



wireless control for everyone

COMMAND-LINE SPECIFICATION

Version 2.3.0

August 14, 2012

Copyright © 2012, Threshold Corporation

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- All products derived from ONE-NET designs will be implemented in adherence to the System Identification (SID) usage guidelines, including both registered and unregistered implementations of the SID.
- Neither the name of Threshold Corporation (trustee of ONE-NET) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

1	Overview	1
2	Command Format	2
3	Verbosity Level	3
4	Commands	3
4.1	list	3
4.1.1	Arguments	3
4.1.2	Replies	4
4.1.3	Examples	5
4.2	invite	6
4.2.1	Arguments	6
4.2.2	Replies	6
4.2.3	Examples	7
4.3	join	7
4.3.1	Arguments	7
4.3.2	Replies	8
4.3.3	Examples	8
4.4	sniff	8
4.4.1	Arguments	9
4.4.2	Replies	9
4.4.3	Example – Verbose Level 1	9
4.4.4	Example – Verbose Level 2	10
4.4.5	Example – Verbose Level 3	10
4.4.6	Example – Verbose Level 6	11
4.5	channel	12
4.5.1	Arguments	12
4.5.2	Replies	12
4.5.3	Examples	12
4.6	change key	13
4.6.1	Arguments	13
4.6.2	Replies	14
4.6.3	Examples	14
4.7	cancel invite	15

4.7.1	Arguments	15
4.7.2	Replies	15
4.7.3	Examples	15
4.8	assign peer	15
4.8.1	Arguments	15
4.8.2	Replies	16
4.8.3	Examples	16
4.9	unassign peer	17
4.9.1	Arguments	17
4.9.2	Replies	17
4.9.3	Examples	18
4.10	single	19
4.10.1	Arguments	19
4.10.2	Replies	19
4.10.3	Examples	20
4.11	set pin	20
4.11.1	Arguments	20
4.11.2	Replies	21
4.11.3	Examples	22
4.12	query pin	22
4.12.1	Arguments	22
4.12.2	Replies	22
4.12.3	Examples	23
4.13	fast query pin	23
4.13.1	Arguments	23
4.13.2	Replies	24
4.13.3	Examples	24
4.14	status pin	24
4.14.1	Arguments	24
4.14.2	Replies	25
4.14.3	Examples	25
4.15	block	26
4.15.1	Arguments	26
4.15.2	Replies	27

4.15.3	Examples.....	27
4.16	stream.....	27
4.16.1	Arguments	28
4.16.2	Replies	28
4.16.3	Examples.....	28
4.17	single text	29
4.17.1	Arguments	29
4.17.2	Replies	29
4.17.3	Examples.....	30
4.18	change fragment delay	31
4.18.1	Arguments	31
4.18.2	Replies	31
4.18.3	Examples.....	32
4.19	set flags.....	32
4.19.1	Arguments	32
4.19.2	Replies	33
4.19.3	Examples.....	33
4.20	change keep-alive	34
4.20.1	Arguments	34
4.20.2	Replies	34
4.20.3	Examples.....	35
4.21	remove device	35
4.21.1	Arguments	35
4.21.2	Replies	35
4.21.3	Examples.....	35
4.22	user pin	36
4.22.1	Arguments	36
4.22.2	Replies	36
4.22.3	Examples.....	37
4.23	echo.....	37
4.23.1	Arguments	37
4.23.2	Reply	37
4.23.3	Examples.....	38
4.24	save	38

4.24.1	Arguments	38
4.24.2	Replies	38
4.24.3	Examples	38
4.25	erase	39
4.25.1	Arguments	39
4.25.2	Replies	39
4.25.3	Examples	39
5	Debugging Tools Commands	40

1 Overview

The ONE-NET Command-Line Mode (OCM) is designed to enable ONE-NET evaluation boards to perform a variety of actions based on commands entered via the serial port.

2 Command Format

Command prompts begin with the string “**ocm-**”, followed by a designation of the Device connected to the terminal. For the Master, this is “**m**”; for a Client running in Serial Mode, it is “**c**”; for a Client running in Auto Mode, it is “**c1**”, “**c2**”, or “**c3**” (depending on the setting of the node select switches); for a Device running in “sniffer mode”, it is “**s**”. The prompt string terminates with the “**>**” character.

The format for command entry is the command name, optionally followed by one or more arguments (as required by the command). Arguments, if present, are immediately preceded by a single colon character, which acts as a delimiter.

Generic examples:

command
command:argument
command:argument1:argument2
command:argument1:argument2:argument3

Specific examples:

list
invite:7777-7777
channel:US:14
single:1:2:002:1250000002
assign peer:002:3:003:1

There should be no spaces immediately before or after the colons.

All input is case insensitive except for quoted text strings.

All commands generate replies. If the command executes successfully, the default reply is **OK**. If an error occurs, an error message is displayed.

For many commands, the initial **OK** is simply an indication that the command was entered and initiated successfully; when the command completes, an additional reply message (either success or failure) is generated.

3 Verbosity Level

All commands are lower-case. All command listings below assume a verbosity level of 1. To get more detailed printouts, raise the verbosity level. To get no printouts, set the verbosity level to 0. Do this with the “verbose level” command. See below for more details.

Command names are case sensitive. Hexadecimal digit arguments are case-insensitive. Hence

single:1:2:002:aaaaaaaaaa

and

single:1:2:002:AAAAAAAAAA

are the same, but

SINGLE:1:2:002:aaaaaaaaaa

would be invalid because “SINGLE” needs to be lower-case.

Quoted strings are also case-sensitive, as are invite keys (see below).

4 Commands

4.1 list

The *list* command is valid for a Master or a Client. This prints the SID, channel, encryption key, peer table, state of each unit, client table (master only), peer assignments state of the User Pins (0 = off, 1 = on), and other information

4.1.1 Arguments

The *list* command does not take any arguments.

4.1.2 Replies

Reply	Meaning
OK	List was displayed

Table 4.1 – list Command Replies

4.1.3 Examples

```
ocm-m> list
ONE-NET Evaluation Version 2.3.0 (Build 107)
# of Network MH Devices : 2
# of Network MH Repeaters : 1
Message key : (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-0e-0f)
Old Message key : (00-00-00-00) - (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b)

NID: 0x0000000001
DID: 0x001

Channel: US 5
Client count: 1
  Client 1 : DID: 0x002

Peer table:
  001:2:002:1
  001:3:002:3

User pins:
  0 output state: 0
  1 input state: 0
  2 output state: 0
  3 input state: 0
OK
```

4.2 invite

The *invite* command is only valid for a Master. This command initiates the invitation of a Client to join the network. Only one outstanding invitation is allowed at a time, so if successive *invites* are issued, only the first will succeed, unless the first completes before the second is issued.

4.2.1 Arguments

The *invite* command takes one or two arguments, the one-time key of the Client to be added and an optional timeout time in milliseconds. If no timeout time is given, ten minutes is used.

Argument	Format
One-time key of Client to invite	XXXX-XXXX (ASCII hex)
Timeout time in milliseconds	Positive Integer

Table 4.2 – *invite* Command Argument

4.2.2 Replies

Reply	Meaning
OK	Invitation initiated successfully
The "invite" command failed – already in progress	Tried a second <i>invite</i> before first finished
The "invite" command failed – invalid format	Parameter was not entered properly
The "invite" command failed – invalid command for CLIENT	<i>invite</i> can only be sent by Master

Table 4.3 – *invite* Command Replies

4.2.3 Examples

```
ocm-m> invite:aaaa-bbbb
OK
ocm-m> Updating ADD DEVICE on 002 succeeded.
Updating ADD DEVICE succeeded.

ocm-m> invite:bbbb-cccc
The "invite" command failed - already in progress

ocm-m> invite:aaaa-bbbb:10000
OK
ocm-m> Device 2222-3333 not added.  Invite timed out.
```

4.3 join

The *join* command is only valid when in Serial Mode. If the device is a Master, the command switches the device to Client mode and begins listening for an invitation to join the network.

When the command is used without parameters, the device will scan all available channels looking for an invitation and will keep scanning indefinitely.

Optional parameters can be specified with the *join* command. If a timeout time is specified, the client will scan the channels for that length of time, then give up if it has not received an invitation within that time period. Additionally, a specific channel can be specified. If a channel is specified, the client will not scan the channels, but will instead only listen on the specified channel.

4.3.1 Arguments

The *join* command takes either 0, 2, or 3 arguments. In the two argument version, the first argument is the region(US or EUR) and the second argument is the channel number. In the three argument version, the first argument is the timeout time in seconds, the second argument is the region(US or EUR) and the third argument is the channel number.

Argument	Format
Timeout time in seconds	Positive Integer
Region specification	"EUR" or "US"
Channel to look for invitations on	Integer (1–25) for US region; (1–3) for EUR region

Table 4.4 – join Command Argument

4.3.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "join" command failed – invalid format	Command not formatted correctly

Table 4.5 – join Command Replies

4.3.3 Examples

```
ocm-c> join
OK
...
Successfully joined network as 002
```

```
ocm-c> join:US:5
OK
...
Successfully joined network as 002
```

```
ocm-c> join:10000:US:6
OK
...
Invite process timed out.
```

4.4 sniff

The *sniff* command is valid when in Serial Mode. This command restarts the Device as a packet sniffer (not part of the ONE-NET network). This sniffer simply listens on a given channel and prints out packets that it hears. The verbosity level of the packets will vary

with the verbosity level. See “verbose level” command below.

4.4.1 Arguments

The *sniff* command takes two arguments, a region specification followed by the channel on which to sniff.

Argument	Format
Region specification	“EUR” or “US”
Channel to sniff	Integer (1–25) for US region; (1–3) for EUR region

Table 4.6 – sniff Command Argument

4.4.2 Replies

Reply	Meaning
OK	Command was successful
The “sniff” command failed – invalid format	Region Specification or Channel number was invalid

Table 4.7 – sniff Command Replies

4.4.3 Example – Verbose Level 1

```
ocm-c> sniff:EUR:5
The “sniff” command failed – invalid format
```

```
ocm-m> sniff:US:10
OK
```

```
6018 received 30 bytes:
55 55 55 33 B4 BC DA B4 B3 B4 B4 B4 B4 BC B4
BC B5 B4 69 D4 C9 B2 B4 55 A4 65 52 33 99
```

4.4.4 Example – Verbose Level 2

```
259759 received 30 bytes:
55 55 55 33 B4 BC A6 B4 B3 B4 B4 B4 B4 BC B4
BC B5 B4 D4 33 92 C9 63 A3 A5 54 9C 34 BC
```

```
Raw PID=100(00)--MH=False SA=False
```

```
Enc. Msg CRC : 0xA6
Repeater DID : 0xB4BC
Dest. DID : 0xB4B3
NID : 0xB4B4B4B4B4BC
Source DID : 0xB4BC
Encoded Payload Length : 11
D4 33 92 C9 63 A3 A5 54 9C 34 BC
```

4.4.5 Example – Verbose Level 3

```
7294 received 30 bytes:
55 55 55 33 B4 BC AA B4 B3 B4 B4 B4 B4 BC B4
BC B5 B CA 3A 6A C5 D2 D3 65 92 A2 33 3C
```

```
Raw PID=100(00)--MH=False SA=False
```

```
Enc. Msg CRC : 0xAA -- Decoded Msg. CRC : 0x6C
Calculated Raw Msg CRC : 0x6C
Repeater DID : 0xB4BC -- Decoded : 0x001
Dest. DID : 0xB4B3 -- Decoded : 0x002
NID : 0xB4B4B4B4B4BC -- Decoded : 0x0000000001
Source DID : 0xB4BC -- Decoded : 0x001
Encoded Payload Length : 11
CA 3A 6A C5 D2 D3 6592 A2 33 3C
Decoded Payload (# of Bytes = 9)
2D 3C CC FF AD 2F 7D 24 40
Decrypted using key (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-0e-0f)
62 02 80 50 0F 10 00 01
Payload CRC = 0x62, Calculated Payload CRC = 0x62, CRCs match.
Msg. ID=028
Decrypted using key (04-05-06-07) - (08-09-0a-0b) - (0c-0d-0e-0f) - (44-44-44-44)
40 AC C8 7B 6D 99 12 19
Payload CRC = 0x40, Calculated Payload CRC = 0xAC, CRCs do not match.
Msg. ID=ACC
```


4.4.6 Example – Verbose Level 6

13318 received 30 bytes:

55 5 55 33 B4 BC D6 B4 B3 B4 B4 B4 B4 BC B4
BC B5 B4 B5 A3 D3 33 99 9C AA A9 66 96 3C

Raw PID=100(00)--MH=False SA=False

Enc. Msg CRC : 0xD6 -- Decoded Msg. CRC : 0xF8

Calculated Raw Msg CRC : 0xF8

Repeater DID : 0xB4BC -- Decoded : 0x001

Dest. DID : 0xB4B3 -- Decoded : 0x002

NID : 0xB4B4B4B4B4BC -- Decoded : 0x000000001

Source DID : 0xB4BC -- Decoded : 0x001

Encoded Payload Length : 11

B5 A3 D3 33 99 9C AA A9 66 96 3C

Decoded Payload (# of Bytes = 9)

11 AE 92 B6 96 DD DA E4 40

Decrypted using key (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-0e-0f)

68 02 90 50 0F 10 00 01

Payload CRC = 0x68, Calculated Payload CRC = 0x68, CRCs match.

Msg. ID=029

Single -- Msg Type=0x0(App)

App payload : 0x500F100001 : Class-->0x500 : Type-->0x00 : Src Unit-->0xF : Dst

Unit-->0x1 : Data-->0x00001

Decrypted using key (04-05-06-07) - (08-09-0a-0b) - (0c-0d-0e-0f) - (44-44-44-44)

98 55 74 57 C9 44 7E 3F

Payload CRC = 0x98, Calculated Payload CRC = 0x66, CRCs do not match.

Msg. ID=557

4.5 channel

The *channel* command is only valid in Serial Mode. This command restarts the device as a Master on the given channel (all Clients must subsequently be re-invited to the network).

4.5.1 Arguments

The *channel* command takes two arguments, a region specification followed by the channel on which to operate.

Argument	Format
Region specification	"EUR" or "US"
Channel to use	Integer value: 1–25 for US region, 1–3 for EUR region

Table 4.8 – channel Command Argument

4.5.2 Replies

Reply	Meaning
OK	Command was successful
The "channel" command failed – invalid format	Channel specification was invalid

Table 4.9 – channel Command Replies

4.5.3 Examples

```
ocm-m> channel:US:2  
OK
```

```
ocm-m> channel:EUR:3  
OK
```

```
ocm-m> channel  
The "channel" command failed - invalid format
```

```
ocm-c> channel:EUR:4  
The "channel" command failed - invalid format
```

4.6 change key

The *change key* command is only valid for a Master. This command initiates updating of the encryption key for all Clients in the network. Only one outstanding key change is allowed at a time; if successive *change keys* are invoked.

4.6.1 Arguments

The *change key* command takes a single argument, the key fragment with which to update the current key.

Argument	Format
Key fragment to update	XX-XX-XX-XX (ASCII hex)

Table 4.10 – change key Command Argument

4.6.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "change key" command failed – already in progress	Tried a second key change before first finished
The "change key" command failed – invalid format	Command not formatted correctly
The "change key" command failed – invalid command for CLIENT	Key can only be changed by a Master
Updating NETWORK KEY succeeded	Key successfully updated in all Clients

Table 4.11 – change key Command Replies

4.6.3 Examples

```
ocm-m> change key:55-55-55-55
```

```
OK
```

```
ocm-m> Updating NETWORK KEY on 002 succeeded.
```

```
Single transaction with 002; return status: SUCCESS
```

```
Updating NETWORK KEY succeeded.
```

```
ocm-m> change key:00-11-AA-BB
```

```
The "change key" command failed – already in progress
```

```
ocm-m> change key:CC-DD-EE-GG (invalid hex digit - GG)
```

```
The "change key" command failed – invalid format
```

```
ocm-c> change key:AA-BB-CC-DD
```

```
The "change key" command failed – invalid command for CLIENT
```

4.7 cancel invite

The *cancel invite* command is only valid for a Master. It cancels the currently active invite action (if there is one).

4.7.1 Arguments

The *cancel invite* command doesn't take any arguments.

4.7.2 Replies

Reply	Meaning
OK	Command was successful

Table 4.12 – *cancel invite* Command Replies

4.7.3 Examples

```
ocm-m> cancel invite
Device 5555-6666 not added.  Invite cancelled.
OK
```

4.8 assign peer

The *assign peer* command is only valid for a Master. It is used to assign a peer to a given Device/Unit.

4.8.1 Arguments

The *assign peer* command takes four arguments, best thought of as two “argument pairs”. The first pair indicates the Device/Unit to whom a peer is being assigned, and the second pair the Device/Unit that is being assigned as the first Device/Unit's peer.

Argument	Format
Device whose peer is being assigned	XXX (three ASCII hex digits)
Unit whose peer is being assigned	X (one ASCII hex digit)
Device to assign peer to	XXX (three ASCII hex digits)
Unit to assign peer to	X (one ASCII hex digit)

Table 4.13 – *assign peer* Command Arguments

4.8.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "assign peer" command failed – invalid format	Command not formatted correctly
Invalid command for CLIENT	Peers can only be assigned by Master
Invalid DID and/or unit	Destination Device/Unit pair does not exist in the network

Table 4.14 – assign peer Command Replies

4.8.3 Examples

```
ocm-m> assign peer:003:1:004:1
The "assign peer" command failed - Invalid DID and/or unit
ocm-m> assign peer:3:1:4:1
The "assign peer" command failed - invalid format
ocm-m> assign peer:002:3:001:2
OK
ocm-m> Updating ASSIGN PEER on 002 succeeded.
Single transaction with 002; return status: SUCCESS
ocm-c> assign peer:003:6:004:2
Invalid command for CLIENT
```

4.9 unassign peer

The *assign peer* command is only valid for a Master. It is used to unassign a peer that was previously assigned with the “assign peer” command.

4.9.1 Arguments

The *unassign peer* command takes four arguments, best thought of as two “argument pairs”. The first pair indicates the Device/Unit to whom a peer is being unassigned, and the second pair the Device/Unit that is being unassigned as the first Device/Unit’s peer.

Argument	Format
Device whose peer is being unassigned	XXX (three ASCII hex digits)
Unit whose peer is being unassigned	X (one ASCII hex digit)
Device to unassign peer from	XXX (three ASCII hex digits)
Unit to unassign peer from	X (one ASCII hex digit)

Table 4.15 – unassign peer Command Arguments

4.9.2 Replies

Reply	Meaning
OK	Command initiated successfully
The “unassign peer” command failed – invalid format	Command not formatted correctly
Invalid command for CLIENT	Peers can only be assigned by Master
Invalid DID and/or unit	Destination Device/Unit pair does not exist in the network

Table 4.16 – unassign peer Command Replies

4.9.3 Examples

```
ocm-m> unassign peer:003:1:004:1
The "unassign peer" command failed - Invalid DID and/or unit
ocm-m> unassign peer:3:1:4:1
The "unassign peer" command failed - invalid format
ocm-m> unassign peer:002:3:001:2
OK
ocm-m> Updating UNASSIGN PEER on 002 succeeded.
Single transaction with 002; return status: SUCCESS
ocm-c> unassign peer:003:6:004:2
Invalid command for CLIENT
```


4.10 single

The *single* command is valid for both Masters and Clients. It is used to send a single message to another ONE-NET Device.

4.10.1 Arguments

The *single* command takes two arguments. The first argument is the 12-bit destination address (DID) of the packet, or a null entry, the latter indicating that the message is to be sent to all peers of the source Unit (when using a null entry, the destination Unit number should be set to 0x000). The second argument is the five bytes of packet data.

Argument	Format
Destination DID	XXX (three ASCII hex digits), or null (which represents all peer Clients/Units)
Data	XXXXXXXXXX (ten ASCII hex digits)

Table 4.17 – single Command Arguments

4.10.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "single" command failed – invalid format	Command not formatted correctly
The "single" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "single" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.18 – single Command Replies

4.10.3 Examples

```
ocm-m> single:002:0051100001
OK
ocm-m> Single transaction with 002; return status: SUCCESS
Unit 1 of 002 has state 1.
ocm-m> single:000:0051F00001OK
ocm-m> Single transaction with 002; return status: SUCCESS
Unit 1 of 002 has state 1.
Single transaction with 002; return status: SUCCESS
Unit 3 of 002 has state 1.
ocm-m> single:1:1:1
The "single" command failed - invalid format
ocm-m>
```

4.11 set pin

The *set pin* command is valid for both Masters and Clients. It is used to set a relay level on another ONE-NET Device.

4.11.1 Arguments

The *set pin* command takes three arguments. The first argument is the 12-bit destination address (DID) of the packet. The second argument is the unit on the destination device that should be set. The third argument is the new state of the destination unit (0 for low, 1 for high).

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Unit To Be Set	Non-negative integer
Destination DID	Pin State (0 = low, 1 = high)

Table 4.19 – set pin Command Arguments

4.11.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "set pin" command failed – invalid format	Command not formatted correctly
The "set pin" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "set pin" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.20 – set pin Command Replies

4.11.3 Examples

```
ocm-m> set pin:002:1:1
OK
ocm-m> Single transaction with 002; return status: SUCCESS
Unit 1 of 002 has state 1.

ocm-m> set pin:002:1
The "set pin" command failed - invalid format
ocm-m>
```

4.12 query pin

The *query pin* command is valid for both Masters and Clients. It is used learn the status of a pin on another ONE-NET device.

4.12.1 Arguments

The *query pin* command takes two arguments. The first argument is the 12-bit destination address (DID) of the packet. The second argument is the unit on the destination device that is being queried.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Unit To Be Queried	Non-negative integer

Table 4.21 – query pin Command Arguments

4.12.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "query pin" command failed – invalid format	Command not formatted correctly
The "query pin" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "query pin" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.22 – query pin Command Replies

4.12.3 Examples

```
ocm-m> query pin:002:3
OK
ocm-m> Single transaction with 002; return status: SUCCESS
Unit 3 of 002 has state 1.
ocm-m> query pin:002
The "query pin" command failed - invalid format
ocm-m>
```

4.13 fast query pin

The *fast query pin* command is valid for both Masters and Clients. It is exactly the same as the *query pin* command except that the result arrives more quickly since the pin statement is sent in the Acknowledgement packet in the fast query and it is sent later in a regular query.

4.13.1 Arguments

The *fast query pin* command takes two arguments. The first argument is the 12-bit destination address (DID) of the packet. The second argument is the unit on the destination device that is being queried.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Unit To Be Queried	Non-negative integer

Table 4.23 – fast query pin Command Arguments

4.13.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "fast query pin" command failed – invalid format	Command not formatted correctly
The "fast query pin" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "fast query pin" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.24 – fast query pin Command Replies

4.13.3 Examples

```
ocm-m> fast query pin:002:3
OK
ocm-m> Single transaction with 002; return status: SUCCESS
Unit 3 of 002 has state 1.
ocm-m> query pin:002
The "fast query pin" command failed - invalid format
ocm-m>
```

4.14 status pin

The *status pin* command is valid for both Masters and Clients. It sends a unit's pin status (0 or 1) to another device.

4.14.1 Arguments

The *status pin* command takes two arguments. The first argument is the 12-bit destination address (DID) of the packet. The second argument is the unit on the source device whose status is being sent.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Source Unit	Non-negative integer

Table 4.25 – status pin Command Arguments

4.14.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "status pin" command failed – invalid format	Command not formatted correctly
The "status pin" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "status pin" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.26 – status pin Command Replies

4.14.3 Examples

```
ocm-m> status pin:002:3
OK
ocm-m> Single transaction with 002; return status: SUCCESS

(On Client)
ocm-c> Unit 3 of 001 has state 1.

ocm-m> status pin:002
The "status pin" command failed - invalid format
ocm-m>
```

4.15 block

The *block* command is valid for both Masters and Clients. It is used to send a block message to another ONE-NET Device. See the document ***Block And Stream Messages In ONE-NET*** for more details. It will send 25 a's, then 25 b's, then 25 c's, ..., then 25 z's, then start with the a's again.

4.15.1 Arguments

The block command takes three arguments. The first argument is the raw Device ID of the destination device. The second argument is the number of bytes to transfer. The third argument is the “chunk pause” (see the document ***Block And Stream Messages In ONE-NET*** for more details).

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Number Of Bytes To Transfer	Positive Integer
Chunk Pause (in milliseconds)	Positive Integer

Table 4.27 – block Command Arguments

4.15.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "block" command failed – invalid format	Command not formatted correctly
The "block" command failed – already in progress	Tried sending a second block transaction before first finished
The "block" command failed –required resources unavailable	Resources needed to complete the command are unavailable
The "block" command failed – invalid DID and/or unit	Destination Device does not exist in the network

Table 4.28 – block Command Replies

4.15.3 Examples

```
ocm-m> block:002:625:50
OK
```

625 bytes will be sent will be sent to device 002 with a chunk pause of 50 milliseconds. 'a' to 'z' will be sent, 25 of each letter.

4.16 stream

The *stream* command is valid for both Masters and Clients. It is used to send a stream message to another ONE-NET Device. See the document ***Block And Stream Messages In ONE-NET*** for more details.

4.16.1 Arguments

The stream command takes three arguments. The first argument is the raw Device ID of the destination device. The second argument is the time in milliseconds for the transfer (see the document *Block And Stream Messages In ONE-NET* for more details).

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Transfer time (in milliseconds)	Positive Integer

Table 4.29 – stream Command Arguments

4.16.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "stream" command failed – invalid format	Command not formatted correctly
The "stream" command failed – already in progress	Tried sending a second stream transaction before first finished
The "stream" command failed –required resources unavailable	Resources needed to complete the command are unavailable
The "stream" command failed – invalid DID and/or unit	Destination Device does not exist in the network

Table 4.30 – stream Command Replies

4.16.3 Examples

```
ocm-m> stream:006:10000  
OK
```

Stream packets will be sent to device 006 for ten seconds (10,000 milliseconds)

4.17 single text

The *single text* command is valid for both Masters and Clients. It is used to send a single text message to another ONE-NET Device.

4.17.1 Arguments

The *single text* command takes four arguments. The first and second arguments are, respectively, the source and destination Units for the packet. The third argument is the 12-bit destination address (DID) of the packet, or a null entry, the latter indicating that the message is to be sent to all peers of the source Unit (when using a null entry, the destination Unit number should be set to 0x000). The fourth and final argument is a quoted ASCII string.

Argument	Format
Source Unit	X (one ASCII hex digit)
Destination Unit	X (one ASCII hex digit)
Destination DID	XXX (three ASCII hex digits), or null (which represents all peer Clients/Units)
Data	Quoted text (1–17 ASCII characters)

Table 4.31 – *single text* Command Arguments

4.17.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "single text" command failed – invalid format	Command not formatted correctly
The "single text" command failed – invalid DID and/or unit	Destination Device/Unit pair does not exist in the network
The "single text" command failed – required resources unavailable	Resources needed to complete the command are unavailable

Table 4.32 – *single text* Command Replies

4.17.3 Examples

```
ocm-m> single text:2:3:002:high:"ab"
```

```
OK
```

```
ocm-m> Single transaction with 002; return status: SUCCESS
```

```
ocm-m> single text:2:3:002:high:"abcdefghijklmnopq"
```

```
OK
```

```
ocm-m> Single transaction with 002; return status: SUCCESS
```

```
ocm-m> single text:hello
```

```
The "single text" command failed - invalid format
```

4.18 change fragment delay

The *change fragment delay* command is only valid for a Master. It is used to change the Fragment Delay of a ONE-NET Client.

4.18.1 Arguments

The *change fragment delay* command takes three arguments. The first argument is the address (DID) of the Device whose Fragment Delay is to be changed. The second argument is the priority of the Fragment Delay to change. The third argument is the new Fragment Delay (in milliseconds) to set; note that the high priority value must be equal to or smaller (i.e., a shorter delay) than the low priority one.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Priority	"low" or "high"
New Fragment Delay	Integer value (16-bit, unsigned)

Table 4.33 – change fragment delay Command Arguments

4.18.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "change fragment delay" command failed – invalid format	Command not formatted correctly, or improper delay change for specified priority
The "change fragment delay" command failed – invalid command for CLIENT	Fragment Delay can only be changed by a Master.

Table 4.34 – change fragment delay Command Replies

4.18.3 Examples

```
ocm-m> change fragment delay:002:high:35
```

```
OK
```

```
...
```

```
Updating FRAGMENT DELAY on 002 succeeded
```

```
ocm-m> change fragment delay:002:low:abc
```

```
The "change fragment delay" command failed - invalid format
```

```
ocm-c> change fragment delay:002:low:13
```

```
The "change fragment delay" command failed - invalid command for  
CLIENT
```

4.19 set flags

The *set flags* command is only valid for a Master. It is used to set the flags of a ONE-NET Client device.

4.19.1 Arguments

The *set update master flag* command takes two arguments. The first argument is the address (DID) of the Device whose flags are to be set. The second argument specifies whether the new flags value.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
New flags value	XX (two ASCII hex digits)

Table 4.35 – *set flags* Command Arguments

4.19.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "set flags" command failed – invalid format	Command not formatted correctly
The "set flags" command failed – invalid command for CLIENT	Set flag can only be implemented by a Master

Table 4.36 – set flags Command Replies

4.19.3 Examples

```
ocm-m> set flags:002:D8
```

```
OK
```

```
...
```

```
Updating SETTINGS on 002 succeeded
```

```
ocm-m> set update master flag:XX
```

```
The "set update master flag" command failed - invalid format
```

```
ocm-c> set update master flag:003:44
```

```
The "set update master flag" command failed - invalid command for  
CLIENT
```

4.20 change keep-alive

The *change keep-alive* command is only valid for a Master. It is used to set the Keep-Alive Interval value in a ONE-NET Client.

4.20.1 Arguments

The *change keep-alive* command takes two arguments. The first argument is the address (DID) of the Device whose Keep-Alive Interval is to be changed. The second argument specifies the new value (in milliseconds) of this Keep-Alive Interval.

Argument	Format
Destination DID	XXX (three ASCII hex digits)
Interval	Integer value (32-bit, unsigned)

Table 4.37 – change keep-alive Command Arguments

4.20.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "change keep-alive" command failed – invalid format	Command not formatted correctly
The "change keep-alive" command failed – invalid command for CLIENT	Keep-Alive Interval can only be set by a Master

Table 4.38 – change keep-alive Command Replies

4.20.3 Examples

```
ocm-m> change keep-alive:002:3600000
```

```
OK
```

```
...
```

```
Updating KEEP-ALIVE INTERVAL on 002 succeeded
```

```
ocm-c> change keep-alive:005:7200000
```

```
The "change keep-alive" command failed - invalid command for CLIENT
```

4.21 remove device

The *remove device* command is only valid for a Master. It is used to inform a ONE-NET Client of its removal from the network.

4.21.1 Arguments

The *remove device* command takes one argument, the address (DID) of the Device that is being removed from the network.

Argument	Format
Destination DID	XXX (three ASCII hex digits)

Table 4.37 – remove device Command Argument

4.21.2 Replies

Reply	Meaning
OK	Command initiated successfully
The "remove device" command failed – invalid format	Command not formatted correctly
The "remove device" command failed – invalid command for CLIENT	Device removal can only be performed by a Master

Table 4.38 – remove device Command Replies

4.21.3 Examples

```
ocm-m> remove device:004
```

OK

...

Updating REMOVE DEVICE on 002 succeeded.

Updating REMOVE DEVICE succeeded.

```
ocm-c> remove device:007
```

The "remove device" command failed - invalid command for CLIENT

4.22 user pin

The *user pin* command is valid for Masters and Clients. It is used to configure individual input/output pins. A user pin set to "input" will, when toggled, send a simple switch message to its assigned peers; one set to "output" will assert that pin based on a received Switch message destined for its associated Unit.

4.22.1 Arguments

The *user pin* command takes two arguments. The first argument is the user pin number to be configured. The second argument is the state to which the specified user pin will be set.

Argument	Format
User Pin Number	Integer value (0–255)
Pin State	"input", "output", or "disable"

Table 4.39 – user pin Command Arguments

4.22.2 Replies

Reply	Meaning
OK	Command was successful
The "user pin" command failed – invalid format	Command not formatted correctly

Table 4.40 – user pin Command Replies

4.22.3 Examples

```
ocm-c> user pin:2:input  
OK
```

```
ocm-m> user pin:3:output  
OK
```

```
ocm-c> user pin:1:disable  
OK
```

4.23 echo

The *echo* command is valid for Masters and Clients. It is used to enable and disable command echoing for the Device on which it is issued.

4.23.1 Arguments

The *echo* command takes a single argument, the desired echo state of the associated Device.

Argument	Format
Echo State	"on" or "off"

Table 4.41 – echo Command Argument

4.23.2 Reply

Reply	Meaning
OK	Command was successful
The "echo" command failed – invalid format	Command not formatted correctly

Table 4.42 – echo Command Replies

4.23.3 Examples

```
ocm-c> echo: on  
OK
```

```
ocm-c> echo off  
The "echo" command failed - invalid format
```

```
ocm-m> echo: off  
OK
```

4.24 save

The *save* command is valid for Masters and Clients. It is used to save the configuration of the network.

4.24.1 Arguments

The *save* command doesn't take any arguments.

4.24.2 Replies

Reply	Meaning
OK	Command was successful

Table 4.43 – save Command Replies

4.24.3 Examples

```
ocm-m> save  
OK
```

4.25 erase

The *erase* command is valid for Masters and Clients. It is used to clear the non-volatile memory of a device. It is the opposite of the save command.

4.25.1 Arguments

The *erase* command doesn't take any arguments

4.25.2 Replies

Reply	Meaning
OK	Command was successful

Table 4.44 erase Command Replies

4.25.3 Examples

```
ocm-m> erase
OK
```

5 Debugging Tools Commands

Please see the document *Debugging Tools And Techniques* for more information on how to use the following commands (note that these are generally only used for debugging tools).

1. Pausing commands
 - a. ratchet - similar to breakpoints
 - b. proceed - similar to breakpoints
 - c. pause - stops all ONE-NET timers
2. Changing timer constants and slowing things down
 - a. interval - adjust timer constants
 - b. csdq - slows down clock
3. Memory commands - adjusting RAM
 - a. memload
 - b. memdump
4. verbose level - adjusts the verbosity of printouts