Acquisition Device's Commands Description

Version 0.1

Revision history			
Authors	Date	Version	Description
Dimitrije Lilić, Haris Turkmanović	03.03.2024	0.1	Initial commands description

Contents

1. Control service description

Within the device firmware, the logic for control and status messages is encapsulated within a distinct service. This service is responsible for receiving control messages, parsing their content, executing the appropriate actions based on the message content, and generating responses accordingly. The core logic and operational mechanisms of this service are depicted in the Figure 1.
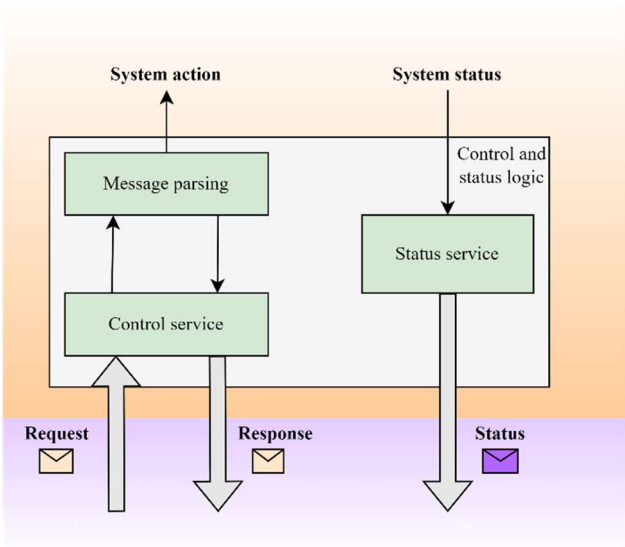

Figure 1 - Control and status service logic

Control messages are transmitted to the Acquisition device in the form of control message requests. These requests originate from the HOST side and are transmitted through a previously established TCP port on the Acquisition device. Upon receipt, the control message request is processed within the Control service server task and forwarded to the Message Parsing logic for analysis of the message content. The Message Parsing logic, which is an integral part of the Control service, is responsible for examining the content of received control messages and invoking the appropriate callback function assigned to each specific control message. Upon execution of the callback functions corresponding to the requested control messages, responses are generated based on the results of the function execution. These responses are then transmitted back to the HOST machine.

All control messages are in ASCII format with predefined structures. Control messages structures are following:

Control message type:	Request	
Control message structure:	`device<ws>[<wsa><argumentsList>]<eom>`	
Fields description		
Field name	**Field description**	**Optional/Mandatory**
`<ws>`	White space character	Mandatory
`<argumentsList>`	List of command arguments defined in - key=value format separated by <ws>.	Optional
`<wsa>`	White space character inside arguments list block	Optional
`<eom>`	End of message is compound of two ASCI characters: '\r\n'	Mandatory

Control message type	Response	
Control message structure	`<mps><ws><ces>[<wsa><argument List>]<eom>`	

Fields description		
Field name	**Field description**	**Optional/Mandatory**
<mps>	Message parsing status. There are two possible values: 1. OK – message is successfully parsed. 2. ERROR - there are issues during parsing messages. Usually caused by invalid messages format or unsupported command.	Mandatory
<ws>	White space character	Mandatory
<ces>	Callback execution status. There are two possible values: 1. OK – callback is successfully executed. 2. ERROR – there is an error during callback execution. Usually caused by invalid messages format or unsupported command.	Mandatory
<wsa>	White space character inside arguments list block	Optional
<argumentsList>	List of command arguments defined in - key=value format separated by <wsa>.	Optional
<eom>	End of message is compound of two ASCI characters: '\r\n'	Mandatory

2. Supported commands list

Here will be listed all supported commands.

2.1. General device commands

2.1.1. Ping device

This command is used to ping device and obtain device name

Command name	
device hello	
Command Arguments	
Key	*Value description*
Return Value	
`<status><ws><deviceName><eom>`	
Return arguments	
Key	*Value description*
deviceName	Acquisition Device name
Example	
Command: device hello Response: OK AcqDevice\r\n	
Example description:	Ping device to obtain device name

2.1.2. Set device name

This command is used to set device name

Command name	
device setname	
Command Arguments	
Key	*Value description*
-value	*Specify device name*
Return Value	
`<status><eom>`	
Return arguments	
Key	*Value description*
/	/

Example
Command: device setname -value=OpenEPT Response: OK \r\n

Example description:	Set device name to "OpenEPT" and get OK as response to notify that device is successfully set

2.2. Status link related commands

Here are listed all information related to the commands that operates over status link.

2.2.1. Create status link

Create status link.

Command name	
device slink create	
Command Arguments	
Key	*Value description*
Return Value	
<status><ws><linkInstanceNo><ws><eom>	
Return arguments	
Key	*Value description*
linkInstanceNo	Link Instance number
Example	
Command: device slink create Response: OK 0 \r\n	

Example description:	Create status link, and get instance number of link created from device

2.2.2. Send status message over IP

Send message over status link

Command name	
device slink send Proveri	
Command Arguments	
Key	*Value description*
-value	*Message to send over status link*
Return Value	
`<status><ws><"""><ws><eom>`	
Return arguments	
Key	*Value description*
Example	
Command: device slink send –value=? Response: OK \r\n	
Example description:	Send message over status link to device

2.3. Stream related commands

Here are listed all information related to the commands that operates over stream.

2.3.1. Create stream link

Create stream link client that will send stream samples on predefined server. Server IP information are sent to Acquisition device though this command.

Command name	
device stream create	
Command Arguments	
Key	*Value description*
-value	*Stream server ip in ipv4 format*
Return Value	
<status><ws><OK><ws><eom>	
Return arguments	
Key	*Value description*
streamIDString	Stream ID
Example	
Command: device stream create –sid=0 –value=192.168.2.100 Response: OK 0 \r\n	
Example description:	Creates stream link with IP address and gets grom device unique stream ID identifier.

2.3.2. Start samples acquisition

Start stream link that is defined with stream ID.

Command name	
device stream start	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* `device stream create command`
Return Value	
`<status><ws><OK><ws><eom>`	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
`Command: device stream start –sid=0` `Response: OK OK \r\n`	
Example description:	Starts streaming of specified of stream with specified ID

2.3.3. Stop samples acquisition

Stop stream link that is defined with stream ID.

Command name	
device stream stop	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* `device stream create command`
Return Value	
`<status><ws><OK><ws><eom>`	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
`Command: device stream stop` `Response: OK OK \r\n`	
Example description:	Stops streaming of specified of stream with specified ID

2.4. Analog input related command

Here are listed all information related to the operations of analog input.

2.4.1. Set Analog input resolution

Set analog input adc resolution by utilizing system service.

Command name	
`device adc chresolution set`	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* `device stream create` *command*
-value	*Stream's ADC resolution{16, 14, 12, 10} [bit]*
Return Value	
`<status><ws><OK><ws><eom>`	
Return arguments	
Key	*Value description*
`OK`	`Return value OK`
Example	
Command: device adc chresolution set –sid=0 –value=12 Response: OK OK \r\n	
Example description:	Set ADC resolution 12 to stream with ID 0. Command returns OK to notify that everything is set successfully.

2.4.2. Get Analog input resolution

Get analog input adc resolution by utilizing system service.

Command name
device adc chresolution get

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by device stream create command*

Return Value
`<status><ws><resolution><ws><eom>`

Return arguments	
Key	*Value description*
Resolution	Resolution read from device

Example
Command: device adc chresolution get –sid=0 Response: OK 16bit \r\n

Example description:	Get ADC resolution of stream with ID 0. Command returns resolution to notify that everything is set successfully.

2.4.3. Set Analog input clock divider

Set analog input adc clock divider by utilizing system service.

Command name	
device adc chclkdiv set	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* `device stream create` *command*
-value	*Stream's ADC clock divider {1, 4, 8, 16, 32, 64, 128, 256}*
Return Value	
`<status><ws><OK><ws><eom>`	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
`Command: device adc chclkdiv set -sid=0 -value=16` `Response: OK OK \r\n`	
Example description:	Set ADC clock div 16 to stream with ID 0. Command returns OK to notify that everything is set successfully.

2.4.4. Get Analog input clock divider

Set analog input adc clock divider by utilizing system service.

Command name
device adc chclkdiv get

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by device stream create command*

Return Value
<status><ws><clockDivider><ws><eom>

Return arguments	
Key	*Value description*
clockDivider	Clock divider read from device

Example
Command: device adc chclkdiv get –sid=0 Response: OK 16 \r\n

Example description:	Get ADC clock div of stream with ID 0. Command returns clock div to notify that everything is set successfully.

2.4.5. Set Analog input channel sampling time

Set analog input adc sampling time by utilizing system service.

Command name	
device adc chstime set	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* device stream create *command*
-value	*Stream's ADC channel sampling time {1, 2, 8, 16, 32, 64, 378, 810}*
Return Value	
<status><ws><OK><ws><eom>	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
Command: device adc chstime set –sid=0 –value=64 Response: OK OK \r\n	
Example description:	Set ADC channel sampling time 64 to stream with ID 0. Command returns OK to notify that everything is set successfully.

2.4.6. Get Analog input sampling time

Get analog input adc sampling time by utilizing system service.

Command name
device adc chstime get
Command Arguments

Key	Value description
-sid	*Unique stream ID obtained by* device stream create command

Return Value
<status><ws><samplingTime><ws><eom>

Return arguments

Key	Value description
samplingTime	Sampling time read from device

Example

```
Command: device adc chstime get –sid=0
Response: OK 32 \r\n
```

Example description:	Get ADC channel sampling time of stream with ID 0. Command returns channel sampling time to notify that everything is set successfully.

2.4.7. Set Analog input averaging ratio

Set analog input adc averaging ratio by utilizing system service.

Command name	
device adc chavrratio set	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* device stream create *command*
-value	*Stream's ADC averaging ratio {1, 2, 4, 8, 16, 32, 64, 128, 256}*
Return Value	
<status><ws><OK><ws><eom>	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
Command: device adc chavrratio set –sid=0 –value=2 Response: OK OK \r\n	
Example description:	Set ADC averaging ratio 2 to stream with ID 0. Command returns OK to notify that everything is set successfully.

2.4.8. Get Analog input averaging ratio

Get analog input adc averaging ratio by utilizing system service.

Command name
`device adc chavrratio get`

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by `device stream create` command*

Return Value
`<status><ws><averagingRatio><ws><eom>`

Return arguments	
Key	*Value description*
averagingRatio	Averaging ratio read from device

Example
Command: device adc chavrratio get –sid=0 Response: OK 128 \r\n

Example description:	Get ADC averaging ratio of stream with ID 0. Command returns averaging ratio to notify that everything is set successfully

2.4.9. Set Analog input sampling time

Set analog input adc sampling time by utilizing system service.

Command name
device adc stime set

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* device stream create *command*
-value	*Stream's ADC sampling time* {value} [us]

Return Value
<status><ws><OK><ws><eom>

Return arguments	
Key	*Value description*
OK	Return value OK

Example	
Command: device adc stime set –sid=0 –value=50 Response: OK OK \r\n	
Example description:	Set ADC input sampling time 50 to stream with ID 0. Command returns OK to notify that everything is set successfully

2.4.10. Get Analog input sampling time

Get analog input sampling time by utilizing system service.

Command name
device adc stime get

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by device stream create command*

Return Value
\<status>\<ws>\<samplingTime>\<ws>\<eom>

Return arguments	
Key	*Value description*
samplingTime	Sampling time read from device

Example
Command: device adc stime get –sid=0 Response: OK 50 \r\n

Example description:	Get ADC input sampling time of stream with ID 0. Command returns sampling time to notify that everything is set successfully

2.4.11. Set Analog input voltage offset

Set analog input adc voltage offset by utilizing system service.

Command name	
device adc chvoffset set	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* device stream create *command*
-value	*Stream's ADC voltage offset {value} [mV]*
Return Value	
<status><ws><OK><ws><eom>	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
Command: device adc chvoffset set –sid=0 –value=100 Response: OK OK \r\n	
Example description:	Set ADC input voltage offset 100 to stream with ID 0. Command returns OK to notify that everything is set successfully

2.4.12. Get Analog input voltage offset

Get analog input adc voltage offset by utilizing system service.

Command name
device adc chvoffset get

Command Arguments	
Key	*Value description*

Return Value
\<status>\<ws>\<voltageOffset>\<ws>\<eom>

Return arguments	
Key	*Value description*
voltageOffset	Voltage offset read from device

Example
Command: device adc chvoffset get –sid=0 Response: OK 100 \r\n

Example description:	Get ADC input voltage offset of stream with ID 0. Command returns voltage offset to notify that everything is set successfully

2.4.13. Set Analog input current offset

Set analog input adc current offset by utilizing system service.

Command name	
device adc chcoffset set	
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by* device stream create *command*
-value	*Stream's ADC current offset {value} [mA]*
Return Value	
<status><ws><OK><ws><eom>	
Return arguments	
Key	*Value description*
OK	Return value OK
Example	
Command: device adc chvoffset set –sid=0 –value=100 Response: OK OK \r\n	
Example description:	Set ADC input current offset 100 to stream with ID 0. Command returns OK to notify that everything is set successfully

2.4.14. Get Analog input current offset

Get analog input adc current offset by utilizing system service.

Command name
device adc chcoffset get
Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by device stream create command*
Return Value	
`<status><ws><currentOffset><ws><eom>`	
Return arguments	
Key	*Value description*
currentOffset	Current offset read form device
Example	
Command: device adc chcoffset get –sid=0 Response: OK 100 \r\n	
Example description:	Get ADC input current offset of stream with ID 0. Command returns current offset to notify that everything is set successfully

2.4.15. Get Analog input clock

Get analog input adc clock by utilizing system service.

Command name
device adc clk get

Command Arguments	
Key	*Value description*
-sid	*Unique stream ID obtained by device stream create command*

Return Value
<status><ws>< adcInputClk ><ws><eom>

Return arguments	
Key	*Value description*
adcInputClk	Input Clock read form device

Example
Command: device adc clk get –sid=0 Response: OK 80000000 \r\n

Example description:	Get ADC input clock of stream with ID 0. Command returns input clock to notify that everything is set successfully