StellarNet Spectrometer Python Module

This module requires Python 2.7 and depends on pyusb 1.0.x. One way to install it is using pip:

Install pip: http://www.pip-installer.org/en/latest/installing.html
Install pyusb: pip install pyusb==1.0.0a3

After plugging in a device, you should be able to test the basic operation of the driver:

sudo python stellarnet.py info

Access to USB devices on Linux systems is restricted and requires root access. However, it is possible to overcome this restriction by installing the custom udev rules file provided: 99-local.rules. Install this file as /etc/udev/rules.d/99-local.rules and reboot the system to provide non-root access to StellarNet devices.

Class description

class stellarnet.**StellarNet**(*device*)¶
Represents a StellarNet spectrometer.

COEFF_C1_ADDR = *128*¶
The address of the string containing coefficient C1

COEFF_C2_ADDR = *160*¶
The address of the string containing coefficient C2

COEFF_C3_ADDR = *192*¶
The address of the string containing coefficient C3

COEFF_C4_ADDR = *224*¶
The address of the string containing coefficient C4

DEVICE_ID_ADDR = *32*¶
The address of the stored string containing device identification.

Exceptions

exception stellarnet.**ArgRangeError**(*message*)¶
Raised when argument is out of range.

exception stellarnet.**ArgTypeError**(*message*)¶
Raised when argument type is incorrect.

exception stellarnet.**ArgumentError**(*message*)¶
Raised when argument in error.

exception stellarnet.**NotFoundError**(*message*)¶
Raised when USB device cannot be found.

A typical application will involve an operational sequence similar to the following:

`find_devices`()

`get_device_id`()

`get_config`()

`set_config`(***kwargs*)

`read_spectrum`()

Alphabetic Listing

compute_lambda(*pixel*)¶
Compute lambda from the pixel index. Returns the pixel's wavelength (float).

Parameters:
pixel – Integer; the pixel index on which to perform the computation.

get_config()¶
Gets the device configuration. Returns a dictionary.

get_device_id()¶
Gets the device id. Returns a string.

get_stored_bytes(*address*)¶
Get stored bytes. Returns bytearray.

Parameters:
address – Integer; the address of the string to get.

get_stored_string(*address*)¶
Get stored bytes. Returns string.

Parameters:
address – Integer; the address of the string to get.

print_info()¶
Print device information.

read_spectrum()¶
Reads and returns a spectrum from the spectrometer. Returns an array of short integers.

See `StellarNet`.set_config() for a description of the parameters that control the operation of the spectrometer or the post-processing of the spectrum.

set_config(***kwargs*)¶
Sets the device configuration.

Parameters:
- **int_time** – (optional) Integer; the integration time in milliseconds.
- **x_timing** – (optional) Integer; the XTiming rate.
- **x_smooth** – (optional) Integer; the boxcar smoothing window size.
- **scans_to_avg** – (optional) Integer; the # of scans to be averaged together.

- **temp_comp** – (optional) Integer; temperature compensation (not implemented).

set_stored_bytes(*address*, *data*)¶
Set stored bytes.

Parameters:
- **address** – Integer; the address of the string to set.
- **data** – String; the string value to be set

exception stellarnet.**StellarNetError**(*message*)¶
Base class for StellarNet errors.

exception stellarnet.**TimeoutError**(*message*)¶
Raised when device operation times out.

find_devices()¶
Find all USB-connected StellarNet devices.

This function returns a tuple of StellarNet objects or raises NotFoundError if no devices are found.

main(*argv=None*)¶
Excerise StellarNet spectrometer driver.