sas32kd

Release 1.0.0

avc

Feb 05, 2021

CONTENTS:

CHAPTER

ONE

SAS 32 KD LIBRARY

A library to simplify communication with TCP/IP Server Module for Sierra Automated Systems audio routers SAS 32KD.

Communication protocol description is available here: http://www3.sasaudio.com/downloads/SAS%20Documentation/USI_Protocol_32KD_Rev_13.pdf.

1.1 Source Code

https://github.com/dgalus/sas32kd

1.2 PyPI package

https://pypi.org/project/sas32kd/

1.3 Documentation

https://sas32kd.rtfd.io

Documentation can be also found in the *docs/* directory as reStructuredText files. Build the docs by running *make* in the *docs/* directory, then viewing *docs/_build/html/index.html*.

1.4 Installation

```
python -m pip install sas32kd
```

1.5 Author

dgalus / avc <avcsec@protonmail.com>

CHAPTER

TWO

SAS LIBRARY EXAMPLES

Import library:

```
from sas32kd import Sas32kd
```

Connect to SAS audio router TCP/IP Server Module:

```
sas = Sas32kd(ip="10.10.10.10", port=1270, timeout=5)
```

Port and timeout are optional arguments.

Disconnecting from TCP/IP Server Module:

```
sas.disconnect()
```

Take command: *take(input: int, output: int)*

```
res = sas.take(10, 300)
```

- inp: Input channel number
- outp: Output channel number

returns: *Reply.OK* or *Reply.ERROR*

Enhanced take command: *enhanced_take(inp: int, outp: int, gain: int, options: EnhancedTakeOptions)*

```
res = sas.enhanced_take(
  10,
  300,
  1500,
  EnhancedTakeOptions(
    PriorityLevel.STANDARD,
    ControlOptions.ON,
    ActionOptions.SUM,
    SuppliedGainValueUsage.YES,
    CurrentXpointTransitionCtlSpec.YES
  )
)
```

- inp: Input channel number.
- outp: Output channel number.
- gain: Source target gain level (1/10 dB steps, 1024 = Unity; valid 0 to 2048).
- options: Options. (Refer: EnhancedTakeOptions class).

returns: *Reply.OK* or *Reply.ERROR*

Relay command: *relay(action: RelayAction, num: int)*

```
res = sas.relay(RelayAction.LATCH, 123)
```

- action: Action to be performed on relay. (Refer: RelayAction class).
- num: Relay number.

returns: *Reply.OK* or *Reply.ERROR*

Opto command: *opto(action: OptoAction, num: int)*

```
res = sas.opto(OptoAction.ON, 123)
```

- action: Action to be performed on opto. (Refer: OptoAction class).
- num: Opto number.

returns: *Reply.OK* or *Reply.ERROR*

Salvo command: *salvo(option: SalvoOption, num: int)*

```
res = sas.salvo(SalvoOption.ACTUAL_SALVO_NUM, 10)
```

- option: Ordering type of salvos. (Refer: SalvoOption class).

- num: Salvo number.

returns: *Reply.OK* or *Reply.ERROR*

Crosspoint transition control: *crosspoint_transition_control(setting: CrosspointTransitionControlSetting, fade_in_time: FadeTime, fade_out_time: FadeTime, channel: int)*

```
res = sas.crosspoint_transition_control(
  CrosspointTransitionControlSetting.FADE_OUT_FADE_IN,
  FadeTime.T_5S,
  FadeTime.T_5S,
  30
)
```

- setting: Type of transition. (Refer: CrosspointTransitionControlSetting class).
- fade_in_time: Fade in time. (Refer: FadeTime class).
- fade_out_time: Fade out time. (Refer: FadeTime class).
- channel: Output channel number.

returns: *Reply.OK* or *Reply.ERROR*

Gain change command: *gain_change(inp: int, outp: int, gain: int, fade_time: FadeTime, stage: GainChangeStage)*

```
res = sas.gain_change(
  123,
  234,
  1500,
  FadeTime.T_3S,
  GainChangeStage.OUTPUT_GAIN_TRIM
)
```

- inp: Input channel number.
- outp: Output channel number.
- gain: Source target gain level (1/10 dB steps, 1024 = Unity; valid 0 to 2048).
- fade_time: Fade time. (Refer: FadeTime class).
- stage: Gain change stage. (Refer: GainChangeStage class).

returns: *Reply.OK* or *Reply.ERROR*

Stereo link modifier: *stereo_link(option: StereoLinkOption, setting: StereoLinkSetting, channel: int)*

```
res = sas.stereo_link(
  StereoLinkOption.INPUT_LINK,
  StereoLinkSetting.LR_MONO_SUM,
  234
)
```

- option: Input or output link. (Refer: StereoLinkOption class).
- setting: Mono, stereo, source dependent or LR mono sum. (Refer: StereoLinkSetting class).
- channel: Output channel number.

returns: *Reply.OK* or *Reply.ERROR*

Console module control command: *console_module_control(action: ConsoleModuleAction, console_id: int, source: int)*

```
res = sas.console_module_control(
  ConsoleModuleAction.TURN_MODULE_ON_WITH_SOURCE_SELECTED,
  2,
  234
)
```

- action: Console source/module control options. (Refer: ConsoleModuleAction class).
- console_id: System console number (1 to 256 or 999 = any).
- source: Source channel number (1 to 9998).

returns: *Reply.OK* or *Reply.ERROR*

Console module channel label override command: *console_module_channel_label_override(self, console_id: int, source: int, label: str)*

```
res = sas.console_module_channel_label_override(
  2,
  3,
  "PGM X   "
)
```

- console_id: System console number (1 to 256 or 999 = any).
- source: Source channel number (1 to 9998).
- label: 8 character alpha label to be displayed by the addressed modules.

returns: *Reply.OK* or *Reply.ERROR*

Inquiry command: *inquiry(outp: int)*

```
res = sas.inquiry(123)
```

- outp: Output channel (1 to 256 or 999 - any).

returns: Three digit input assigned to specified output or inputs assigned to each output in ascending order.

Expanded channel inquiry command: *expanded_channel_inquiry(destination: int)*

```
res = sas.expanded_channel_inquiry(123)
```

- destination: Destination channel number (1 to 9998).

returns: Dict with all sources currently assigned to output, with priority levels.

Alphanumeric name inquiry command: *alphanumeric_name_inquiry(input_output: AlphanumericNameInquiryInputOutput, channel_num: int)*

```
res = sas.alphanumeric_name_inquiry(
  AlphanumericNameInquiryInputOutput.INPUT,
  123
)
```

- input_output: Input or output. (Refer: AlphanumericNameInquiryInputOutput class).
- channel_num: Channel number. (1 to 256. 998 - all channel sorted alphabetically. 999 - all channels in order of channel number).

returns: Alpha label for input or output.

Feedback command: *feedback(replies: FeedbackReplies, feedback_tally: FeedbackTally, feedback_protocol: FeedbackProtocol)*

```
res = sas.feedback(
  FeedbackReplies.ENABLED,
  FeedbackTally.NUMERICAL_TALLY_AND_ALPHA_CHANGE_NOTIFICATION,
  FeedbackProtocol.THREE_DIGIT_ASCII_STYLE_XPOINT_TALLY
)
```

- replies: Enable or disable replies. (Refer: FeedbackReplies class).
- feedback_tally: Feedback tally variants. (Refer: FeedbackTally class).
- feedback_protocol: Feedback protocol variants. (Refer: FeedbackProtocol class).

returns: *Reply.OK* or *Reply.ERROR*.

CHAPTER

THREE

SAS LISTENER LIBRARY EXAMPLES

Library to attach own actions on SAS event occurences.

Import library:

```
from sas32kd import Sas32kdListener
```

Connect to SAS audio router TCP/IP Server Module:

```
sas = Sas32kdListener(ip="10.10.10.10", port=1270)
```

Port is optional arguments.

Disconnecting from TCP/IP Server Module:

```
sas.disconnect()
```

Attach action on opto turned on: *on_opto_turned_on(opto_num: int, func, *args, **kwargs)*

```
def func(arg):
  print("OPTO ON " + str(arg))

sas.on_opto_turned_on(222, func, "some argument")
```

- opto_num: Opto number.
- func: Function.

- args: Function args.
- kwargs: Function kwargs.

Attach action on opto turned off: *on_opto_turned_off(opto_num: int, func, *args, **kwargs)*

```
def func(arg):
  print("OPTO OFF " + str(arg))

sas.on_opto_turned_off(222, func, "some argument")
```

- opto_num: Opto number.
- func: Function.
- args: Function args.
- kwargs: Function kwargs.

Attach action on relay turned on: *on_relay_turned_on(relay_num: int, func, *args, **kwargs)*

```
def func(arg):
  print("RELAY ON " + str(arg))

sas.on_relay_turned_on(222, func, "some argument")
```

- relay_num: Relay number.
- func: Function.
- args: Function args.
- kwargs: Function kwargs.

Attach action on relay turned off: *on_relay_turned_off(relay_num: int, func, *args, **kwargs)*

```
def func(arg):
  print("RELAY OFF " + str(arg))

sas.on_relay_turned_off(222, func, "some argument")
```

- relay_num: Relay number.
- func: Function.
- args: Function args.
- kwargs: Function kwargs.

Attach action on take: *on_take(self, inp: int, outp: int, func, *args, **kwargs)*

```
def func(arg):
  print("TAKE " + str(arg))

sas.on_take(222, 333, func, "some argument")
```

- inp: Input number.
- outp: Output number.
- func: Function.
- args: Function args.
- kwargs: Function kwargs.

After attaching functions, execute run() function!

```
sas.run()
```

CHAPTER

FOUR

ENUMS

```
class RelayAction(IntEnum):
    """Possible relay actions."""
    MOMENTARY_ACTIVATION = 1
    LATCH = 2
    RELEASE = 3
    INQUIRY = 9


class OptoAction(IntEnum):
    """Possible opto actions."""
    OFF = 0
    ON = 1
    INQUIRY = 9


class SalvoOption(IntEnum):
    """Describes how salvos are organized."""
    ACTUAL_SALVO_NUM = 1
    ALPHANUMERIC_POSITION_OF_SALVO = 2


class CrosspointTransitionControlSetting(IntEnum):
    """Possible transition control settings."""
    CUT_OUT_CUT_IN = 0
    CUT_OUT_FADE_IN = 1
    FADE_OUT_CUT_IN = 2
    FADE_OUT_FADE_IN = 3
    CROSS_FADE = 4
    CUT_OUT_CUT_IN_WITH_DSP = 9


class FadeTime(IntEnum):
    """Possible fade times."""
    INSTANT = 0
    T_10MS = 1
    T_50MS = 2
    T_100MS = 3
    T_200MS = 4
    T_500MS = 5
    T_1S = 6
    T_2S = 7
    T_3S = 8
    T_4S = 9
    T_5S = 10
```

(continues on next page)

(continued from previous page)

```
    T_6S = 11
    T_7S = 12
    T_8S = 13
    T_9S = 14
    T_10S = 15


class GainChangeStage(IntEnum):
    """Possible gain change stages."""
    SOURCE_INPUT_SENSITIVITY = 1
    OUTPUT_GAIN_TRIM = 3
    DSP_COEFFICIENT_LEVEL_NO_FADE_TIME = 10
    DSP_COEFFICIENT_LEVEL_FADE_TIME = 11
    DSP_MIXER_OUTPUT_MASTER_LEVEL_NO_FADE_TIME = 15
    DSP_MIXER_OUTPUT_MASTER_LEVEL_FADE_TIME = 16


class StereoLinkOption(IntEnum):
    """Describes if input or output should be modified."""
    INPUT_LINK = 0
    OUTPUT_LINK = 1


class StereoLinkSetting(IntEnum):
    """Types of inputs and outputs."""
    MONO = 0
    STEREO = 1
    SOURCE_DEPENDENT = 2
    LR_MONO_SUM = 3


class EnhancedTakeOptions:
    """Describes arguments for enhancement_take() method."""

    class PriorityLevel(IntEnum):
        """Possible priority levels."""
        STANDARD = 0
        IFB = 1

    class ControlOptions(IntEnum):
        """Possible control options."""
        OFF = 0
        ON = 1
        MOMENTARY = 2

    class ActionOptions(IntEnum):
        """Possible actions."""
        TAKE = 0
        SUM = 1
        DIRECT_RELAY_CONTROL = 2

    class SuppliedGainValueUsage(IntEnum):
        """Should use specified gain value."""
        NO = 0
        YES = 1

    class CurrentXpointTransitionCtlSpec(IntEnum):
```

(continues on next page)

(continued from previous page)

```
        """Should use current xpoint transition control specification."""
        NO = 0
        YES = 1

    def __init__(self,
                 priority_level: PriorityLevel,
                 control_options: ControlOptions,
                 action_options: ActionOptions,
                 use_supplied_gain_value: SuppliedGainValueUsage,
                 use_current_xpoint_transition_ctl_spec:␣
↪CurrentXpointTransitionCtlSpec
                 ):
        self.value = int(priority_level)
        self.value += int(control_options) << 2
        self.value += int(action_options) << 5
        self.value += int(use_supplied_gain_value) << 9
        self.value += int(use_current_xpoint_transition_ctl_spec) << 10

    def get(self):
        """
        Get calculated enhanced take command options numeric value.
        :return: value calculated for arguments provided in constructor.
        """
        return self.value


class ConsoleModuleAction(IntEnum):
    """Possible console module actions."""
    TURN_MODULE_OFF_WITH_SOURCE_SELECTED = 0
    TURN_MODULE_ON_WITH_SOURCE_SELECTED = 1
    TURN_CUE_OFF_ON_MODULE_WITH_SOURCE_SELECTED = 2
    TURN_CUE_ON_ON_MODULE_WITH_SOURCE_SELECTED = 3


class AlphanumericNameInquiryInputOutput(IntEnum):
    """Alphanumeric name inquiry should be shown for input or output."""
    INPUT = 0
    OUTPUT = 1


class FeedbackReplies(IntEnum):
    """Feedback replies should be enabled or disabled."""
    ENABLED = 1
    DISABLED = 0


class FeedbackTally(IntEnum):
    """Possible feedback tally options."""
    NO_TALLY_OF_XPOINT_ACTIVITY_OR_ALPHA_CHANGE_NOTIFICATION = 0
    XPOINT_TALLY_IN_NUMERICAL_FORMAT_ONLY = 1
    XPOINT_TALLY_AS_CHANNEL_ALPHA_LABELS_ONLY = 2
    XPOINT_TALLY_BOTH_NUMERICAL_AND_ALPHA_LABELS = 3
    NOTIFICATION_OF_CHANGES_TO_THE_ALPHA_LABELS = 4
    NUMERICAL_TALLY_AND_ALPHA_CHANGE_NOTIFICATION = 5
    ALPHA_LABEL_TALLY_AND_ALPHA_CHANGE_NOTIFICATION = 6
    NUMERICAL_AND_ALPHA_LABEL_TALLY_WITH_ALPHA_CHANGE_NOTICE = 7
    NOTICE_OF_CONSOLE_MODULE_OPERATIONS_ONLY = 8
```

(continues on next page)

(continued from previous page)

```
    NOTICE_OF_CONSOLE_MODULE_OPERATIONS_AND_NUMERICAL_TALLY_AND_ALPHA_CHANGE_
↪NOTIFICATION = 9


class FeedbackProtocol(IntEnum):
    """Possible feedback variants."""
    THREE_DIGIT_ASCII_STYLE_XPOINT_TALLY = 0
    TWO_DIGIT_ASCII_HEX_STYLE_XPOINT_TALLY = 1
    FOUR_DIGIT_ASCII_STYLE_XPOINT_TALLY = 2


class Reply(IntEnum):
    """Default replies."""
    OK = 0
    ERROR = 1
```

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search