KONGSBERG

Python library for the Radio configuration and status API

The Radio configuration and status API allow users to set configuration parameters and check system status.

The Python library compatible with Python 2.7 and is available in 64-bit version for Linux and 32- and 64-bit versions for Windows.

To use the library you must have “NumPy” installed, see www.numpy.org

- Windows 64-bit: radio_api_lib_win64.zip
- Windows 32-bit: radio_api_lib_win32.zip
- Linux 64-bit: radio_api_lib.tar.gz

Example code that scans the local network for radios and prints wireless status:

```
import radio_api_lib
for sn in radio_api_lib.discover():
        m = radio_api_lib.radio(sn)

        print m.get_wireless_status()
```

radio_api_lib.radio = class radio

```
Class representing a radio.
```

Methods defined here:

__init__(self, sn)

```
Initializes an instance of the radio class.
Calculates and sets the MAC and IP address based on the supplied s
erial number.
Args:
    sn (int): serial number (also called ID) of unit.
Raises:
    RadioException
```

__repr__(self)

add_host(self, host_ip, host_mac, desc=None, type_=None, site_id=None, dest_ip=None, dest_id=None)

```
Add a host to this radios host list. Creates a mac-
ip route on this unit such that the host can talk to other hosts w
irelessly over the network.
Devices have to be added to one of the radios on their local netwo
rk.
WARNING: If the host is added, then connected to the local network
 of another site on the same wireless network, it will not be able
 to communicate with the unit on this new site.
Args:
    host_mac (string): the mac address of the host as a ":" or "-
" separated string (i.e "aa:bb:cc:dd:aa:bb").
    host_ip (string): the ip address of the host as a "." separate
d string (i.e "1.2.3.4").
    desc (str, optional): description of the host. Shown in the we
b-gui.
    type_ (str, optional): type description of the host. Shown in
the web-gui.
    site_id (int, optional): if this is set the host will be added
 to the host list of the site with this id.
    dest_ip (int, optional): if this is set the host will be added
 to the site list of the radio with this ip.
    dest_id (int, optional): if this is set the host will be added
 to the site list of the radio with this id.
Raises:
    RadioException
```

add_site(self, name, radios)

```
Add a new site to the site list of the radio.
Args:
    name (str): the name of the site to be added (i.e "site_1").
    radios (list or int): a list of ids (serial number) of radios
on the site to be added, or one id as int
Raises:
    RadioException
```

get_frequency(self)

```
Get frequency as float
```

get_local_site_setup(self)

```
Get the local site setup (site name, associated radios and hosts)
as a dictionary.
```

get_network_id(self)

```
Get the network id as hex formated long value (i.e 0xaabbccddeeL).
```

get_network_superframe_time(self)

```
Get superframe time as float
```

get_network_time_allocation(self)

```
Get current time allocation as a dictionary.
```

get_network_time_allocation_advanced(self)

```
Get current advanced time allocation as a dictionary.
```

get_site_table(self)

```
Get the site table (site names and associated radio serial numbers
) as a dictionary.
```

get_system_status(self)

```
Get the system status information as a dictionary.
```

get_temperature(self)

```
Get the internal temperature as float
```

get_wireless_status(self)

```
Get wireless link status information as a dictionary.
```

init(self, site_name=None, radio_list=None)

```
Initializes the site this radio belongs to with name site_name and
 the radios in radio_list.
Args:
    site_name (str): name of this site.
    radio_list (list of int): the radios on this site.
```

remove_host(self, host_ip, host_mac)

```
Add a host to this radios host list. Creates a mac-
ip route on this unit such that the host can talk to other hosts w
irelessly over the network.
Devices have to be added to one of the radios on their local netwo
rk.
WARNING: If the host is added this unit, then connected to the loc
al network of another site on the same network, it will not be abl
e to communicate with the units on this new site.
Args:
    host_mac (string): the mac address of the host as a ":" or "-
" separated string (i.e "aa:bb:cc:dd:aa:bb").
    host_ip (string): the ip address of the host as a "." separate
d string (i.e "1.2.3.4").
Raises:
    RadioException
```

remove_site(self, radio)

```
Remove a site from the site list of the radio.
Args:
    name (str): the name of the site to be added (i.e "site_1").
    radios (list/int): a list of ids (serial numbers as int) of ra
dios on the site to be added, or one id as int
Raises:
    RadioException
```

set_datarate(self, mbps)

```
Set the transmit datarate.
Args:
    datarate (int): datarate in Mbps. Valid datarates are 0.5, 1,
2, 7 and 15 Mbps.
Raises:
    RadioException
```

set_frequency(self, mhz)

```
Set the radio frequency.
Args:
    mhz (float): frequency in MHz. If the supplied value has decim
als the unit will only use the first two decimals.
Raises:
    RadioException
```

set_network_id(self, network_id)

```
        Set the network id.
        Args:
            network_id (int): the network id to be set.
        Raises:
            RadioException
```

set_network_name(self, network_name)

```
        Set the network name.
        Args:
            network_name (str or int): the network name to be set.
        Raises:
            RadioException
```

set_network_time_allocation(self, network_id, superframe_length, allocation)

```
        Set the network time allocation in a network.
        Args:
            network_id (int): the network id of the network.
            superframe_length (float): superframe lenght in milliseconds.
            allocation (dict): dict with serial numbers as int as key and
        allocation percentage as float as value.
                                All radios on the same site share the same
        resource allocation, so only one serial number for each site is ne
        cessary.
                                If the values don't add up to 100%, the res
        t is set as contention period.
                                ex: allocation = {90: 12.9, 91: 81.1}
                                    This gives 12.9% to site containing the
         unit with sn 90 and 81.1% to the site containing unit with sn 91.

        Raises:
            RadioException
```

set_network_time_allocation_advanced(self, superframe_length, time_table)

```
        Set the network time allocation in a network.
        Args:
            superframe_length (float): superframe length in milliseconds.
            time_table (dict): dict with serial numbers as int as key and
        desired start of the slot in milliseconds as float as value.
                                All radios on the same site share the same
        resource allocation, so any serial number used on the site will wo
        rk.
                                Special IDs that can be defined in the key
        are 0 for contention, and 65535 for radio silence.
                                ex: time_table = {90: 0.0, 91: 80.5, 0: 180
         }
                                    This gives a slot 80.5 ms slot to site
        containing the unit with sn 90, 99.5 ms to the site containing uni
        t with sn 91, and
                                    20 ms for contention.
        Raises:
            RadioException
```

set_power_control(self, max_power, target_margin, max_reduction)

```
        Sets the power control settings of the radio
```

radio_api_lib.radio_api_functions

index /home/tob/git/snapgear-2.6-p40-CRE2/python/radio_api_lib/radio_api_functions.py

```
Wrapper for datagram library
```

Modules

binascii	os	re
json	radio_api_lib.radio_api_lib	socket

Functions

add_host(dest_ip, dest_id, source_id, site_id, macStr, device_ip, desc, type_)

```
Adds a new device to the device list of a radio.
    This creates a mac route to a device connected to one of th
e radios on the network.
    This mac route will be synched between all the radios on th
e network, and the device
    with mac "device_mac" will be available to all other device
s in the network on
    the ip "device_ip".
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    site_id (int): id of one of the radios on the site the devi
ce is connected to.
    device_mac (str): mac of the device
    device_ip (str): ip of the device
    desc (str):       description of the device
    type_ (str):      the type of the device
Raises:
    RadioException:
```

add_site(dest_ip, dest_id, source_id, name, radio_list)

```
Adds a new site to the site list of a radio.
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    name (str): the name of the site to be added (i.e "Site_1")
.
    radio_list (list): a list of ids (serial numbers as int) of
 radios on the site to be added.
Raises:
    RadioException:
```

get_local_site_setup(dest_ip, dest_id, source_id)

get_network_time_allocation(dest_ip, dest_id, source_id)

get_network_time_allocation_advanced(dest_ip, dest_id, source_id)

get_site_table(dest_ip, dest_id, source_id)

get_system_status(dest_ip, dest_id, source_id)

get_wireless_status(dest_ip, dest_id, source_id)

remove_host(dest_ip, dest_id, source_id, macStr, device_ip)

remove_site(dest_ip, dest_id, source_id, site_id)

```
Adds a new site to the site list of a radio.
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    name (str): the name of the site to be added (i.e "Site_1")
.
    radio_list (list): a list of ids (serial numbers as int) of
 radios on the site to be added.
Raises:
    RadioException:
```

set_datarate(dest_ip, dest_id, source_id, datarate)

```
Sets the datarate of a Radio
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    datarate (int): datarate in Mbps. Valid datarates are 0.5,
1,2,7 and 15 Mbps.
Raises:
    RadioException:
```

set_frequency(dest_ip, dest_id, source_id, mhz, khz)

```
Sets the radio frequency of a Radio
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    mhz (int): mhz part of the frequency
    khz (int): khz part of the frequency
Raises:
    RadioException:
```

set_network_id(dest_ip, dest_id, source_id, network_id)

```
Sets the network id of a Radio
Args:
    dest_ip (str): ip of the radio you wish to send the configu
ration to.
    dest_id (int): id (serial number) of the radio you wish to
send the configuration to.
    source_id (int):    id of one of the radios on the local si
te
                        (i.e one of the radios you share a wire
d network with).
    network_id (int): the network id to be set
```

```
        Raises:
            RadioException:
```

set_network_name(dest_ip, dest_id, source_id, network_name)

```
        Sets the network name of a Radio
        Args:
            dest_ip (str): ip of the radio you wish to send the configu
        ration to.
            dest_id (int): id (serial number) of the radio you wish to
        send the configuration to.
            source_id (int):    id of one of the radios on the local si
        te
                                (i.e one of the radios you share a wire
        d network with).
            network_name (str): the network name to be set
        Raises:
            RadioException:
```

set_network_time_allocation(dest_ip, dest_id, source_id, network_id, superframe_length, allocation)

```
        Sets the time allocation in a Radio network.
        Args:
            dest_ip (str): ip of the radio you wish to send the configu
        ration to.
            dest_id (int): id (serial number) of the radio you wish to
        send the configuration to.
            source_id (int):    id of one of the radios on the local si
        te
                                (i.e one of the radios you share a wire
        d network with).
            network_id (int): the network id of the network
            superframe_length (int): superframe lenght in miliseconds
            allocation (dict): dict with Radio ids as keys and allocati
        on percentage  as values.
                               ex: allocation = {90: 12, 91: 88}
                                   This gives 12% to Radio with id 90 a
        nd 88 to Radio with id 91.
        Raises:
            RadioException:
```

set_network_time_allocation_advanced(dest_ip, dest_id, source_id, network_id, superframe_length, time_table)

```
        Sets the time allocation in a Radio network.
        Args:
            dest_ip (str): ip of the radio you wish to send the configu
        ration to.
            dest_id (int): id (serial number) of the radio you wish to
        send the configuration to.
            source_id (int):    id of one of the radios on the local si
        te
                                (i.e one of the radios you share a wire
        d network with).
            network_id (int): the network id of the network
            superframe_length (int): superframe lenght in miliseconds
            allocation (dict): dict with Radio ids as keys and allocati
        on percentage  as values.
                               ex: allocation = {90: 12, 91: 88}
                                   This gives 12% to Radio with id 90 a
        nd 88 to Radio with id 91.
        Raises:
            RadioException:
```

set_power_control(dest_ip, dest_id, source_id, max_power, target_margin, max_reduction)

radio_api_lib.utilities

index
/home/tob/git/snapgear-2.6-p40-CRE2/python/radio_api_lib/utilities.py

Modules

binascii
os
radio_api_lib.radio_api_lib
re
socket

Classes

exceptions.Exception(exceptions.BaseException)
    RadioException

class RadioException(exceptions.Exception)

```
Radio Exception class
```

Method resolution order:
- RadioException
- exceptions.Exception
- exceptions.BaseException
- __builtin__.object

Methods defined here:

__init__(self, message)

Data descriptors defined here:

__weakref__

```
list of weak references to the object (if defined)
```

Data and other attributes inherited from exceptions.Exception:

__new__ = <built-in method __new__ of type object>

```
T.__new__(S, ...) -
> a new object with type S, a subtype of T
```

Methods inherited from exceptions.BaseException:

__delattr__(...)

```
x.__delattr__('name') <==> del x.name
```

__getattribute__(...)

```
x.__getattribute__('name') <==> x.name
```

__getitem__(...)

```
x.__getitem__(y) <==> x[y]
```

__getslice__(...)

```
x.__getslice__(i, j) <==> x[i:j]

Use of negative indices is not supported.
```

__reduce__(...)

__repr__(...)

```
x.__repr__() <==> repr(x)
```

__setattr__(...)

```
x.__setattr__('name', value) <==> x.name = value
```

__setstate__(...)
__str__(...)
x.__str__() <==> str(x)
__unicode__(...)

Data descriptors inherited from exceptions.BaseException:
__dict__
args
message

Functions

discover()
```
Discovers all radios on the local network
Returns:
    list:  Containing the serial number of all radios found.
Raises:
    socket.error
```
int_to_ip_string(ip)
```
Convert ip as in to ip as string
```
ip_string_to_int(ip_addr)
```
Converts an ip as a string to an int
```
ip_to_sn(ip)
```
Convert ip as string to serial number as int
Returns:
    int:  serial number corresponding to the supplied ip.
```
is_ip(ip_route)
```
Checks if a string is a valid ip
```
is_mac(mac)
```
Checks if a string is a valid mac
```
mac_string_to_int(mac)
```
Convert a mac address as a "-
" or ":" separated string to an int
```
ping(host)
```
Returns True if host (str) responds to a ping request. Works fo
r devices supporting ICMP.
```
send_buffer(dest_ip, dest_port, datagram, timeout=2)
```
Sends a string to ip dest_ip at port dest_port
Args:
    dest_ip (str): Destination ip as string
    dest_port (int): Destination port as int
    datagram (str): String containing the data to be sent
    timeout (int): Socket timeout parameter. Defaults to 2 seco
nds.
Raises:
    socket.error
```
types_ok(arg_list, type_list)
```
Checks if a list of arguments are of the correct types specifie
d by a list of types
```