Reimagined Quantum Documentation

Release 1.0.0

Juan Barbosa

Apr 11, 2017

CONTENTS:

CHAPTER

ONE

DOCUMENTATION

ReimaginedQuantum library

Created on Mon Apr 10 11:19:25 2017

@author: juan

class reimaginedQuantum.**Channel**(*name*, *port*)

Constants

END_COMMUNICATION = 4

End of message

READ_VALUE = 14

Reading operation signal

START_COMMUNICATION = 2

Begin message signal

WRITE_VALUE = 15

Writing operation signal

construct_message(*read=False*)

Construcs a message with project requirements.

Returns: list: list of bytes containing channel info.

exchange_values(*read=True*)

Exchanges values from computer to utility.

Returns: can return None, or a list containing a *hex_list*

read_value(*hex_list*)

Reads a *hex_list* and updates class attributes values.

set_value(*value*)

Writes and incoming int value to class attributes.

split_value()

Updates the most/least significant byte.

verify_values(*hex_list*)

Verifies if current values have changed.

Returns: bool: The return value. True if values are the same, False otherwise.

class reimaginedQuantum.**CommunicationPort**(*device*, *baudrate=115200*, *timeout=0.02*, *bounce_timeout=20*)

Constants

BAUDRATE = 115200
Default baudrate for the serial port communication

BOUNCE_TIMEOUT = 20
Number of times a specific transmition is tried

BYTE_SIZE = 8
One byte = 8 bits

PARITY = 'N'
Message will not have any parity

STOP_BITS = 1
Message contains only one stop bit

TIMEOUT = 0.02
Maximum time without answer from the serial port

begin_serial()
Initializes pyserial instance.

Returns: pyserial.serial object

Raises: PermissionError: user is not allowed to use port. SerialException: if it could not open port

checksum(*hex_list*)
Implements a simple checksum to verify message integrity.

Returns: bool: The return value. True for success, False otherwise.

message(*content*, *wait_for_answer=False*)
Sends a message, and waits for answer.

Returns:

list: each postion on list is made up with a tuple containing channel and value in hexadecimal base.

Raises: Exception: any type ocurred with during *bounce_timeout*.

read()
Reads a message through the serial port.

Returns: list: hexadecimal values decoded as strings.

Raises: Exception: Noisy answer, or timeout.

receive()
Organices information according to project requirements.

Returns:

list: each position on list is made up with a tuple containing channel and value in hexadecimal base.

Raises: Exception: if wrong checksum.

send(*content*)
Sends a message through the serial port.

Raises: PySerialExceptions

class reimaginedQuantum.**DataChannel**(*prefix*, *port*)
Constants

class reimaginedQuantum.**Detector**(*identifier*, *port*, *data_interval=100*, *timer_check_interval=1000*)
Constants

BASE_DELAY = 1e-09
Default channnel delay time (seconds)

BASE_SLEEP = 1e-09
Default channel sleep time (seconds)

class reimaginedQuantum.**Experiment**(*port*, *number_detectors=2*)
Constants

BASE_COINWIN = 1e-09
Default coincidence window (seconds)

BASE_SAMPLING = 0.001
Default sampling time (seconds)

class reimaginedQuantum.**TimerChannel**(*prefix*, *port*, *base*)
Constants

Graphical User Interface

Created on Tue Apr 11 11:31:32 2017

@author: juan

class mainGUI.**AutoSizeLabel**(*text*, *value*)
from reclosedev at http://stackoverflow.com/questions/8796380/automatically-resizing-label-text-in-qt-strange-behaviour and Jean-Sébastien http://stackoverflow.com/questions/29852498/syncing-label-fontsize-with-layout-in-pyqt

class mainGUI.**Main**

Defines the mainwindow.

Constants

channelsCaller()
creates a property window to define number of channels

choose_file()
user interaction with saving file

eventFilter(*source*, *event*)
Creates event to handle serial combobox opening.

file_changed = None
set

format = None
fig

select_serial(*index*)
Selects port at index position of combobox.

serial_refresh()
Loads serial port described at user combobox.

widget_activate(*status*)
most of the tools will be disabled if there is no UART detected

class mainGUI.**RingBuffer**(*rows*, *columns*, *output_file*, *fmt*, *delimiter='t'*)
Based on https://scimusing.wordpress.com/2013/10/25/ring-buffers-in-pythonnumpy/

extend(*x*)
adds array x to ring buffer

get()
Returns the first-in-first-out data in the ring buffer

save()
Saves the buffer

mainGUI.**heavy_import**()
Imports matplotlib and NumPy.

Useful to be combined with threading processes.

class mainGUI.**propertiesWindow**(*parent=None*)
Defines the channel configuration dialog.

DEFAULT_CHANNELS = 2
Default number of channels

creator(*n*)
creates the spinboxes and labels required by the user

delete(*n*, *N*)
delets unneccesary rows of labels and spinboxes

reset()
sets everything to default

update()
sends message with the updated information

mainGUI.**savetxt**(*file*, *matrix*, *delimiter=' '*, *fmt='%.3f'*, *typ=<class 'float'>*)
Saves data to a text file.

Used to save matrix contents to plain text files. Depening whether or not matrix contains strings or floats uses np.savetxt function.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

m

r

INDEX

T

U

V

W