**hyperdaq.serialcom**

serialcom.py

GaborDAQ functions for communicating with equipment via serial

Last Updated: January 2020

Trevor Arp

Gabor Lab

University of California, Riverside

All Rights Reserved

***class*hyperdaq.serialcom.lakeshore\_335(*com\_port*, *sys\_time*)**

Bases: [**hyperdaq.serialcom.lakeshore\_336**](file:///C:\Users\evers\OneDrive\Desktop\html-hyperDAQ\html-hyperDAQ\hyperdaq.serialcom.html#hyperdaq.serialcom.lakeshore_336)

Defines a controller for a LakeShore Model 336 Temperature Controller

Temperatures are accessible as read-only attributes as self.A, self.B, etc. for temperature sensors A,B,… in Kelvin

**read\_data()**

Reads data from the controller

***class*hyperdaq.serialcom.lakeshore\_336(*com\_port*, *sys\_time*)**

Bases: **[hyperdaq.serialcom.serial\_device\_listener](file:///C:\\Users\\evers\\OneDrive\\Desktop\\html-hyperDAQ\\html-hyperDAQ\\hyperdaq.serialcom.html" \l "hyperdaq.serialcom.serial_device_listener" \o "hyperdaq.serialcom.serial_device_listener)**

Defines a controller for a LakeShore Model 336 Temperature Controller

Temperatures are accessible as read-only attributes as self.A, self.B, etc. for temperature sensors A,B,… in Kelvin

**get\_PID()**

Gets the current gain values of the PID loop

**get\_output\_percent()**

Gets the heater output percentage

**get\_output\_setting()**

Gets the heater output setting

**get\_setpoint()**

Gets the temperature setpoint

**read\_data()**

Reads data from the controller, currently only reads A due to serial port issues

**set\_PID(*P*, *I*, *D*)**

Sets the gain values of the PID loop

**set\_output\_setting(*setting*)**

Sets the heater output setting, 0 is off, 1 is low, 2 is medium and 3 is high

**set\_setpoint(*value*)**

Sets the temperature setpoint

**turn\_heater\_off()**

Turns off the heater

***class*hyperdaq.serialcom.lakeshore\_625(*com\_port*, *sys\_time*)**

Bases: **[hyperdaq.serialcom.serial\_device\_listener](file:///C:\\Users\\evers\\OneDrive\\Desktop\\html-hyperDAQ\\html-hyperDAQ\\hyperdaq.serialcom.html" \l "hyperdaq.serialcom.serial_device_listener" \o "hyperdaq.serialcom.serial_device_listener)**

Defines a controller for a LakeShore Model 625 Superconducting Magnet Power supply

**Parameters**

**com\_port** – string that sets the COM port used by the serial device e.g. “COM6”

**voltage**

The output voltage, Units in Volts

**current**

The output current, Units in Amps

**field**

The central field, computed from the current based on calibration, Units in Tesla

**read\_data()**

Reads the current and voltage and computes the field

***class*hyperdaq.serialcom.mira\_900\_OPO(*com\_port*, *sys\_time*)**

Bases: **[hyperdaq.serialcom.serial\_device\_listener](file:///C:\\Users\\evers\\OneDrive\\Desktop\\html-hyperDAQ\\html-hyperDAQ\\hyperdaq.serialcom.html" \l "hyperdaq.serialcom.serial_device_listener" \o "hyperdaq.serialcom.serial_device_listener)**

Defines a controller for the Coherent Mira 900 OPO controller

**Parameters**

**com\_port** – string that sets the COM port used by the serial device e.g. “COM6”

**wavelength**

The Wavelength, in nm

**power**

The IR power in arbitrary units

**piezo**

the piezo voltage

**RH**

The relative Humidity in the cavity

**get(*command*, *nbytes*)**

**read\_data()**

Reads data from the serial device, override to add functionality

**set\_piezo(*p*)**

**set\_wavelength(*w*)**

**toggle\_manual\_mode()**

Toggles manual mode, when in manual data is not collected continuously, instead only when external function calls are made.

***class*hyperdaq.serialcom.serial\_device(*com\_port*, *Baudrate=57600*, *Parity='O'*, *Stopbits=1*, *Bytesize=7*, *Terminator='rn'*, *Timeout=0.05*)**

Bases: **threading.Thread**

A base class for communicating with serial devices

**Parameters**

**com\_port** – string that sets the COM port used by the serial device e.g. “COM2”

**get\_data(*command*)**

Sends a request for data to the serial device and listens for a response Returns none and prints to terminal if no response was received

**get\_data\_queue()**

returns the data Queue

**get\_raw\_data(*command*)**

Similar to get\_data, but works in a different way, reads until it times out and returns raw serial output as a byte array

**reset(*stoptime=0.5*)**

Resets the serial object

**send\_command(*command*)**

Sends a command and does not wait for a response

**set\_data(*command*)**

Sends a command to the serial device

**stop()**

Closes the serial connection and stops the thread

***class*hyperdaq.serialcom.serial\_device\_listener(*com\_port*, *sys\_time*, *Baudrate=57600*, *Parity='O'*, *Stopbits=1*, *Bytesize=7*, *Terminator='rn'*, *Timeout=0.05*)**

Bases: **[hyperdaq.serialcom.serial\_device](file:///C:\\Users\\evers\\OneDrive\\Desktop\\html-hyperDAQ\\html-hyperDAQ\\hyperdaq.serialcom.html" \l "hyperdaq.serialcom.serial_device" \o "hyperdaq.serialcom.serial_device)**

Defines a controller for a serial device that is continuously taking data

Continuously calls the read\_data() function, override to define read data from the device

**Parameters**

* **com\_port** – string that sets the COM port used by the serial device e.g. “COM2”
* **sys\_time** – a synchronized Stopwatch object for timing

**get\_current\_data(*command*)**

Pauses main task to send command to the device and waits for response Returns None if the device isn’t running

**read\_data()**

Reads data from the serial device, override to add functionality

**read\_data\_call()**

Calls the read\_data function

**run()**

Runs the thread

**set\_data(*command*)**

Sends a command to the serial device

**stop()**

Closes the serial connection and stops the thread

**toggle\_queueing()**

Toggles the Queueing of data, not Queueing by default

***class*hyperdaq.serialcom.spectrapro\_2300(*com\_port*, *sys\_time*)**

Bases: **[hyperdaq.serialcom.serial\_device](file:///C:\\Users\\evers\\OneDrive\\Desktop\\html-hyperDAQ\\html-hyperDAQ\\hyperdaq.serialcom.html" \l "hyperdaq.serialcom.serial_device" \o "hyperdaq.serialcom.serial_device)**

Defines a controller for the SpectraPro-2300 Monochromator

**Parameters**

**com\_port** – string that sets the COM port used by the serial device e.g. “COM6”

**wavelength**

The Wavelength, in nm

**read\_data()**

Reads data from the controller

**set\_wavelength(*w*)**

Sets the wavelength