tenma-serial

Release 1.0.0

Jul 03, 2020

Contents:

CHAPTER 1

tenma API

Contents

1.1 Instantiation

You can use the tenma class you deem necessary by directly creating an instance of the class referencing your model.

The api can autodetect the model and return a proper instance (or a default one which may or may not suit your limits) by simply using the provided function:

```
from tenma import instantiate_tenma_class_from_device_response
tenma = instantiate_tenma_class_from_device_response('/dev/ttyUSB0')
```

or the more manual (with your own knowledge) approach:

```
from tenma import Tenma72_2550
tenma = Tenma72_2550('/dev/ttyUSB0')
```

In case something does not work as you would expect, instantiate the class with the debug option enabled so all the transmissions to and from the unit are printed to stdout:

```
from tenma import Tenma72_2550
tenma = Tenma72_2550('/dev/ttyUSB0')
```

1.2 API Documentation

tenmaDcLib is small python library to control a Tenma 72-XXXX programmable DC power supply, either from USB or Serial.

Supported models:

- 72_2545 -> tested on HW
- 72_2535 -> Set as manufacturer manual (not tested)
- 72_2540 -> Set as manufacturer manual (not tested)
- 72_2550 -> Set as manufacturer manual (not tested)
- 72_2930 -> Set as manufacturer manual (not tested)
- 72_2940 -> Set as manufacturer manual (not tested)

Other units from Korad or Vellman might work as well since they use the same serial protocol.

class tenma.tenmaDcLib.**Tenma72Base**(*serialPort*, *debug=False*)

Control a tenma 72-XXXX DC bench power supply

Defaults in this class assume a 72-2540, use subclasses for other models

OFF()

Turns OFF the output

ON()

Turns on the output

getStatus()

Returns the power supply status as a dictionary of values

- ch1Mode: "C.V | C.C"
- ch2Mode: "C.V | C.C"
- **tracking:**
  - 00=Independent
  - 01=Tracking series
  - 11=Tracking parallel
- BeepEnabled: True | False
- lockEnabled: True | False
- outEnabled: True | False

getVersion()

Returns a single string with the version of the Tenma Device and Protocol user

recallConf(*conf*)

Load existing configuration in Memory. Same as pressing any Mx button on the unit

runningCurrent(*channel*)

Returns the current read of a running channel

runningVoltage(*channel*)

Returns the voltage read of a running channel

saveConf(*conf*)
Save current configuration into Memory.

Does not work as one would expect. SAV(4) will not save directly to memory 4. We actually need to recall memory 4, set configuration and then SAV(4)

saveConfFlow(*conf*, *channel*)
Performs a full save flow for the unit. Since saveConf only calls the SAV<NR1> command, and that does not work as advertised, or expected, at least in 72_2540.

This will:

- turn off the output
- Read the voltage that is set
- recall memory conf
- Save to that memory conf

Parameters

- **conf** – Memory index to store to
- **channel** – Channel with output to store

setBEEP(*enable=True*)
Enable or disable BEEP

There's no feedback from the serial connection to determine whether BEEP was set or not.

Parameters **enable** – Boolean to enable or disable

setOCP(*enable=True*)
Enable or disable OCP.

There's no feedback from the serial connection to determine whether OCP was set or not.

Parameters **enable** – Boolean to enable or disable

setOVP(*enable=True*)
Enable or disable OVP

There's no feedback from the serial connection to determine whether OVP was set or not.

Parameters **enable** – Boolean to enable or disable

class tenma.tenmaDcLib.**Tenma72_2535**(*serialPort*, *debug=False*)

MATCH_STR = '72-2535'

MAX_MA = 3000

MAX_MV = 30000

NCHANNELS = 1

NCONFS = 5

class tenma.tenmaDcLib.**Tenma72_2540**(*serialPort*, *debug=False*)

MAX_MA = 5000

MAX_MV = 30000

NCHANNELS = 1

NCONFS = 5

Only 4 physical buttons. But 5 memories are available

class tenma.tenmaDcLib.**Tenma72_2545**(*serialPort*, *debug=False*)

MATCH_STR = '72-2545'

MAX_MA = 2000

MAX_MV = 60000

NCHANNELS = 1

NCONFS = 5

class tenma.tenmaDcLib.**Tenma72_2550**(*serialPort*, *debug=False*)

MATCH_STR = '72-2550'

MAX_MA = 3000

MAX_MV = 60000

NCHANNELS = 1

NCONFS = 5

class tenma.tenmaDcLib.**Tenma72_2930**(*serialPort*, *debug=False*)

MATCH_STR = '72-2930'

MAX_MA = 10000

MAX_MV = 30000

NCHANNELS = 1

NCONFS = 5

class tenma.tenmaDcLib.**Tenma72_2940**(*serialPort*, *debug=False*)

MATCH_STR = '72-2940'

MAX_MA = 5000

MAX_MV = 60000

NCHANNELS = 1

NCONFS = 5

exception tenma.tenmaDcLib.**TenmaException**

tenma.tenmaDcLib.**instantiate_tenma_class_from_device_response**(*device*, *debug=False*)

Get a proper Tenma subclass depending on the version response from the unit.

The subclasses mainly deal with the limit checks for each unit.

CHAPTER 2

tenma-control

Contents

2.1 General Description

tenma-control (tenmaControl.py) lets you control your bench power supply directly from the command line via a serial connection.

This page provides a set of common examples on how to use the tenma-control command. For a full description of the options provided simply run:

```
tenma-control -h
```

The command always expects its input input as milliAmperes and milliVolts:

2.2 Print the Tenma version

```
tenma-control /dev/ttyUSB0
```

2.3 Set the current and the voltage

For example: 2.2 Amperes 5V:

```
tenma-control -c 2200 -v 5000 /dev/ttyUSB0
```

2.4 Turn on the channel output

```
tenma-control --on /dev/ttyUSB0
```

2.5 Turn off the channel output

```
tenma-control --off /dev/ttyUSB0
```

2.6 Load an existing memory

```
tenma-control -r 1
tenma-control --recall 2
```

2.7 Create a new value for a memory 4

```
tenma-control -c 2200 -v 5000 --save 4 /dev/ttyUSB0
```

2.8 Verbose output

```
tenma-control -c 2200 -v 5000 --save 4 --verbose /dev/ttyUSB0
```

2.9 Serial output

```
tenma-control -c 2200 -v 5000 --save 4 --debug /dev/ttyUSB0
```

CHAPTER 3

tenma-applet

Contents

3.1 General Description

tenma-applet (gtkIndicator.py) lets you control your bench power supply using a simple GTK applet sitting on the system bar.

It provides basic ON/OFF and Memory selection. Current version is limited to the values already set in the unit.

To start it:

```
tenma-applet
```

3.2 Select the serial connection

tenma-applet lists the available serial connections under the *serial* menu. Simply selecting a connection and the applet tries to retrieve the model information and connect to it.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

Index

S

T