisun get statistics > wisun get app
wisun set <section>.[setting] <value>	Set a setting variable	> wisun set wisun.network_size test
wisun join_fan10	Connect to the selected Wi-SUN FAN1.0 network	> wisun join_fan10
wisun join_fan11	Connect to the selected Wi-SUN FAN1.1 network	> wisun join_fan11
wisun join_explicit	Connect to a Wi-SUN (FSK only) network using explicit channel settings	> wisun join_explicit
wisun join_ids	Connect to a Wi-SUN network using radio conf ids	> wisun join_ids
wisun join_custom_fsk	Connect to a Wi-SUN network using custom FSK PHY settings	> wisun join_custom_fsk
wisun join_custom_ofdm	Connect to a Wi-SUN network using custom OFDM PHY settings	> wisun join_custom_ofdm
wisun join_custom_oqpsk	Connect to a Wi-SUN network using custom OQPSK PHY settings	> wisun join_custom_oqpsk
wisun disconnect	Disconnect from the Wi-SUN network	> wisun disconnect
wisun mac_allow <mac address>	Add an allowed MAC address to the access list broadcast address: allow all MAC addresses unique address: allow the given MAC address	> wisun mac_allow ff:ff:ff:ff:ff:ff > wisun mac_allow ffffffff:ffffff
wisun mac_deny <mac address>	Add a denied MAC address to the access list broadcast address: deny all MAC addresses unique address: deny the given MAC address	> wisun mac_deny 00:01:02:03:04:05:06:07 > wisun mac_deny 0001020304050607
wisun ping <remote address>	Ping a remote host	> wisun ping fd00:7283:7e00:0:fd6f:d00:a8c0:20fe
wisun reset	Reset variables to default values	> wisun reset
wisun save	Save variables to NVM	> wisun save
wisun socket_close <socket>	Close an open socket	> wisun socket_close 3

Command	Description	Example
wisun socket_list	List open sockets	> wisun socket_list
wisun socket_read <socket> <amount of bytes>	Read buffered data from a socket	> wisun socket_read 3 14
wisun socket_set_option <socket> <option> <option data>	Configure a socket	> wisun socket_set_option 3 event_mode indication
wisun socket_write <socket> <string>	Write a string to a socket	> wisun socket_write 3 "hello world"
wisun socket_writeto <socket> <remote address> <remote port> <string>	Write a string to an unconnected socket, i.e. UDP server socket	> wisun socket_writeto 3 fc00::1 5001 "hello world"
wisun tcp_client <remote address> <remote port>	Open a TCP connection to a remote host	> wisun tcp_client fd24:120b:802c:0001:705d:9179:8607:fd21 5001
wisun tcp_server <local port>	Open a TCP server port	> wisun tcp_server 5001
wisun udp_client <remote address> <remote port>	Open a UDP connection to a remote host	> wisun udp_client fd00:6172:6d00:0:3038:5115:26:27 7
wisun udp_server <local port>	Open a UDP server port	> wisun udp_server 5001
wisun socket_get_option <socket> <option>	Get a socket option	> wisun socket_get_option
wisun clear_credential_cache	Clear the wisun credential cache	> wisun clear_credential_cache
wisun reset_statistics	Reset the statistics	> wisun reset_statistics
wisun set_trace_level <string>	Set <a href="#">trace level</a> for a <a href="#">trace group</a>	> wisun set_trace_level all,debug [Set 64 trace groups] > wisun set_trace_level 1,none;4,info [Set 2 trace groups]
wisun set_lfn_support <lfn_limit>	configure LFN support: - [uint8] lfn_limit: Number of LFN children (10 maximum)	> wisun set_lfn_support 3
wisun mode_switch <mode> <phy_mode_id> <neighbor_mac_address>	configure mode switch: - [uint8] mode: 0=disabled, 1=enabled, 2=default - [uint8] PhyModeID - [string] neighbor MAC address (optional)	> wisun mode_switch 1 84 > wisun mode_switch 1 84 01:02:03:04:05:06:07:08

Command	Description	Example
wisun trigger <frame type>	Trigger a transmission of a periodic frame - [string] pas: PAN Advertisement Solicit, pa: PAN Advertisement, pcs: PAN Configuration Solicit, pc: PAN Configuration, dis: DODAG Information Solicitation, dio: DODAG Information Object	> wisun trigger pas
wisun getsockname <socket>	Get local IP address of the socket	> wisun getsockname 3
wisun getpeername <socket>	Get peer IP address of the socket	> wisun getpeername 3
wisun set_phy_sens <phy_mode_id> <sensitivity>	Set the sensitivity for the given PhyModelID - [uint8] PhyModelID - [int16] sensitivity in dBm	> wisun set_phy_sens 1 -70
wisun set_direct_connect_state <state>	Set Direct Connect state - [uint8] state: 0=disabled, 1=enabled	wisun set_direct_connect_state 1
wisun accept_direct_connect_link <remote address>	Accept a Direct Connect link request	wisun accept_direct_connect_link fe80::b6e3:f9ff:fec5:83df

## Wi-SUN Settings

The command-line interface maintains a number of settings. The Wi-SUN settings are distributed in three sections: *wisun*, *statistics*, and *app*. They can be listed by entering:

```
wisun get <section name>
```

The Wi-SUN stack settings are listed with their current state/value. Some of them can be modified by using the following command prototype:

```
wisun set <section name>.<settings name> <value>
```

To modify the network name the Wi-SUN device should connect to, enter:

```
wisun set wisun.network_name "My Network"
```

Next time you issue the `wisun join_xxxxx` command, the device starts a connection process with the Wi-SUN network named "My Network".

## *wisun Section Settings*

- The settings in the *wisun* section are directly related to the Wi-SUN stack behavior. A detailed settings list can be found below.
- Writable settings are valid for all PHYs unless it is specified otherwise in the last column .

Variable	R/W	Type	Values	Description	Applies to
wisun.network_name	R/W	string	up to 31 ASCII characters. Use double quotes (") to include spaces	Name of the Wi-SUN network	
wisun.phy_config_type	R	string	- FAN1.0 (0) - FAN1.1 (1) - explicit (2) - IDs (3)	Wi-SUN PHY configuration type (depends on the wisun_join_xxxx used)	
wisun.regulatory_domain	R/W	integer	- WW (0): Worldwide - NA (1): North America - JP (2): Japan - EU (3): Europe - CN (4): China - IN (5): India - MX (6): Mexico - BZ (7): Brazil - AZ (8): Australia/New Zealand - KR (9): South Korea - PH (10): Philippines - MY (11): Malaysia - HK (12): Hong Kong - SG (13): Singapore - TH (14): Thailand - VN (15): Vietnam	Regulatory domain of the Wi-SUN network	- FAN1.0 - FAN1.1
wisun.operating_class	R/W	integer	1 to 4	Wi-SUN FAN1.0 operating class	FAN1.0
wisun.operating_mode	R/W	integer	- 0x1a (1a) - 0x1b (1b) - 0x2a (2a) - 0x2b (2b) - 0x3 (3) - 0x4a (4a) - 0x4b (4b) - 0x5 (5)	Wi-SUN FAN1.0 operating mode	FAN1.0
wisun.network_size	R/W	integer	- automatic (0) - small (default): less than 100 nodes (1) - medium: 100 to 1000 nodes (2) - large: more than 1000 nodes (3) - test (4) - certification (6)	Used to optimize network timings considering the number of expected nodes on the network.	
wisun.tx_power	R/W	integer	-45 to 20	Maximum TX power in dBm	
wisun.tx_power_ddbm	R/W	integer	-450 to 200	Maximum TX power in deci-dBm	
wisun.unicast_dwell_interval	R/W	integer	10 to 255	Unicast dwell interval in milliseconds	
wisun.chan_plan_id	R/W	integer	- Europe: 32-37 - North America: 1-5 - Brazil: 1-5 - Japan: 21-24	Wi-SUN channel plan ID [uint8]	FAN1.1

Variable	R/W	Type	Values	Description	Applies to
wisun.phy_mode_id	R/W	integer	- Europe: FSK without FEC: 1/3/5 FSK FEC: 17/19/21 OFDM: 84-86 - North America: FSK without FEC: 2/3/5/6/8 FSK FEC: 18/19/21/22/24 OFDM: 34-38 / 51-54 / 68-70 / 84-86 - Brazil: FSK without FEC: 2/3/5/6/8 FSK FEC: 18/19/21/22/24 34-38 / 51-54 / 68-70 / 84-86 - Japan: FSK without FEC: 2/4/5/7/8 FSK FEC: 18/20/21/23/24 OFDM: 51-54 / 68-70 / 84-86	Wi-SUN FAN1.1 PHY mode ID	- FAN1.1 - explicit - join by IDs
wisun.ch0_frequency	R/W	integer		Explicit channel plan: Central frequency of the first channel in kHz	explicit
wisun.channel_spacing	R/W	integer	- 100 kHz (0) - 200 kHz (1) - 400 kHz (2) - 600 kHz (3) - 250 KHz (4) - 800kHz (5) - 1200KHz (6)	Explicit channel plan: Channel spacing	explicit
wisun.number_of_channels	R/W	integer		Explicit channel plan: Number of channels	explicit
wisun.protocol_id	R/W	integer		Radioconf protocol ID [uint16]	join by IDs
wisun.channel_id	R/W	integer		Radioconf channel ID [uint16]	join by IDs
wisun.ip_addresses	R	list of IPv6 addresses		List of all IP addresses assigned to the device	
wisun.border_router	R	list of IPv6 addresses		List of known IPv6 addresses of the border router	
wisun.parents	R	list of IPv6 addresses		List of known IPv6 addresses of the parents	
wisun.neighbors	R	list of IPv6 addresses		List of known IPv6 addresses of the RPL neighbors	
wisun.join_state	R	integer	- Idle (0) - Select PAN (1) - Authenticate (2) - Acquire PAN Config (3) - Configure Routing (4) - Operational (5)	Current join state in the connection process	
wisun.mac_address	R/W	MAC address		MAC address to use	
wisun.allowed_channels	R/W	string		Allowed channel ranges (e.g. 0-54,57-60,64,67-68)	
wisun.fec	R/W	bool	- disable FEC (0) - enable FEC (1)	Use FEC flag	FAN1.0
wisun.regulation	R/W	string	- None (0) - ARIB (1)	Regional regulation	
wisun.regulation_warning_threshold	R/W	integer	- threshold value (0-100) - disable (-1)	Transmission power warning threshold in percent (-1 to disable)	
wisun.regulation_alert_threshold	R/W	integer	- threshold value (0-100) - disable (-1)	Transmission power alert threshold in percent (-1 to disable)	
wisun.device_type	R/W	string	- FFN: Full Function Node (0) - LFN: Limited Function Node (1)	Device type	

Variable	R/W	Type	Values	Description	Applies to
wisun.trace_filter.000-031	R/W	hexadecimal	(Refer to sl_wisun_types.h / sl_wisun_trace_group_t for a list of all 37 trace groups)	RTT Trace filter enable bitfield for first 32 groups	
wisun.trace_filter.032-063	R/W	hexadecimal	(Refer to sl_wisun_types.h / sl_wisun_trace_group_t for a list of all 37 trace groups)	RTT Trace filter enable bitfield for last 32 groups	
wisun.lfn_profile	R/W	integer	- Test (0) - Balanced (1) - Eco (2)	Wi-SUN LFN profile	FAN1.1
wisun.rx_phy_mode_ids	R/W	integer	list of phy_mode_ids	List of PhyModeId to advertise in POM-IE	FAN1.1
wisun.crc_type	R/W	integer	- no CRC (0) - 2-byte CRC (1) - 4-byte CRC (2)	CRC for custom FSK/OFDM/OQPSK PHY	
wisun.preamble_length	R/W	integer		Preamble length in bits for custom FSK/OQPSK PHY	
wisun.stf_length	R/W	integer		Short Training Field length in symbols for custom OFDM PHY	
wisun.async_channel_mask	R	xx:....xx	- Not masked (0) - Masked (1)	Excluded channel mask applied to async frames	
wisun.unicast_channel_mask	R	xx:....xx	- Not masked (0) - Masked (1)	Excluded channel mask applied to unicast frequency hopping	
wisun.broadcast_channel_mask	R	xx:....xx	- Not masked (0) - Masked (1)	Excluded channel mask applied to broadcast frequency hopping	
wisun.neighbor_table_size	R/W	integer	1 to 254	Size of the neighbor table, excluding temporary entries and RPL parents	
wisun.preferred_pan_id	R/W	integer	0 to 65535	Preferred PAN ID (0xFFFF to disable)	
wisun.keychain	R/W	integer	- automatic (0) - built-in (1) - NVM (2)	Keychain to use for device credentials	
wisun.keychain_index	R/W	integer	0 to 255	Device credential index to use for built-in keychain	
wisun.direct_connect_pmk <pmk>	R/W	string	64 hex characters	Preshared Master Key for Direct Connect	

## Non-Standard ('explicit') Channel Plan

Customers willing to experiment with non-standard frequency plans need to set:

- wisun.regulatory\_domain 255
- wisun.operating\_class 255
- wisun.ch0\_frequency
- wisun.channel\_spacing
- wisun.number\_of\_channels
- wisun.channel\_mask

and join the network using wisun join\_explicit

## *statistics Section Settings*

The settings part of the *statistics* section are counters maintained by the Wi-SUN stack. A detailed setting list is available below.

Variable	Read/Write	Description
statistics.phy	R	PHY statistics
statistics.mac	R	MAC statistics
statistics.fhss	R	Frequency hopping statistics
statistics.wisun	R	Wi-SUN layer statistics
statistics.network	R	6LoWPAN/IP stack statistics
statistics.regulation	R	Regional regulation statistics and errors
statistics.heap	R	Memory consumption statistics

## *info Section Settings*

The settings part of the *info* section contain information about the connected Wi-SUN network. A detailed setting list is available below.

Variable	Read/Write	Description
info.network	R	Wi-SUN network information
info.rpl	R	RPL information

## *app Section Settings*

The settings in the *app* section relate to the application options. A detailed setting list is available below.

Variable	R/W	Type	Values	Description
app.printable_data_length	R/W	integer	0: received socket data is not printed 1 - 64: amount of characters per line	If enabled, received socket data is printed
app.printable_data_as_hex	R/W	integer	0: print received socket data as ASCII 1: print received socket data as hex	Output type for received socket data
app.autoconnect	R/W	bool	0: disable autoconnect 1: enable autoconnect	Enable or disable autoconnect [bool]



Variable	R/W	Type	Values	Description
app.pti_state	R/W	bool	0: disable PTI 1: enable PTI	Disable or enable PTI

## ping Section Settings

The settings in the *ping* section relate to the ping command. A detailed setting list is available below.

Variable	R/W	Type	Values	Description
ping.identifier	R/W	integer	0 - 65535	Identifier to aid in matching ICMPv6 Echo Requests and Replies
ping.sequence_number	R/W	integer	0 - 65535	Sequence number to aid in matching ICMPv6 Echo Requests and Replies
ping.packet_length	R/W	integer	8 - 2048	Length of ICMPv6 Echo Request packet in bytes, including the header
ping.pattern	R/W	string		Repeating pattern used to fill ICMPv6 Echo Request payload

## RF test Commands

To see the available commands, enter the following command in the console.

```
rfctest help
```

The list of available commands is output on the console with the associated help. The following is an extended description and examples of how to use each command.

Command	Description	Example
rfctest start_stream <channel ID>	Start an RF test stream on a specific channel	> rfctest start_stream 5 RF Test stream started 21
rfctest stop_stream	Stop the RF test	> rfctest stop_stream RF Test stream stopped 2
rfctest start_tone <channel ID>	Start an RF test tone on a specific channel	> rfctest start_tone 5 RF Test tone started 21
rfctest stop_tone	Stop RF test tone	> rfctest stop_tone RF Test tone stopped 2
rfctest set_tx_power <value>	Set Tx power for the RF test in dBm	> rfctest set_tx_power 17 RF Test tx Power set to 17

## RTT traces

To control RTT traces, set the trace level using `wisun set_trace_level [group_id],[trace level];`

All RTT trace groups and trace levels are listed in `sl_wisun_types.h/sl_wisun_trace_group_t`

```
typedef enum {
SL_WISUN_TRACE_GROUP_MAC      = 0,      ///< Mac
SL_WISUN_TRACE_GROUP_NW      = 1,      ///< Network
SL_WISUN_TRACE_GROUP_LLC     = 2,      ///< LLC
SL_WISUN_TRACE_GROUP_6LO     = 3,      ///< 6lowpan
SL_WISUN_TRACE_GROUP_IPV6    = 4,      ///< IPV6
. . .
SL_WISUN_TRACE_GROUP_WSIE    = 35,     ///< Wi-SUN IE
SL_WISUN_TRACE_GROUP_CONFIG  = 36,     ///< Configuration
// 36 to 63 reserved for future used
SL_WISUN_TRACE_GROUP_INT     = 63,     ///< Internal usage
SL_WISUN_TRACE_GROUP_COUNT   = 64     ///< Max number of trace group in this enum
} sl_wisun_trace_group_t;

. . .
typedef enum {
///< No trace
SL_WISUN_TRACE_LEVEL_NONE    = 0,
///< Error only
SL_WISUN_TRACE_LEVEL_ERROR   = 1,
///< Warning + error
SL_WISUN_TRACE_LEVEL_WARN    = 2,
///< Info + warning + error
SL_WISUN_TRACE_LEVEL_INFO    = 3,
///< Debug + info + warning + error
SL_WISUN_TRACE_LEVEL_DEBUG   = 4,
} sl_wisun_trace_level_t;
```

Example: to set 'Network' to 'Debug' and 'IPV6' to 'Warning', use:

```
wisun set_trace_level 1,0;4,4; or wisun set_trace_level 1,none;4,debug;
```

## Troubleshooting

Before programming the radio board mounted on the WSTK, ensure the power supply switch is in the AEM position (right side), as shown.



## Resources

---

- [Wi-SUN Stack API documentation](#)
- [AN1330: Wi-SUN Mesh Network Performance](#)
- [AN1332: Wi-SUN Network Setup and Configuration](#)
- [AN1364: Wi-SUN Network Performance Measurement Application](#)
- [QSG181: Wi-SUN Quick-Start Guide](#)
- [UG495: Wi-SUN Developer's Guide](#)

## Report Bugs & Get Support

---

You are always encouraged and welcome to ask any questions or report any issues you found to us via [Silicon Labs Community](#).