ConnectionHandler
local.connection_handler.
LocalConnectionHandler

```
+ tcp
+ udp
+ message_queue
+ handle_connections
_task_end_now
+ msg_ttl_task
+ handle_connections_task
- _attr_current_addr
_connections
- _attr_network_interface
- _attr_controller_data
- _attr_devices
- _attr_tcp
- _attr_udp
- _attr_server_ip
- _attr_message_queue
- __attr_send_loop_sleep
- __attr_tasks_done
- __attr_tasks_undone
- _attr_handlec_connections_task
- _attr_handle_connections
_task_end_now
- __attr_read_tcp_task
- __send_loop_sleep
- __tasks_undone
- __read_tcp_task
```

```
+ None __init__(self,
 ControllerData controller
_data, str server_ip="0.0.0.0",
 str|None network_interface=None)
+ str|None network_interface
(self)
+ None network_interface
(self, str|None network
_interface)
+ ControllerData controller
_data(self)
+ None controller_data
(self, ControllerData
 controller_data)
+ dict[str, Device] devices
(self)
+ None devices(self,
 dict[str, Device] devices)
+ socket.socket|None
 tcp(self)
+ None tcp(self, socket.socket|
None tcp)
+ socket.socket|None
 udp(self)
+ None udp(self, socket.socket|
None udp)
+ str server_ip(self)
+ None server_ip(self,
 str server_ip)
+ dict[str, list[Message]
] message_queue(self)
+ None message_queue
(self, dict[str, list
[Message]] message_queue)
+ Task|None handle_connections
_task(self)
+ None handle_connections
_task(self, Task|None
 handle_connections_tasks)
+ bool handle_connections
_task_end_now(self)
+ None handle_connections
_task_end_now(self, bool
 handle_connections_task
_end_now)
+ set[str] current_addr
_connections(self)
+ None current_addr_connections
(self, set[str] current_addr
_connections)
+ None shutdown(self)
+ bool bind_ports(self)
+ DeviceTcpReturn process
_device_identity_package
(self, TcpConnection connection,
 bytes data, ReferencePass device_ref)
+ DeviceTcpReturn process
_aes_initial_vector_package
(self, TcpConnection connection,
 bytes data, Device device)
+ DeviceTcpReturn process
_message_answer_package
(self, TcpConnection connection,
 bytes answer, Device device,
 Message|None msg_sent)
+ None remove_msg_from
_queue(self, Message
 msg, Device|None device)
+ DeviceTcpReturn handle
_connection(self, ReferencePass
 device_ref, TcpConnection connection,
 ReferencePass msg_sent_r)
+ DeviceTcpReturn process
_tcp_package(self, TcpConnection
 connection, ReferencePass data
_ref, Message|None msg_sent, ReferencePass
 device_ref)
+ DeviceTcpReturn device
_handle_local_tcp(self,
 Device device, TcpConnection
 connection)
+ bool send_udp_broadcast
(self)
+ None standby(self)
+ tuple[list[Any], list
[Any], list[Any]]|None
 read_incoming_tcp_con
_task(self)
+ None check_messages
_time_to_live(self)
+ None connection_tasks
_time_to_live(self, int
 proc_timeout_secs=DEFAULT
_MAX_COM_PROC_TIMEOUT_SECS)
+ None handle_incoming
_tcp_connection(self,
 int proc_timeout_secs)
+ bool handle_connections
(self, int proc_timeout
_secs=DEFAULT_MAX_COM_PROC
_TIMEOUT_SECS)
+ None handle_connections
_task_stop(self)
+ None search_and_send
_loop_task_alive(self)
+ Message|None send_message
(self, list[Command] send
_msgs, UnitId target_device
_uid, float time_to_live_secs
=-1.0, **Any kwargs)
+ None discover_devices
(self, float timeout
_secs=2.5)
+ LocalConnectionHandler
 create_default(Any cls,
 ControllerData controller
_data, str server_ip="0.0.0.0",
 str|None network_interface=None)
- Task|None __send_loop
_sleep(self)
- None __send_loop_sleep
(self, Task|None send
_loop_sleep)
- list[tuple[Task, datetime.datetime,
 datetime.datetime]] __tasks
_done(self)
- None __tasks_done(self,
 list[tuple[Task, datetime.datetime,
 datetime.datetime]] tasks_done)
- list[tuple[Task, datetime.datetime]
] __tasks_undone(self)
- None __tasks_undone
(self, list[tuple[Task,
 datetime.datetime]] tasks
_undone)
- Task|None __read_tcp
_task(self)
- None __read_tcp_task
(self, Task|None read
_tcp_task)
```