

---

# **nessclient Documentation**

**Nick Whyte**

**Jun 09, 2023**



---

## Contents:

---

<b>1</b>	<b>Installing nessclient</b>	<b>3</b>
1.1	CLI . . . . .	3
1.2	API . . . . .	3
1.3	Examples . . . . .	4
1.4	Developing . . . . .	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>



A python implementation/abstraction of the [Ness D8x / D16x Serial Interface ASCII protocol](#)



# CHAPTER 1

---

## Installing nessclient

---

nessclient is available directly from pip:

```
pip install nessclient
```

### 1.1 CLI

This package includes a CLI which uses the library to interface with the Ness Serial Interface.

To use the CLI you must install it's dependencies by installing it with extras for `cli`:

```
pip install nessclient[cli]  
ness-cli --help
```

### 1.2 API

#### 1.2.1 nessclient Core

#### 1.2.2 nessclient.event

#### 1.2.3 nessclient.connection

.

## 1.3 Examples

### 1.3.1 Sending Commands

A list of commands that can be sent can be found in the *Input Commands* section of the [protocol documentation](#). Additional information on the user-level commands available can be found in the *Operation Summary* of the [installer manual](#).

```
import asyncio

from nessclient import Client

loop = asyncio.get_event_loop()
host = '127.0.0.1'
port = 65432
client = Client(host=host, port=port)

# Send arming command via library abstraction
loop.run_until_complete(client.arm_away('1234'))
# Send panic command via library abstraction
loop.run_until_complete(client.panic('1234'))
# Send disarm command via library abstraction
loop.run_until_complete(client.disarm('1234'))
# Send aux control command for output 2 via library abstraction
loop.run_until_complete(client.aux(2))
# Send custom command
# In this instance, we are sending a status update command to view
# output status
loop.run_until_complete(client.send_command('S15'))

client.close()
loop.close()
```

### 1.3.2 Listening for Events

```
import asyncio

from nessclient import Client, ArmingState, BaseEvent

loop = asyncio.get_event_loop()
host = '127.0.0.1'
port = 65432
client = Client(host=host, port=port)

@client.on_zone_change
def on_zone_change(zone: int, triggered: bool):
    print('Zone {} changed to {}'.format(zone, triggered))

@client.on_state_change
def on_state_change(state: ArmingState):
    print('Alarm state changed to {}'.format(state))
```

(continues on next page)



(continued from previous page)

```
@client.on_event_received
def on_event_received(event: BaseEvent):
    print('Event received:', event)

loop.run_until_complete(asyncio.gather(
    client.keepalive(),
    client.update(),
))

client.close()
loop.close()
```

## 1.4 Developing

Use `pipenv` to setup the local environment:

```
pipenv install --dev
```

### 1.4.1 Running tests

```
pipenv run python setup.py test
```

### 1.4.2 Linting

```
pipenv run flake8 nessclient nessclient_tests
```

### 1.4.3 Type Checking

```
pipenv run mypy --strict nessclient
```



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`