Chapter 1

WLM class

WLM is used to control a High Finesse WS6 or WS7 wavemeter. Only one WLM object can be instantiated. The company provides a C-library of functions to configure and control the wavemeter, hence the WLM object loads this library ('C:\Windows\System32\wlmData.dll') and the corresponding C-header ('wlmData.hml') in the contructor function. C-functions from the library are called within the matlab functions described below. A complete summary of the C functionality can be found in the wavemeter manual, provided by High Finesse.

MATLAB EXAMPLE

handle = **WLM.getInstance()**
output = *handle.*[anyFunction];

Properties

- *WL*
  Last measured wavelength in nanometers.

- *freq*
  Last measured frequency in THz.

- *pool*
  Handle to parallel worker pool object. Used for toggling the wavemeter

channel in parallel with other matlab scripts for simultaneous wavemeter locking of multiple lasers.

- *toggling*
  String that shows whether toggling of the wavemeter channels is 'on' or 'off'.

- *parf*
  Handle to parallel function, executed by the parallel worker pool *pool*.

- *ar* (private)
  Struct containing the argument name strings from WLM header.

- *active_channel* (private)
  Channel that is currently active.

- *num_channels* (private)
  Total number of switchbox channels.

Functions

handle = **getInstance()**

Instantiates a WLM object and puts it under the name *handle* (or any other name that is put before the = sign).

obj = **WLM()** (private)

Constructor that is called by getInstance(). Here, the C-header is called, its parameters are stored in private property *ar*. Also, the C-library with functions to communicate with the wavemeter is loaded. To enable parallel execution of functions, the library is also loaded into the workers in *pool*.

[WL , freq] = **ReadWLM(** *what , channel* **)** (private)

Switches the wavemeter to switchbox *channel* and reads frequency and/or wavelength, depending on whether *what* is 'WLM', 'freq' or 'both' (cap-sensitive!). Called by GetWL(), GetFreq() and GetBoth().

pool = **CreateParPool()** (private)

Checks if there is a parallel worker pool available and stores it in output *pool*. If not available, it creates a new one. This might take a minute.

ParToggle(*channels* , *time* **)** (private)

Starts an infinite loop that switches to the next channel in vector *channels* every *time* seconds. Called as a parallel function by ToggleChannels().

ToggleChannels(*channels* , *time* , *onOffStr* **)**

Is called by the user to start toggling between channels for simultaneous locking of multiple lasers. If string *onOffStr* is 'on', function ParToggle(*channels* , *time*) is started as a job in the parallel worker pool. A handle to this job is stored in property *parf*. If *onOffStr* is 'off', the job stored in *parf* is canceled and the toggling stops.

SwitchToChannel(*channel* **)**

Changes the active switchbox channel to *channel*, then waits for 0.5s.

freq = **GetFreq(** *channel* **)**

Calls ReadWLM(‘freq’ , *channel*) to measure the frequency for *channel*.

WL = **GetWL(** *channel* **)**

Calls ReadWLM(‘WLM’ , *channel*) to measure the wavelength for *channel*.

[WL , freq] = **GetBoth(** *channel* **)**

Calls ReadWLM(‘both’ , *channel*) to measure both frequency and wavelength for *channel*.

channel = **GetChannel()**

Returns the property *active_channel*.