

---

# **hut2 Documentation**

***Release 0.1***

**Robert Jördens**

**Apr 22, 2019**



## CONTENTS

<b>1 hut2.protocol module</b>	<b>3</b>
<b>2 Indices and tables</b>	<b>7</b>
<b>Python Module Index</b>	<b>9</b>



Contents:



---

CHAPTER  
ONE

---

## HUT2 . PROTOCOL MODULE

```
class hut2.protocol.HUT2(user=b'admin', password=b'anel')  
    ANEL HUT2 Driver
```

Power relay switch with 8 channels, and 8 channels general purpose IO.

- Relay and IO indices start with 1.
- **Commands tend to keep the device busy for a while.** Pace them, read back the status after a while and retry them.
- **Some commands sometimes return a status update. Since they are UDP,** neither commands nor replies are guaranteed to arrive.

### Parameters

- **user** – Device user name (bytes)
- **password** – Device password (bytes)

```
classmethod connect(host, port=7500, local_addr=('255.255.255.255', 7700), loop=None,  
                   **kwargs)
```

Connect to a Anel HUT2 via UDP

### Parameters

- **host** – IO or host name of the device
- **port** – UDP port on the device. Default is 7500 (unprivileged) as opposed to 75 (privileged). Ensure this is set correctly in the web interface.
- **local\_addr** – Tuple of local address (or hostname) and UDP port. Default is (“255.255.255.255”, 7500) (broadcast, unprivileged). The device default is to reply to port 75 (privileged). Ensure this is set correctly in the web interface.

**Returns** HUT2 instance connected to the device

### connection\_lost(exc)

Called when the connection is lost or closed.

The argument is an exception object or None (the latter meaning a regular EOF is received or the connection was aborted or closed).

### connection\_made(transport)

Called when a connection is made.

The argument is the transport representing the pipe connection. To receive data, wait for data\_received() calls. When the connection is closed, connection\_lost() is called.

**datagram\_received**(*data, addr*)  
Called when some datagram is received.

**do**(*cmd*)  
Execute a command.

**Parameters** (**bytes**) (*cmd*) – Command

**get\_io\_states**()  
Return current IO states as a list.

**Returns** List of eight integers

**get\_port\_states**()  
Return current relay states as a list.

**Returns** List of eight integers

**get\_status**()  
Get a status update.

This is a coroutine that performs the status query and returns the result.

**Returns** *Status* with the parsed fields

**get\_temp**()  
Return current temperature.

**Returns** Temperature in C

**io**(*ios*)  
Set all IO lines.

**Parameters** (**int**) (*ios*) – Bitmask of the IO state

**io\_off**(*io*)  
Turn an IO off

**Parameters** (**int**) (*switch*) – IO line index

**io\_on**(*io*)  
Turn an IO on.

**Parameters** (**int**) (*io*) – IO line index

**query**()  
Query device status and settings.

**st\_off**(*switch, delay*)  
Turn a relay off after a delay

**Parameters**

- (**int**) (*delay*) – Relay index
- (**int**) – Delay in seconds

**sw**(*switches*)  
Set all relays.

**Parameters** **switches** – Bitmask of the relay settings

**sw\_off**(*switch*)  
Turn a relay off

**Parameters** (**int**) (*switch*) – Relay index

**sw\_on** (*switch*)

Turn a relay on

**Parameters** **(int)** (*switch*) – Relay index

**wait()**

Return a new status message.

This returns a future.

**Returns** *Status* with the parsed fields

**class** hut2.protocol.**Status**

HUT2 Status message parser and representation

**classmethod from\_bytes** (*data*, *time=None*)

Parse status message and return a namedtuple of the fields



---

**CHAPTER  
TWO**

---

**INDICES AND TABLES**

- genindex
- modindex
- search



## PYTHON MODULE INDEX

h

hut2.protocol, 3



# INDEX

## C

connect () (*hut2.protocol.HUT2 class method*), 3  
connection\_lost () (*hut2.protocol.HUT2 method*),  
    3  
connection\_made () (*hut2.protocol.HUT2 method*),  
    3

## D

datagram\_received ()         (*hut2.protocol.HUT2  
method*), 3  
do () (*hut2.protocol.HUT2 method*), 4

## F

from\_bytes () (*hut2.protocol.Status class method*), 5

## G

get\_io\_states () (*hut2.protocol.HUT2 method*), 4  
get\_port\_states () (*hut2.protocol.HUT2 method*),  
    4  
get\_status () (*hut2.protocol.HUT2 method*), 4  
get\_temp () (*hut2.protocol.HUT2 method*), 4

## H

HUT2 (*class in hut2.protocol*), 3  
hut2.protocol (*module*), 3

## I

io () (*hut2.protocol.HUT2 method*), 4  
io\_off () (*hut2.protocol.HUT2 method*), 4  
io\_on () (*hut2.protocol.HUT2 method*), 4

## Q

query () (*hut2.protocol.HUT2 method*), 4

## S

st\_off () (*hut2.protocol.HUT2 method*), 4  
Status (*class in hut2.protocol*), 5  
sw () (*hut2.protocol.HUT2 method*), 4  
sw\_off () (*hut2.protocol.HUT2 method*), 4  
sw\_on () (*hut2.protocol.HUT2 method*), 4

## W

wait () (*hut2.protocol.HUT2 method*), 5