

# moglabs

## MWM wavemeter



Revision 2.16

## Limitation of Liability

MOG Laboratories Pty Ltd (MOGLabs) does not assume any liability arising out of the use of the information contained within this manual. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of MOGLabs, nor the rights of others. MOGLabs will not be liable for any defect in hardware or software or loss or inadequacy of data of any kind, or for any direct, indirect, incidental, or consequential damages in connections with or arising out of the performance or use of any of its products. The foregoing limitation of liability shall be equally applicable to any service provided by MOGLabs.

## Copyright

Copyright © MOG Laboratories Pty Ltd (MOGLabs) 2014 – 2017. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying or otherwise, without the prior written permission of MOGLabs.

## Contact

For further information, please contact:

MOG Laboratories P/L  
49 University St  
Carlton VIC 3053  
AUSTRALIA  
+61 3 9939 0677  
info@moglabs.com  
www.moglabs.com

MOGLabs USA LLC  
419 14th St  
Huntingdon PA 16652  
USA  
+1 814 251 4363  
info@moglabsusa.com  
www.moglabsusa.com

MOGLabs Europe  
Goethepark 9  
10627 Berlin  
Germany  
+49 30 21 960 959  
info@moglabs.eu

# Preface

The MOGLabs MWM wavemeter provides a compact, high resolution and cost effective instrument for measuring laser wavelength and output spectrum. The instrument is USB and ethernet connected and can also operate standalone with battery power for remote applications. The cost is low enough that you can devote one to each laser and immediately know via your control program that each laser is on resonance and single mode.

We hope that you enjoy using the MWM, and please let us know if you have any suggestions for improvement of the MWM or this document, so that we can make life in the lab better for all.

MOGLabs, Melbourne, Australia  
[www.moglabs.com](http://www.moglabs.com)



# Contents

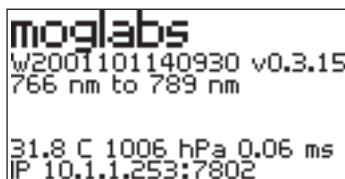
<b>Preface</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Basic setup . . . . .	1
1.2 Front panel display and buttons . . . . .	3
1.3 Software installation . . . . .	3
<b>2 Communications</b>	<b>5</b>
2.1 TCP/IP . . . . .	5
2.2 USB . . . . .	8
<b>3 MOGWAVE</b>	<b>13</b>
3.1 Device discovery . . . . .	13
3.2 Main display . . . . .	14
3.3 Menu . . . . .	15
3.4 Acquisition . . . . .	19
3.5 Options . . . . .	20
3.6 Wavelength . . . . .	20
3.7 Functions . . . . .	21
3.8 Status bar . . . . .	23
<b>4 Calibration</b>	<b>25</b>
4.1 Existing wavelength range . . . . .	25
4.2 Calibrating with HighFinesse wavemeter . . . . .	27
4.3 New wavelength range . . . . .	28
4.4 New fibre . . . . .	29
<b>5 Programming</b>	<b>31</b>
5.1 python example . . . . .	31

<b>A Specifications</b>	<b>35</b>
<b>B Command language</b>	<b>39</b>
B.1 Wavelength measurement . . . . .	39
B.2 General functions . . . . .	40
B.3 Display functions . . . . .	40
B.4 Auxilliary measurements . . . . .	41
B.5 Imaging sensor . . . . .	42
B.6 Configuration settings . . . . .	44
B.7 EEPROM access . . . . .	47
<b>C mogwave configuration</b>	<b>49</b>
C.1 Command-line options . . . . .	49
C.2 Configuration file . . . . .	51
<b>D mogwave server</b>	<b>53</b>
D.1 Enabling the server . . . . .	53
D.2 Communication with the server . . . . .	53
D.3 mogwave commands . . . . .	54
<b>E Ugrading firmware</b>	<b>57</b>
<b>References</b>	<b>58</b>

# 1. Introduction

The MOGLabs MWM optical instrument is a high resolution diffraction grating spectrometer in Littrow configuration. Optical interference creates an image of a singlemode fibre core at a wavelength-dependent position on a semiconductor imaging sensor. The image is digitised at high spatial resolution and with high dynamic range and speed. The wavelength spectrum is determined from the digitised image and a fitting procedure determines the peak wavelength. It provides picometre (0.001 nm) accuracy and picowatt sensitivity, with up to 100 readings per second. The relative accuracy, for example in measuring the difference between two laser wavelengths, is around 10 MHz (0.01pm).

Some basic device parameters and wavelength can be displayed on the instrument (see figure 1.1), or communicated to a host computer via USB or ethernet (TCP/IP). Computer interface is very straightforward using simple text-based commands.



moglabs  
w2001101140930 v0.3.15  
766 nm to 789 nm  
  
31.8 C 1006 hPa 0.06 ms  
IP 10.1.1.253:7802

**Figure 1.1:** Wavemeter on-device display, showing serial number, firmware version, wavelength range, temperature, pressure, exposure duration and IP address.

## 1.1 Basic setup

1. Connect a 5V 500mA power supply via the USB interface. Some computer USB ports will provide sufficient power but in general a dedicated power supply or USB hub is recommended.
2. Second-generation (Rev 2) instruments have power on/off via the OK

pushbutton. To power on or off, hold the OK button down for 2 to 3 seconds.

3. If possible, use a network connection for improved response times and faster data acquisition. Connect to an ethernet switch. The MWM will determine an IP address via DHCP. If that fails, it will default to an internal IP which will be shown on the instrument display.
4. The MWM must be coupled to the light source via a single-mode optical fibre. Multimode fibre is *not* suitable. The instrument sensitivity is very high and so the fibre mode diameter need not be well matched; for example, a 480 nm fibre can be used to measure 780 nm light and vice versa, provided the fibres are not excessively long. Fibres with FC/APC or FC/PC connectors can be used, and standard or polarisation-maintaining fibre.

**Note:** Your device has been configured and calibrated using a fibre with either FC/PC or FC/APC connector, usually with FC/PC at the wavemeter. Please refer to the test data provided with your device. If you use a different connector type, you will need to reconfigure and recalibrate. Please see section 3.8.

**Caution:** The fibre tip is an integral part of the instrument. Unclean or damaged fibre tips will give unsatisfactory results.

5. The target laser should be directed into the other end of the fibre patch cord. Note that a lens is not needed; the bare fibre will normally couple sufficient light. A few nanowatts is good, though when first setting up, a few microwatts will make life much easier.
6. Start the `mogwave` software application on a suitable computer, as described in the next chapter.



## 1.2 Front panel display and buttons

The push-buttons provide limited functionality described below. Future upgrades will allow full menu control of essential functions such that the device can be configured and used standalone.

← Activate firmware upgrade if held down during startup.

↑ Load network configuration from hard-coded definitions instead of EEPROM, if held down during startup.

↓ Enable device-based measurement with on-screen display.

OK + ↓ Hold both down to reset.

OK Rev 2: power on/off; hold down for 2 to 3 seconds to power on or off.

## 1.3 Software installation

The MWM can be used directly from many standard laboratory data acquisition systems such as `python`, `matlab` or `LabVIEW`, via simple text commands. A simple example is included in Chapter 5. We also provide a standalone GUI application, `mogwave`, which runs under Microsoft Windows™.

To install the `mogwave` application, run the appropriate installer, for example `mogwave_setup.exe`.

### 1.3.1 USB drivers

When you connect the MWM, your computer should automatically find and install USB drivers from the Windows™ update service. If not, in the install directory (typically `c:\Program Files\MOGLabs`) you should find the `USB_VCPdriver` folder which will contain setup programs for 32-bit and 64-bit systems.

If you have problems connecting with USB, in particular no driver found:

1. Ensure the device is connected to your computer via USB cable.
2. Double-click the appropriate 32-bit or 64-bit driver from the `USB_VCPdriver` folder in the install directory, typically `c:\Program Files\MOGLabs`.
3. Once installation is completed, restart your computer.
4. When your computer is finished booting, disconnect the USB cable from your wavemeter, and then reconnect.

### 1.3.2 Starting `mogwave`

Once the software is installed, run `mogwave` from the Start menu or desktop shortcut. If you are connecting by TCP/IP, and have only one MWM in your subnet, then `mogwave` should automatically find the device.

If using USB, or you have multiple MWM devices, or the network is a little slow, and the application does not find your device, you can either edit the shortcut link which was created during the install process, to provide command-line parameters that specify the device address, or edit the `default.cfg` file. Further details are provided in the `mogwave` chapter 3, or in appendix C.

## 2. Communications

The MWM can be connected by USB or ethernet (TCP/IP). From the user and device perspective, the two are equivalent.

### 2.1 TCP/IP

When ethernet is connected, the MWM will attempt to obtain an IP address by DHCP. If DHCP fails, an internally defined address will be used. In both cases, the address will be shown on the device display, something like:

```
10.1.1.180:7802
```

The dotted-quad number (10.1.1.180) is the device IP address; you can type that into the address bar on your web browser and you will see a simple webpage for the device (via the standard `http` port 80).

The second number (7802) is the TCP/IP port number. This port provides full access to device functionality. It is through this port that your control software should access the device. Simple commands can be sent to the device using the `telnet` protocol, for example:

```
telnet 10.1.1.180 7802
disp,rgb,100,100,100
```

Here the `telnet` command line specifies IP address 10.1.1.180 and port 7802. You can also use a GUI `telnet`, in particular on Windows<sup>TM</sup> we recommend PuTTY available at [www.putty.org](http://www.putty.org). Specify the Host Name as 10.1.1.180, Connection type as Telnet, and the Port which is normally 22, should be changed to 7802.

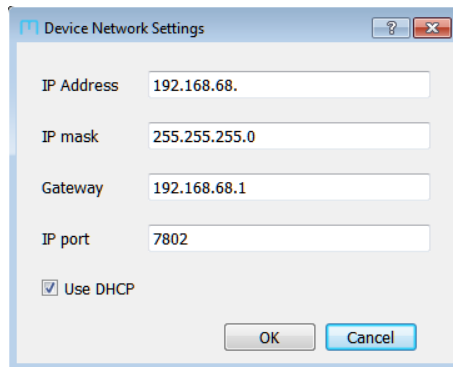
The above example sends the `disp,rgb` command to the device (to set the display backlight intensity). The full suite of commands is described in Appendix B.

You can also access the device indirectly through a socket server established by `mogwave` on your host computer. The `mogwave` server provides fully calibrated wavelength measurement, allowing `mogwave` to act as an API for data acquisition systems, for example `matlab` or `LabVIEW`. See appendix D for more information.

### 2.1.1 Changing IP address

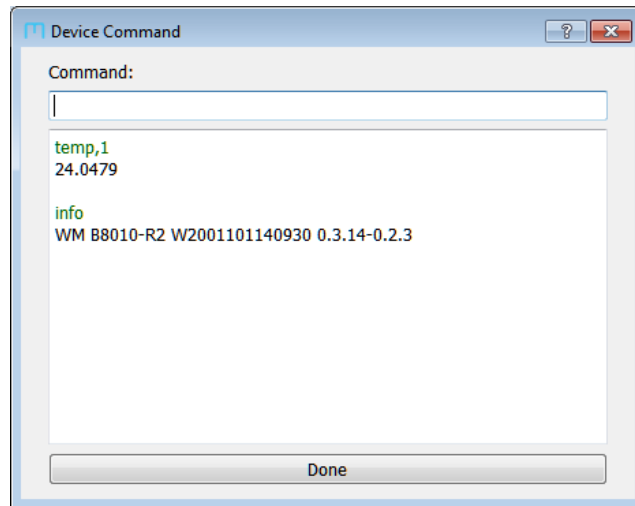
If you do not have a DHCP server to define the device IP address, and the default is not suitable, you will need to manually change the IP address. This is most easily done by connecting via USB, and using the `mogwave` application to configure the device IP settings. The `mogwave` application is described in further detail in the following chapter, but two specific functions are particularly useful for establishing communications.

**Device network settings** To configure the device network connection settings (IP address, mask, gateway and port), particularly useful if the device is connected by USB.



**Device command** This function (`File` menu, `Device command`) opens a new dialog for interactive command access to the device, similar to `telnet` but this approach will work via USB. Type a command into the command line window, and hit `Enter` to send that command to the device (see appendix B); the response will appear in the box below the command line window.

Some commands are not permitted because they require complex data flow.



### 2.1.2 Configuration in alternate subnet

Another approach is to plug the device into a network which is in the same address space, then send the appropriate commands by `telnet`, and reconnect in the desired address space. For example, if the MWM default address is 10.1.1.180, and you want the device to work in the 192.168.1.xxx space:

1. Connect the ethernet directly to a computer ethernet socket, not via a network switch or hub.
2. Configure the computer network settings to have an address in the same address space as the MWM; for this example, you can set the computer IP address to 10.1.1.189.
3. Telnet to the device on address 10.1.1.180 and port 7802, for example:

```
telnet 10.1.1.180 7802
set,ipaddr,"192.168.1.100"
set,ipgw,"192.168.1.1"
reset
```

After reset, the device will default to the new settings. If using PuTTY ([www.putty.org](http://www.putty.org)) as a telnet client, set the Host Name to 10.1.1.180, Connection type to Telnet, and the Port which is normally 22, should be changed to 7802.

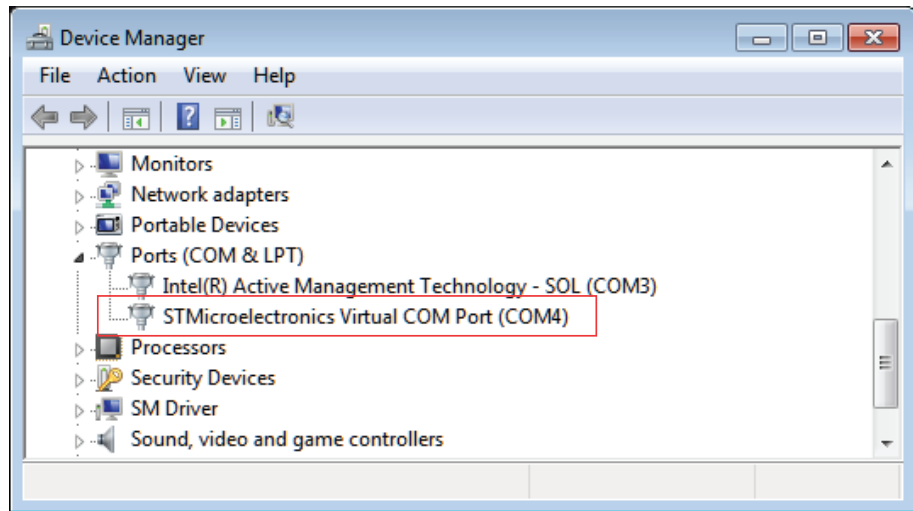
## 2.2 USB

The MWM will appear as a VCP: a virtual COM port; that is, a fast serial port like an RS232 connection. When the device is connected to a Windows™ computer, the computer will normally automatically download and install the STM32 Virtual COM Port Driver. Your software can then communicate as with any normal COM port.

To determine which port number is required, go to Device Manager (Start, then type Device Manager into the Search box). You should see a list of devices including “Ports”, and within that,

```
STMicroelectronics Virtual COM Port (COMxx)
```

where xx is a number, typically 4 or 15 (see figure below), the USB port number.



### 2.2.1 USB driver

If you do not see a virtual COM port under “Ports” in the Device Manager then manually install the USB driver. The driver install package should be in `c:\Program Files (x86)\MOGLabs\USB_VCPdriver`.

1. Run the USB driver setup program. This will copy the drivers to your computer, typically in

`C:\Program Files (x86)\STMicroelectronics\Software\Virtual comport driver`

Currently there are folders `Win7` with drivers for Windows 7 and `Win8` with drivers for Windows 8 to 10.

2. Ensure the wavemeter is connected to your computer via USB cable.
3. Double-click the appropriate 32-bit (`dpinst_x86.exe`) or 64-bit (`dpinst_amd64.exe`) driver for your version of Windows.
4. Once installation is completed, restart your computer.
5. When your computer is finished booting, disconnect the USB cable from your wavemeter, and then reconnect.

Look again for the virtual COM port within Device Manager. If you see STMicroelectronics COM Port (COMxx) and the device is not connecting then you need to tell the software (mogwave) which port to use. This is most easily done by editing the default configuration file, C:\Program Files (x86)\MOGLabs\mogwave\configs\default.cfg. Change the two lines

```
# usb_port = COM15
# usb = False
```

to

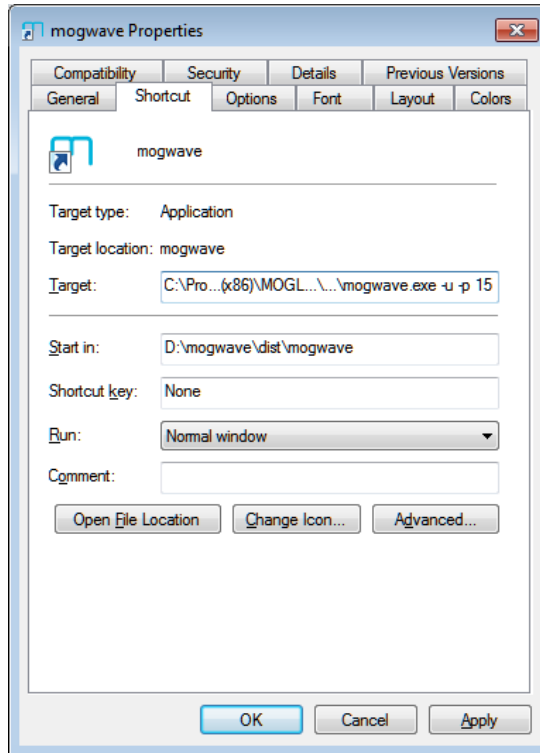
```
usb_port = COMxx
usb = True
```

where xx is the COM port shown in Device Manager.

You can also specify USB and the port via command-line options for mogwave. Edit the shortcut link to add `-u` to force USB, and `-p xx` to specify the port number.

1. Right-click on the mogwave desktop shortcut and select Properties.
2. Edit the Target to add `-u -p xx` at the end (see figure below). It should be something like  
C:\Program Files (x86)\MOGLabs\mogwave\mogwave.exe -u -p 15.
3. Select Apply to save the new shortcut.





**Figure 2.1:** mogwave shortcut properties, edited to add USB port number. Note that the Target has been abbreviated here.



# 3. MOGWAVE

The host software `mogwave` provides a graphical user interface that allows measurement and calibration functions. See appendix C for information on starting `mogwave` and configuring default parameters.

## 3.1 Device discovery

By default, a device discoverer (Figure 3.1) is initiated when `mogwave` is started. The discoverer scans the USB ports of the host computer looking for an MWM device, and then scans the local network subnet. Starting the application is then as simple as selecting the device to communicate with and clicking *Connect*.

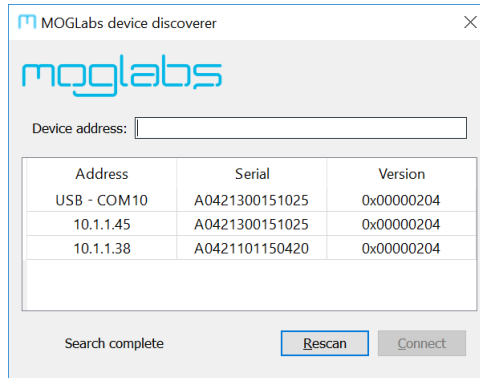


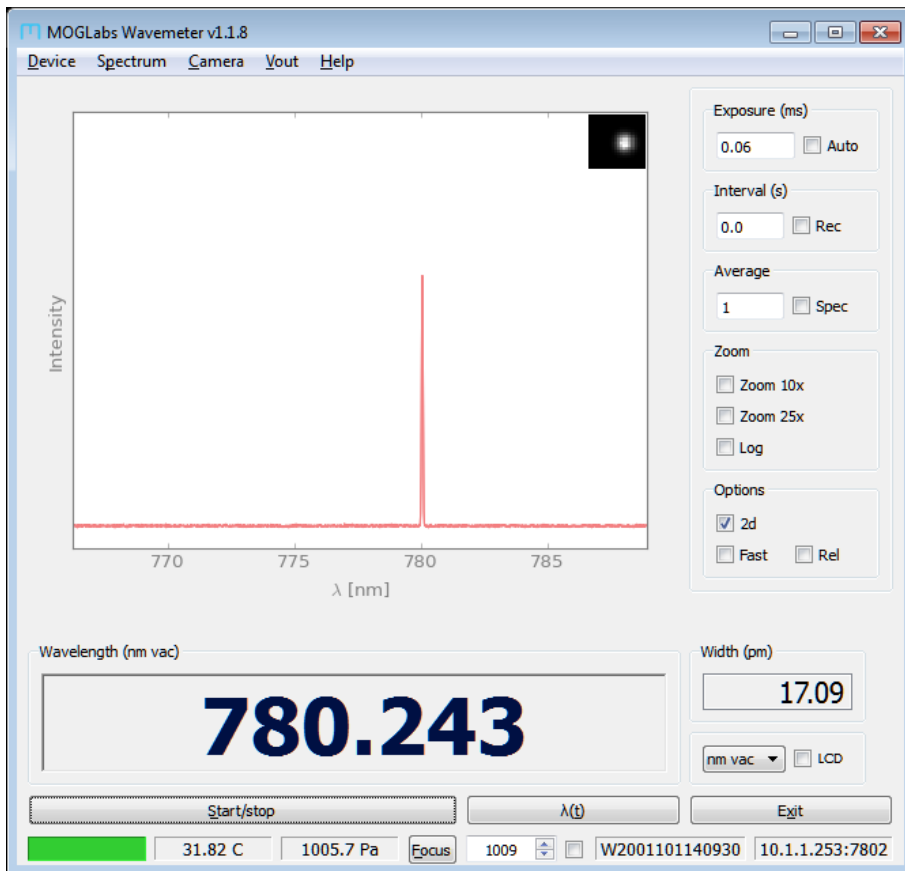
Figure 3.1: Example of the *device discoverer* window.

If the network does not permit device discovery and/or your MWM does not appear in the list, it is possible to enter the IP address of the unit in the *Device address* box and connect regardless.

The discoverer will not start if the connection is specified in the default configuration file or on the command line or equivalent shortcut link (see appendix C).

## 3.2 Main display

The primary window of `mogwave` is shown below. Note that the window size can be altered dynamically by simply stretching from a corner. To adjust the font size for small or large displays, change the `wmfontsize` parameter in the configuration file (appendix C.2).



**Spectrum** The largest area shows the most recently acquired spectrum, or an average of a set number of spectra (see below). The display can be zoomed into an area around the highest peak using 10x and 25x zoom functions, and the spectrum can be shown in linear or log form.

**2d image inset** The inset shows a 2d image of the area around the highest spectral peak, which should be approximately Gaussian and symmetric. If the laser is multi-mode, the main peak will become distorted or even show two distinct spots.

**Wavelength** The calculated wavelength of the highest peak is shown. The wavelength is measured in ambient conditions (air temperature and pressure) and converted to vacuum equivalent wavelength, frequency or wavenumber by selecting the appropriate units. The width of the highest peak is also shown, or (by clicking on the number) the statistical uncertainty of the wavelength measurement.

**Status bar** The status bar shows temperature and pressure inside the device, the device serial number, and the communications channel: IP address and port (if using ethernet) or USB port. Within the status bar, there is a coloured indicator region which can be red, yellow or green, a Focus button, a row number, and an auto-row toggle; these are described in greater detail below.

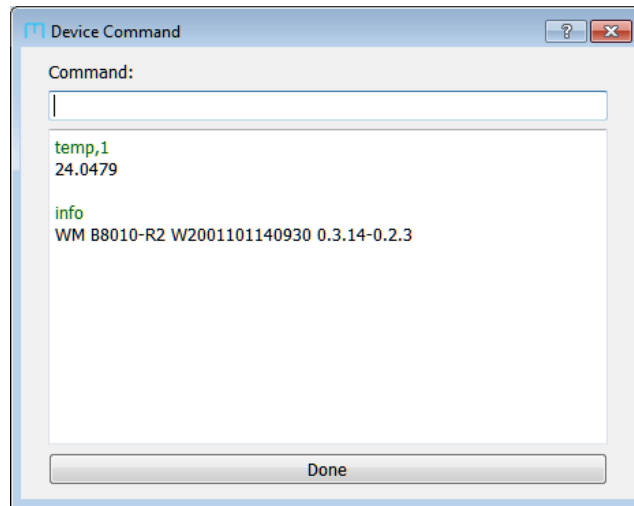
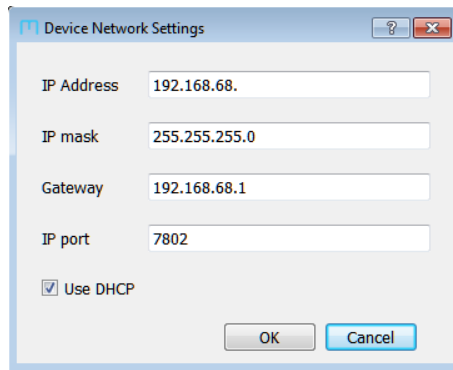
## 3.3 Menu

### 3.3.1 Device

**Device network settings** To configure the wavemeter network connection settings (IP address, mask, gateway and port). Particularly useful for configuring the network settings via USB.

**Device command** This opens a new dialog for interactive command access to the device, similar to `telnet` but this approach will work via USB. Type a command into the command line window, and hit **Enter** to send that command to the device (see appendix B); the response will appear in the box below the command line window. Some commands are not permitted (for example `spectrum`) due to the large amount of response data generated, particularly when communicating via USB.

**Console** Opens a new window with `mogwave` console text output; that is, the text that is normally displayed in the console if `mogwave` is started from a console window such as a Windows<sup>TM</sup> `cmd` window.



**Error log** Opens a new window for mogwave error messages; that is, the text that is output when software errors occur.

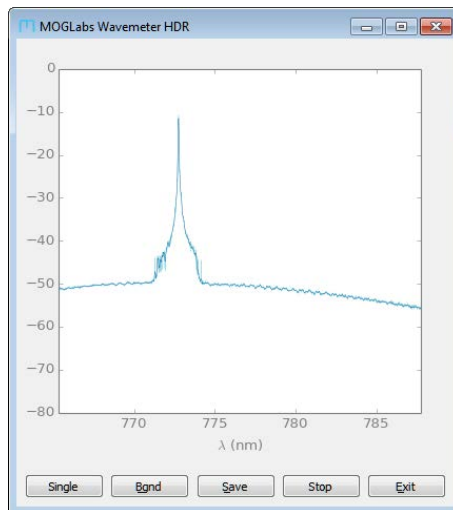
### 3.3.2 Spectrum

**Save spectrum** The current spectrum can be saved to a tab-separated .dat file using the File menu at the top of the window.

**Save screenshot** Saves currently displayed spectrum as a .pdf file.

**HDR spectrum** Experimental. Opens a new window (see below) for acquiring high dynamic range (HDR) spectra. A sequence of spectra are obtained at increasing exposure times and combined to form a single high dynamic range spectrum. The result is qualitative, but can provide information on spectral purity and ASE (amplified spontaneous emission).

Background images must be obtained before generating HDR spectra. The first time an HDR spectrum is acquired, a dialogue box will pop up asking you to block the light into the fibre, and to press OK. A sequence of background measurements will be acquired and stored.



**Single** Generates a single HDR spectrum. If background data has not been obtained, you will be prompted to block the light into the fibre so that the background noise can be determined.

**Bgnd** Acquire new background measurements.

**Save** Save current HDR spectrum. Saves all separate exposure time spectra and background measurements, with calculated HDR spectrum and a smoothed spectrum.

**Go/Stop** To generate HDR spectra in real time. The main application must be actively collecting spectra (if not, press the Start/Stop button in the main

application). The spectrum and background images each require about 1 second of collection time so the main application response rate will be severely reduced.

**Exit** Close the HDR window.

**Calibrate** For device calibration; see following chapter.

### 3.3.3 Camera

**Camera gain adjust** Select electronic gain on imaging sensor (1x, 2x, 4x, 8x, 16x). Allows faster measurement with extremely weak signals. For best results use higher power to keep exposure times below 1 ms at 1x gain.

**Acquire full image** Captures a full image from the sensor. Due to limited memory on the device, the image is captured in bands. Acquisition will take about 2 minutes using USB and five or six seconds on fast (100 Mb/s) ethernet.

### 3.3.4 Analogue out

**V<sub>out</sub>** The MWM can output an analogue signal on the SMA connector, either a fixed value or a signal that depends on the wavelength relative to some set wavelength. The dialogue box allows configuration of the analogue output signal.

In *Error out* mode, the output voltage is

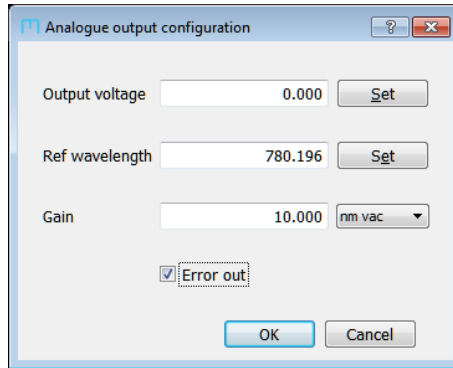
$$V_{\text{out}} = G(\lambda - \lambda_0)$$

limited to  $[-2.5.. +2.5]$ V, where  $\lambda$  is the most recently measured wavelength,  $\lambda_0$  is the set reference wavelength, and  $G$  is the gain.

**Output voltage** Set a fixed value, limited to  $-2.5$ V and  $+2.5$ V. The value will remain until changed or until error signal output is enabled.

**Reference  $\lambda_0$**  Defines the reference wavelength  $\lambda_0$ . Click on the *Set* button to use the currently measured wavelength.





**Gain** Defines the gain,  $G$ . The gain is in volts per unit, where the drop-down unit selector defines the units. The gain and output voltage do not change if the current *measurement* units are changed.

Changing the units will change the displayed reference wavelength and gain so that both are correct for the units shown. These units are independent of the *measurement* units in the main *mogwave* window.

**Error out** When checked, the output voltage will be changed every time a new wavelength measurement is made.

### 3.4 Acquisition

**Exposure** Sets the exposure time, in milliseconds. If the Auto button is pressed, the software will attempt to find an optimum time. The maximum time is currently 400 ms.

**Interval** Defines a dead-time between measurements. Can be anything up to 15 s. If the Rec box is checked, measurements are recorded in a tab-separated file until the Rec is unchecked. A new file is created each time recording is started. The filename is generated automatically based on the current time.

**Average** The displayed numerical wavelength is the average of the most recent  $n$  measurements, where  $n$  is the number in the Average field.

If the Spec box is checked, then instead  $n$  spectra are averaged, and the wavelength calculated from the spectrum average.

**Zoom** The 10x and 25x reduce the wavelength range, to show greater detail of the peak and increase the measurement rate. The wavelength range is determined automatically to be centred on the spectral peak.

The Log button changes the display to show  $20 \log_{10}$  of the measurements, relative to 0 dB at the saturated pixel value.

### 3.5 Options

**2d** Select to operate in high accuracy 2d mode. Unselect to operate in faster 1d mode.

**Fast** Select to operate without showing small 2d images of diffraction spots.

**Rel** When selected, the current wavelength is saved as a reference. The reference is subtracted from future measurements so the displayed wavelength shows the change relative to the reference value.

### 3.6 Wavelength

**Wavelength** The primary wavelength measurement is shown in a variable size (stretchable) text window. The last digit is slightly smaller and in grey text as a reminder that its value is imprecise. Click on the number to cycle through the different units.

**(nm, vac)** The in-vacuum wavelength, converted from the measured value using measurements of the air pressure and temperature.

**(nm, air)** The in-air wavelength, converted from the measured value using corrections for pressure or temperature.

**(nm, raw)** The raw in-air wavelength, as measured by the MWM, without corrections for pressure or temperature.

**(THz)** The frequency, converted from the measured in-air wavelength, using measurements of the air pressure and temperature.

**(cm<sup>-1</sup>)** The wavenumber; that is, the number of wavelengths per cm, in vacuum, converted from the measured in-air wavelength using measurements of the air pressure and temperature.

**Uncertainty/Width** Shows either the statistical uncertainty of the wavelength measurement, or the peak width. Click on the number to flip from one to the other. The uncertainty is just a guide to indicate the reliability of the measurement and should not be used for quantitative purposes.

**LCD** If the LCD toggle is on, the measured wavelength will be displayed on the device LCD screen, in current units. This can be helpful for tuning a laser, if the MWM is close to the laser and far from the host computer.

**wave\_mult** The `wave_mult` parameter defined in the config file can be used to multiply the measured wavelength by a constant, for example if you are working with a frequency doubled or parametric downconverted source. Thus you might measure the wavelength of a 1064 nm laser but your experiment uses the 532 nm doubled output, so `wave_mult` would be set to 0.5.

### 3.7 Functions

**Start/stop** Starts and stops acquisition and display of spectra and calculated wavelength.

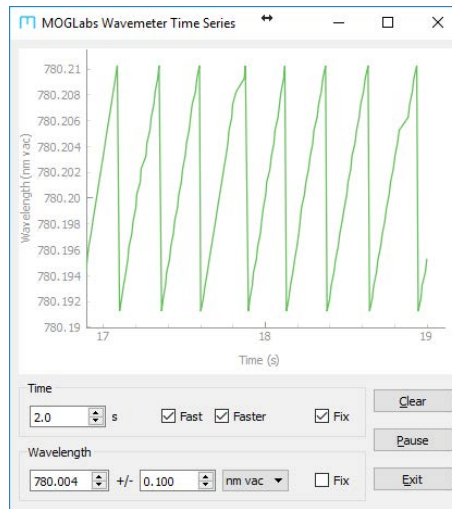
**Exit** To quit `mogwave`.

**$\lambda(t)$**  Opens *Time Series*, a new graphing window, which displays the wavelength measurement as a function of time.

**Time** Configures the horizontal (time) axis range. The range will dynamically update to show all data acquired from when the Time Series window is opened, or the most recent measurements for a fixed duration if `Fix` is selected.

**Fix** If selected, show only most recent measurements, defined by value in Time spinbox.

**Fast** When selected, the main spectrum is only updated once per second, thus allowing faster update of the Time Series plot.



**Faster** Select device-based wavelength measurements, indicated by cyan coloured status bar. Device-based measurements are much faster, allowing much more rapid updates of the Time Series plot. Also reduces the rate of update of the main wavelength numeric display. *Fast* and *Faster* can both be selected for fastest possible Time Series update rate.

Device-based measurements are less accurate than can be achieved with full *mogwave* analysis, particularly when the default in-device imaging row is different to the optimum row determined by *mogwave*.

**Wavelength** The vertical (measurement) axis can be configured to display in any of the allowed units. The range will dynamically update to show all measured data, or the axis can be fixed.

**Clear** Clears all measured data.

**Pause** Suspend plotting of measurements. Note that measurements can continue in the main *mogwave* window.

**Exit** Close the Time Series window.

### 3.8 Status bar

The status bar shows secondary information including the measured air device temperature and air pressure, the serial number, and the communications channel including IP address and port number if communicating by TCP/IP, or the USB port number. There is also the ability to change sensor row used for acquisition of spectra.

**Status** A colour indicator shows green when the main peak is properly acquired, red otherwise, indicating the displayed wavelength is not valid.

**Focus** The focus button will open a new window showing the sensor area around the current peak (typically 160 x 160 pixels), to allow convenient optimisation of the focus (see calibration in the next chapter). It can also be useful for finding the optimum pixel row (see below). The row number should be adjusted until the laser spot is at the centre of the focus window.

**Ctrl-focus** Hold the Ctrl key down, and click on Focus, to open a very wide but not very high focus window. This is useful for adjusting the diffractive optical element for different wavelengths (see calibration chapter).

**Row** The imaging sensor has many pixel rows. At the factory, the MWM is aligned and focussed and the optimal sensor row is determined and stored on the device.

The row can change, due to thermal drift or if the fibre is changed, in particular if changing from FC/PC to FC/APC or vice versa. The row is tracked automatically but the row must initially be close, to within about 10 pixels of the centre of the spectral peak.

The row can be set via the up/down arrows or typed in numerically.

**Auto-row** The automatic tracking of the row for the main spectral peak can be enabled or disabled via the check-box.



# 4. Calibration

Recalibration of the MWM can be required due to drift and metal creep, accidental mechanical shock, or to change the operating wavelength. Usually it is a very simple process of measuring a known laser wavelength and clicking one button. Multi-point calibration across the operating wavelength range will improve the accuracy.

## 4.1 Existing wavelength range

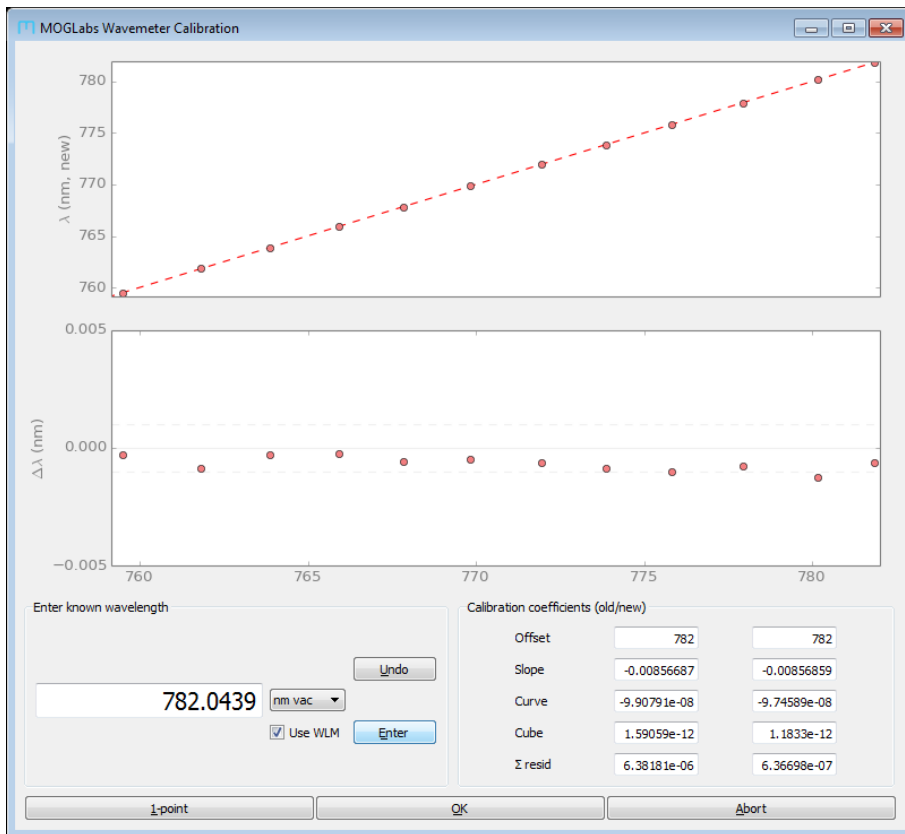
Here we describe recalibration, if the operating wavelength is unchanged.

From the `mogwave` application, click on the `Calibrate` button to start the calibration application. The normal wavelength measurement program will continue to run, so that you can see the measured wavelength using the current calibration data.

The calibration process requires measuring the wavelength of a light source, and entering the known wavelength for that source. Before entering data, please make sure that the units are the same in both the normal and `Calibrate` windows, typically (`nm`, `vac`). Avoid using (`nm`, `air`) because the measurements are subject to arbitrary variations in pressure and temperature.

The `Calibration` window shows two graphs. The upper graph shows the entered (known) wavelength against spectrum channel or bin; effectively the uncalibrated wavelength. The lower graph shows the difference between the entered (known) wavelength and the calculated wavelength using the old calibration.

To calibrate, simply enter the known wavelength for your light source. Repeat at as many different wavelengths as desired, across the measurement range of the device. The sum of the residuals can provide an indication of whether the new calibration coefficients are more precise than the old. If an error is made in entering a calibration point, the `Undo` button can



be used to remove it. Repeated use of **Undo** can remove all entered data points, in reverse order.

**NOTE** It is important that the calibration wavelengths extend over a significant fraction of the operating range of the wavemeter. If you attempt to calibrate to several closely-spaced wavelengths, the calibration will be completely erroneous. If you cannot tune your calibration source across a large range, use single-point calibration only. You can recover the factory calibration; please contact MOGLabs for assistance.

If doing a single point measurement, click on the **1-point** button. The



offset will be corrected, and a new slope will be calculated based on the new wavelength.

If more than four calibration points have been entered, then press the OK button to calculate the full calibration function.

Both single and multipoint calibrations will save the new calibration coefficients on the device. They also save all calibration points (measured and known wavelengths) to a .csv file.

To avoid reprogramming the device with new calibration data, hit the Abort button.

## 4.2 Calibrating with HighFinesse wavemeter

If you have a HighFinesse wavemeter, you can automatically load the known wavelength from the HighFinesse measurements in real time, making calibration very fast.

1. Use the `-w` command-line option, or edit the config file (see appendix C) options for the `[wavemeter]` section:

```
[wavemeter]
useWLM = True
```

2. Start the HighFinesse application and ensure that it is successfully measuring your laser wavelength.

Now when you run `Calibrate` from `mogwave`, the known wavelength will automatically be copied from the HighFinesse application.

### 4.2.1 Multiple HighFinesse wavemeters

HighFinesse wavemeters require a separate installation of device-specific software for each different HighFinesse device. If you have installed more than one on your computer, the default `wlmData.dll` may not be the

correct version for the device you are trying to use. In that case, copy the correct `wlmData.dll` file to the `mogwave` installation.

The `wlmData.dll` file is typically in the `Com-Test` folder of the HighFinesse application for the device you will be using, for example

```
C:\Program Files (x86)\
  HighFinesse\Wavelength Meter WS6 1234\Com-Test\wlmData.dll.
```

Copy this to `C:\Program Files (x86)\MOGLabs\mogwave`.

Unfortunately HighFinesse have changed the way they store and load their own `wlmData.dll`, and the above may not work for your system. Contact MOGLabs if you are having difficulty.

### 4.3 New wavelength range

Reconfiguring the device to operate at a new wavelength, outside the range of the imaging sensor, is a quick and simple process.

1. Remove the top plastic cover, and the bottom aluminium cover, of the device.
2. Loosen the  $\lambda$  screw holding the diffractive optical element (see below), and for Rev 2 devices, the three locking screws holding the element in place at the bottom of the device, inside the optical chamber.
3. Couple light into the MWM, preferably with plenty of power (a few microwatts) output from the fibre endface (at the MWM end).
4. Run the `mogwave` program, and rotate the diffractive optical element (see below) until a spectral peak is seen. On Rev2 devices, the indicator line on the diffractive element holder should point towards the fibre such that the surface of the diffractive element is exposed to the light.

To find the peak as the element is rotated, it can be helpful to use the `Ctrl-Focus` feature to see the full width of the sensor but some

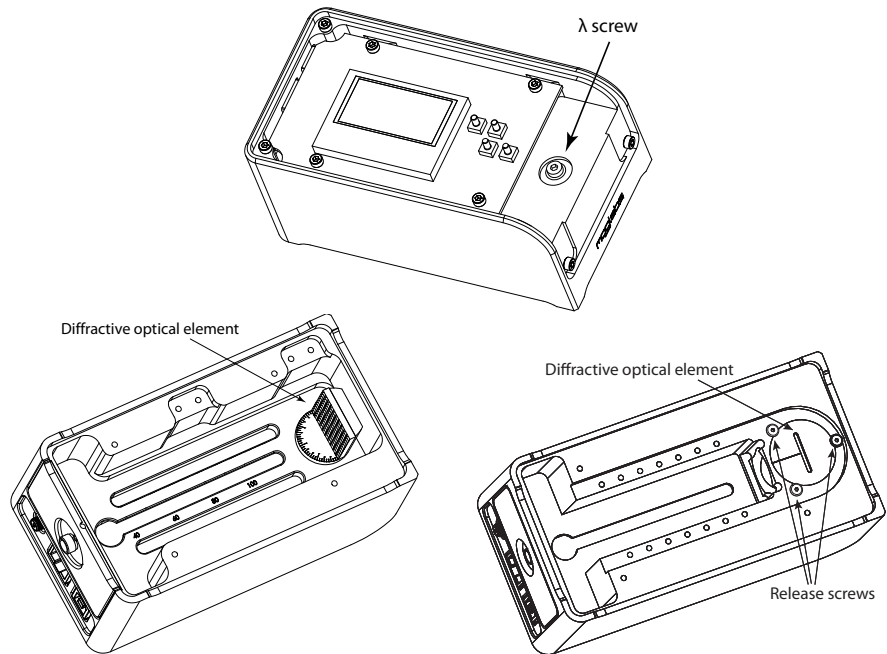


Figure 4.1: Adjustment of wavelength range; Rev 1 device at left, Rev 2 at right.

adjustment of the active pixel row may be needed.

5. Firmly retighten the screws holding the optical element in place and replace covers.
6. Calibrate as above.

#### 4.4 New fibre

The wavemeter relies on the precise location of the fibre core. Some fibre patchcords (e.g. Diamond Fibre Optics) have highly repeatable fibre core locations, but most require some correction. Also changing between PC and APC terminations requires correction. The corrections are determined by taking a full sensor image and finding the appropriate diffraction spots.

Ensure light is diffracted onto the sensor. Acquire a Full Image, with

exposure time of 10 ms or longer. You may need to run `mogwave` as Administrator to save the image file in the MOGLabs folder. Open the `fullimageYYMMDD_hhmmss.tiff` file, normally saved in the folder in which `mogwave` is installed, with an image viewer (e.g. `IrfanView` or `ImageJ`). You should see one bright spot, and may also see one or two additional secondary spots depending on which generation of wavemeter you have (contact MOGLabs if unsure). Find the `x,y` location of the main spot; for example, in `IrfanView`, click on the spot, and while holding the mouse button down, at the top left of the `IrfanView` window you will see the `x,y` location.

The location should be programmed into the device. Using `telnet` or the `mogwave` Device Command option, execute the following command:

```
set,row,mainY
```

where `mainY` is the `y` location of the main spot.

Reset the device, then calibrate as above for new wavelength range.

# 5. Programming

The MWM is controlled via simple text commands of the form:

```
DISPLAY,LINE,0,0,"Hello"
```

Each command begins with a text key; only the first three letters are required but the full command can be used. Some commands will have a sub-command, and/or parameters including integers, floating-point numbers, and text strings. Each command will return a requested value, or a default "OK".

These commands can be sent to the device via TCP/IP (for example, using `telnet`) or USB. A full list of commands is provided in Appendix B.

## Watchdog timer

The MWM has a watchdog timer that resets the device if any internal function or operation fails to complete within approximately 5 seconds. Thus if your application fails to receive an expected response then it can attempt to re-establish a connection after that time.

### 5.1 python example

A simple program involves opening the communications channel, configuring the imaging sensor, optimising the exposure time, and then either acquiring the spectrum or the calculated wavelength. An example, written in `python`, is outlined below. Further examples will be provided for `matlab` and `LabVIEW`.

```
#-----  
# MWM wavemeter python example  
# (c) MOGLabs 2014  
#-----
```

```

import matplotlib.pyplot as plt
ipaddr = '10.1.1.23'
ipport=7802
usbport='COM4'

#-----
# Establish communications channel
#-----
usb=True
usb=False # prefer TCP/IP; set this True to use USB

if not usb:
    import socket
    print "Connecting to wavemeter at IP:port=",ipaddr,":",
        ipport
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    try:
        s.connect((ipaddr,ipport))
        # set non-blocking so send/recv can return
        # without having done anything
        s.setblocking(0)
        s.settimeout(3) # timeout if frozen more than 3 seconds
    except:
        print 'Failed to connect to ',ipaddr,ipport
        sys.exit()
else: # must be USB
    import serial # Serial comms (USB Virtual COM Port)
    print "Connecting to wavemeter via USB port %s" % usbport
    # establish serial connection
    # timeout must be longer than maximum exposure time
    s=serial.Serial(usbport,1,timeout=1.5)
    s.send=s.write # redefine send and recv to use usb
    s.recv=s.readline
    s.flushInput()

#-----
# Demonstrate simple command
#-----
s.send("info\n")
print "Connected to "+s.recv(20).strip()

#-----

```

```

# Initialise sensor
#-----
s.send("get,row\n")
row=int(s.recv(20),16)
# define fov: two rows 2592 pixels each
rows=2
cols=2592
cmd="cam,fov,%d,0,1,%d\n" % (row,cols-1)
s.send(cmd)
ok=s.recv(80)
bufsize=cols*2 # two bytes per pixel
maxy=2*4096 # 12-bit per pixel, two rows summed

#-----
# Find optimum exposure time
# cam,<time,<max> finds optimum but with
# maximum of <max> milliseconds
#-----
cmd="cam,atime,400\n"
s.send(cmd)
shutterStr=s.recv(80)
shutterTime=float(shutterStr)
print "Exposure_time:_%%.2fms" % shutterTime

if (400-shutterTime < 1):
    print "Insufficient_input_power"
    sys.exit()

#-----
# Acquire spectrum
#-----
buflen=0
bindata=''
s.send('spe\n')
if (usb):
    bindata=s.read(bufsize)
else: # deal with fragmented tcp/ip packets
    while buflen<bufsize:
        chunk=s.recv(bufsize-buflen)
        chlen=len(chunk)
        bindata+=chunk
        buflen+=chlen

```

```
image=[]
# convert to list
for i in xrange(0,bufsize/2):
    word=ord(bindata[2*i])+(ord(bindata[2*i+1])<<8)
    image.append(word)
y=image[0:cols]

#-----
# Plot spectrum
#-----
plt.clf()
plt.ylim(0,maxy)
plt.xlim(0,cols-1)
plt.plot(y,'-')
plt.show()

#-----
# Report measured wavelength
#-----
s.send('wavelength\n')
print "Wavelength:␣",s.recv(20).strip()

# expose and measure again
s.send('cam,exp\n')
s.recv(20)
s.send('wav\n')
print "Wavelength:␣",s.recv(20).strip()
s.close()
```



# A. Specifications

Parameter	Specification
-----------	---------------

Wavelength	
Wavelength	350 nm – 1100 nm
Range	23 nm at 780 nm
Min input power	1 pW at exit of singlemode fibre
Accuracy	$\pm 1$ GHz ( $\pm 0.001$ nm) at 780 nm (at time of calibration)
Precision	100 MHz (0.1 pm) at 780 nm

Electronics	
Power	+5 Vdc 350 mA via USB port
Display	In-device LCD or host computer
Battery	External via USB
PID feedback	12-bit DAC output, $\pm 2.5$ V (1 mV resolution)
Exposure time	60 $\mu$ s to 20 s
Readout rate	Up to 100 Hz, dependent on user computer

Sensors	
Imaging	Up to 2750 $\times$ 2004 2.2 $\mu$ m $\times$ 2.2 $\mu$ m pixels
Spectral response	See below
Temperature	Resolution $\pm 0.008^\circ\text{C}$ Range 10 $^\circ$ to +40 $^\circ\text{C}$

Parameter	Specification
-----------	---------------

Pressure	Resolution $\pm 1$ Pa Range 260 hPa – 1260 hPa
Aux ADC input	0 to +5 V, 12-bit (1 mV)

Interface	
Ethernet	10/100 TP RJ45
USB	USB1.1, plug type USB-B
SMA	12-bit DAC output, $\pm 2.5$ V (1 mV resolution)
Optical input	FC/APC or FC/PC single-mode fibre

Software	
mogwave	Windows-based GUI
API	Text commands via TCP/IP or USB python, Labview examples provided

Dimensions	
Dimensions	165 mm $\times$ 85 mm $\times$ 70 mm
Weight	1.5 kg

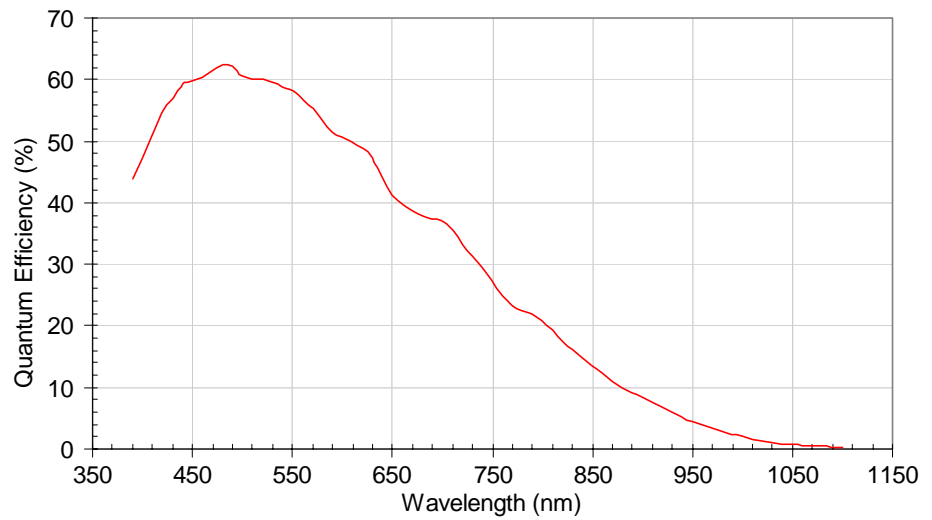


Figure A.1: Typical detector spectral response.



## B. Command language

The MWM is controlled via simple text commands of the form:

```
wavelength, vacuum
```

Each command begins with a text key. Only a few letters of the text key are required, just enough to uniquely identify the command against all other valid commands. Some commands will have a sub-command, and/or parameters including integers, hexadecimal integers (e.g. 0xf31b), floating-point numbers, and text strings. Each command will return a requested value, or a default “OK”.

### B.1 Wavelength measurement

**WAVELENGTH** `wave, raw`

```
wave, vac
```

```
wave, air
```

```
wave, thz
```

```
wave, num
```

Acquire image at current field of view and exposure time, and return calculated wavelength or frequency. The value is returned as a floating-point string. The different sub-commands return the *raw* (as measured) wavelength [nm], *vacuum* calculated wavelength [nm], *air* calculated (NTP) wavelength [nm], the frequency [THz], or wavenumber [ $\text{cm}^{-1}$ ].

Note that the device measures wavelength in air, for ambient temperature, pressure and humidity, given by the *raw* result. The *air* result is corrected for normal temperature and pressure (NTP, 20°C and 1 atm. 101.325 kPa). The raw measurement should not be used for quantitative determinations.

**UNITS** `units, raw`

```
units, vac
```

```
units, air
```

`units,thz`

`units,num`

Set display units for wavelength when device is operating standalone, initiated by pressing the ↓ (DOWN) button.

## B.2 General functions

**RESET** Reset, equivalent to power off and back on. Note that the communications link will be closed.

**CLOSE, QUIT, EXIT** Close communications channel.

**HELP, ?** List commands.

**INFO** Report device serial number.

**VERSION** Report firmware version.

**FRESET** Restore configuration from factory defaults.

**STATUS** Report status of the unit. Currently not supported.

**SER, STR** Conversion of serial numbers between string and integer.

**SERialtostr** `ser,u32`

Convert unsigned 32-bit code to serial number string.  
u32 must be a hexadecimal number.

**STRtoserial** `str,str`

Convert string to unsigned 32-bit serial number (8-digit hex).

## B.3 Display functions

**DISPLAY** Commands to control the on-device LCD display.

**line** `display,line,x,y,str`

Display string `str` at location `x,y`.  
 $0 \leq x \leq 16$ ;  $0 \leq y \leq 7$

**bigline** `display, bigline, x, y, str`

Display string `str` at location `x, y`, using large font.

$0 \leq x \leq 16$ ;  $0 \leq y \leq 7$

**blank** `display, blank`

Erase the display.

**clear** `display, clear, y`

Clear one line of text at `y`.

$0 \leq y \leq 7$

**contrast** `display, contrast, val`

Adjust display contrast.

$0 \leq val \leq 63$  (default is 14)

**rgb** `display, rgb, r, g, b`

Set backlight intensity for colours `r, g, b`.

$3 \leq r, g, b \leq 254$

Note that most models have white backlights, which respond only to the `r` value.

**dump** `display, dump`

Returns a binary dump of the LCD bit pattern.

## B.4 Auxilliary measurements

**ADC** `adc, chan`

Read voltage on ADC input, 0 to +5V unipolar, 12-bit.

Use `adc, 3`. Allowed values are  $1 \leq chan \leq 8$  but other channels are not implemented in current models.

**DAC** `dac, v`

Set voltage on SMA connector,  $\pm 2.5$  V 12-bit.

$0 \leq v \leq 4095$  corresponding to  $-2.5.. + 2.5$  V.

**POWER** `pow`

Report power supply voltages.

**TEMP** `temp,chan`  
Report temperature. `chan=1,2` for main unit and imaging sensor respectively.

**PRESSURE** `press`  
Report pressure in hPa.  
`press,Pactual`  
Define actual pressure (hPa). The value will be used to recalibrate the pressure sensor.

**PRAW** `praw`  
Report raw pressure in hPa. Pressure correction stored in EEPROM will be programmed into pressure sensor.

**WDG** `wdg,wdtime`  
Set the watchdog time, in seconds. `wdtime = 3..16` (default 3 s).  
The watchdog timer automatically resets the device if the device firmware freezes. For example, if the watchdog time is set for 1 second, and a 2 second exposure is initiated, the watchdog timer will reset the device halfway through the exposure. This facility allows automatic restart after interruptions or if there are software or firmware bugs.

**ONTIME** `ontime`  
Time in seconds since last reset or power-on.

**TBOOT** `tboot`  
Temperature at last reset or power-on.

## B.5 Imaging sensor

**SPECTRUM** `spectrum`  
Acquire currently defined field of view using currently defined exposure time, sum first two rows and return sum in binary. Thus for typical 2592 columns, return 2592 16-bit integers (two bytes each, least significant byte first). Each integer has a maximum value of  $2^{13} - 1 = 8191$ .



**SFIRST** `sfirst`

As above, but returns only the first (even) row. Returns 16-bit integers (two bytes each, least significant byte first). Each integer has a maximum value of  $2^{12} - 1 = 4095$ .

**SSECOND** `ssecond`

As above, but returns only the second (odd) row. Returns 16-bit integers (two bytes each, least significant byte first). Each integer has a maximum value of  $2^{12} - 1 = 4095$ .

**CAMERA** Commands to control the imaging sensor. The imaging sensor is typically 1944 rows of 2592 columns, and each pixel can have a 12-bit value (0..4095).

**snap** `cam,snap`

Acquire new image and return image data as plain text. The values are output as four-character hexadecimal numbers separated by spaces. Because of limited data packet size, the `snap` command is limited to image sizes of 300 pixels.

**bsnap** `cam,bsnap`

Acquire new image and return image data as binary data, two bytes per pixel. The maximum image size is 10 rows of 2592 columns, and the data is sent as two bytes, least significant first (little-endian).

**expose** `cam,exp`

Acquire new image but do not return image data. Acquired data can be used for wavelength determination with `wavelength` command.

**fov** `cam,fov,rowstart,colstart,nrows,ncols`

Define the field of view (i.e. the imaging area).

$0 \leq \text{rowstart} \leq 1944$  and must be even.

$0 \leq \text{colstart} \leq 2592$  and must be multiple of 4.

$0 \leq \text{nrows} \leq 1943$  and must be odd.

$0 \leq \text{ncols} \leq 2591$  and must be odd.

Note that the actual number of rows acquired is one greater than `nrows` and similarly for `ncols`. The maximum number of pixels, i.e. `nrows+1 × ncols+1`, must be less than 25,920.

- time** `cam,time,u20`  
Set exposure time. The value is 20-bit unsigned. The actual exposure time is approximately  $(u20 \times ncols/48 - 8.5)$  microseconds.
- atime** `cam,atime,max`  
Automatically set exposure time, with a maximum of `max` milliseconds. The exposure is adjusted until an interference peak is close to the maximum pixel value, up to a maximum of `max` milliseconds.
- bin** `cam,bin,binx,biny`  
Sum neighbouring pixels to improve dynamic range. Untested.  
`binx`, `biny` can each be 0, 1 or 3 for 1x, 2x or 4x binning.
- gain** `cam,gain,d,m,g`  
Set imaging sensor digital gain, multiplier and analogue gain. Untested.  
 $0 \leq d \leq 120$   
 $m = 0, 1$ ; set to 1 for a factor of 2 in digital gain  
 $8 \leq g \leq 63$
- vars** `cam,vars`  
Report imaging sensor field of view, exposure time, binning and gain settings. The values are tab-separated. They are:
- ```
rowstart first row
colstart first column
  nrows number of rows - 1
  ncols number of columns -1
  swu,swl exposure time
binx,biny binning (0, 1 or 3 for 1x, 2x or 4x)
  gain gain
```

## B.6 Configuration settings

**SET, GET** Set and report EEPROM configuration values. Each `set` command described below has a corresponding `get` command to report the relevant parameter.

- sid** `set,sid,u32`  
Binary serial number, unsigned 32-bit u32.
- serial** `set,serial,str`  
Serial number string str.
- iapversion** `set,iap,u16`  
Boot code version, unsigned 16-bit u16.
- wmversion** `set,wmv,u16`  
Firmware version, unsigned 16-bit u16.
- flash** `set,flash,u16`  
Flags to communicate between IAP and WM, in particular set bit 0 for firmware upload on next restart.
- ipaddr** `set,ipaddr,"xxx.xxx.xxx.xxx"`  
Set IP address based on dotted-quad string, for example `ipmask,"10.1.1.180"`. Each number should be base 10 (normal decimal).
- ipmask** `set,ipmask,"xxx.xxx.xxx.xxx"`  
Set IP mask based on dotted-quad string, for example `ipmask,"255.255.255.0"`
- ipgw** `set,ipgw,"xxx.xxx.xxx.xxx"`  
Set IP gateway based on dotted-quad string, for example `ipgw,"10.1.1.1"`
- mac** `set,mac,"xx.xx.xx.xx.xx.xx"`  
Set ethernet MAC address. The device is factory configured with a MAC address reserved by MOGLabs from the IEEE REgistration Authority. It should not be necessary to change the default MAC address.
- ipport** `set,ipport,u16`  
Set the TCP/IP port number to unsigned 16-bit u16. The device will respond to telnet queries on that port. Note that if the command is being sent via TCP/IP, and the port number changes, the communications link will be closed.

- dhcp** `set,dhcp,u16`  
Enable or disable DHCP. Zero or  $-1$  (0xffff) disables; any other value enables.
- pcorrect** `set,pcorr,int`  
Correction value for pressure, signed 16-bit int.
- camvars** `set,camvars,rowstart,colstart,nrows,ncols,swu,swl,binx,biny,gain`  
Set imaging sensor field of view, shutter time, binning and gain.
- avgcnt** `set,avgcnt,u16`  
Number of measurements to average when calculating and displaying wavelength on the device.
- row** `set,row,u16`  
Defines row of imaging sensor used for acquiring interference pattern, calculating spectrum and wavelength.  
 $0 \leq u16 \leq 1943$ , should be even.
- maxshutter** `set,max,u16`  
Maximum exposure shutter time in milliseconds, unsigned 16-bit integer. Default is 400. Note that the device will self-reset if suspended for more than about 16 seconds, imposing a hard limit on the exposure time.
- qcal** `set,qcal,double0,double1,double2,double3`  
Calibration coefficients for polynomial calibration. Each coefficient is a double-precision real number.
- scal** `set,scal,double0,double1,double2,double3`  
Calibration coefficients for trigonometric calibration.
- grating** `set,grating,double`  
Defines grating frequency (per mm), at calibration temperature.
- tco** `set,tco,double`  
Grating temperature coefficient.
- ctemp** `set,ctemp,double`  
Device temperature (in °C) at time of calibration.

**cpresure** `set, cpress, double`

Barometric pressure (in hPa) at time of calibration.

**focal** `set, foc, double`

Focal length of lens.

**age** `get, age, u32`

Currently not used.

**rgb** `set, rgb, r, g, b`

Set backlight intensity. Three unsigned integers should be provided, but most devices have one colour (white) backlights and their intensity is defined by the `r` value.

$3 \leq r, g, b \leq 254$ .

**userid** `set, userid, str`

Defines the user ID for firmware upload. Default value is “moglabs”. When uploading new firmware, you will be asked for a user ID and password. This command allows you to set the user ID to limit access to firmware changes.

**password** `set, pass, str`

Defines the password for firmware upload.

## B.7 EEPROM access

**EEPROM** Low-level commands to set and retrieve EEPROM values.

**read** `eeeprom, read, addr16`

Read single byte at address `addr16`.

**rword** `eeeprom, rword, addr16`

Read 16-bit word at address `addr16`.

**rlong** `eeeprom, rlong, addr16`

Read 32-bit long at address `addr16`.

**seq** `eeeprom, seq, addr16, num16`

Read `num16` bytes starting at address `addr16`.

- double** `eeeprom, double, addr16`  
Read a double-precision floating value at address `addr16`.
- write** `eeeprom, write, addr16, val`  
Write single byte `val` at address `addr16`.
- wword** `eeeprom, wword, addr16, val16`  
Write 16-bit `val16` at address `addr16`.
- wlong** `eeeprom, wlong, addr16, val32`  
Write 32-bit `val32` at address `addr16`.
- page** `eeeprom, page, addr16, num, str`  
Write `num` characters from string `str` to address `addr16`. String should be in double-quotes, for example  
`eeep, page, 4, 16, "W1001101140602"`  
Warning: data must fit within 128-byte page boundary.
- str** `eeeprom, str, addr16`  
Write string `str` to `addr16`.  
Warning: data must fit within 128-byte page boundary.
- freset** `eeeprom, freset`  
Restore configuration from factory defaults.
- fset** `eeeprom, fset`  
Copy current configuration to factory default preservation area.

# C. mogwave configuration

mogwave can be configured via command-line options for some critical settings, and more generally through a configuration file. Note that command-line options override values set in the configuration file.

## C.1 Command-line options

To change the command-line options, edit the shortcut link to mogwave, typically the desktop link created during installation.

1. Right-click on the desktop or Start menu shortcut, then select **Properties**.
2. Edit the **Target** to add the appropriate command-line options, for example:

```
C:\Program Files\MOGLabs\mogwave\mogwave.exe -u -p 15
```

to specify USB on port 15.

3. Press **OK** to save the shortcut changes.

The shortcut should then start your software appropriately.

You can also specify network connection, for example:

```
...\mogwave\mogwave.exe -ip 10.1.1.18
```

where the IP address should match what shows on the device screen.

Additional command line options:

- h help
- c configuration filename
- d debug options
- ip IP address
- l enable logging of errors
- m maximum exposure time (ms)
- p TCP/IP port or USBCOM channel
- r sensor row
- s simulation wavelength
- t timeout multiplier
- u USB enable
- w HighFinesse WLM (y/n)

- c The configuration filename; see below. *mogwave* will first try to open the file within the current working directory (the directory where *mogwave* is located, or the “Start in” directory specified in the shortcut link). If not found, it will search for the file in the default configs directory within the *mogwave* directory.
- d Debugging flags. Integer value; set odd to save etailed measurements when recording data.
- m The maximum exposure time is by default 400 ms. Longer times can make the software unresponsive but may be useful for measuring very low-power lasers or pulsed lasers with low pulse rate.
- r The sensor row specifies the pixel row on the sensor to use for data acquisition. The default is set in the factory but adjustments may be needed when changing fibres or after mechanical disturbance.
- s Floating point wavelength in nm. *mogwave* can simulate measurement if no laser light is available. Value must be within wavelength measurement range.
- t The timeout multiplier is used to make the software a little more patient in waiting for responses from the wavemeter device, particularly



where the network connection is slow.

- w Enable HighFinesse WLM wavemeter reading for calibration (see section 4.2). Values are y, yes, n or no.

See Chapter 2 for more information on communications options and Chapter 3 for more information on the `mogwave` application.

## C.2 Configuration file

Default values for communications and other useful parameters are defined in a configuration file. The default location is in the `configs` directory within the `mogwave` home directory, and the default file is `default.cfg`, for example

```
C:\Program Files\MOGLabs\mogwave\configs\default.cfg
```

The config file is in plain text, in the Microsoft `.ini` format, with a number of sections, and within each section, a number of options. The default data type is string. The default config file includes comments on the meaning of the different parameters.

Important parameters include the font size, default directory (folder) for saving data files, default TCP/IP address, USB port, maximum shutter time, measurement interval, and the spectrum plot colour.



# D. mogwave server

`mogwave` provides a server that can be accessed via TCP/IP, providing an API so that wavelength measurements are accessible to other applications such as `matlab` or `LabVIEW`. To use the server, you should start `mogwave` and ensure that it is making sensible measurements. You can then stop `mogwave` and make single measurements on demand through the server, or leave `mogwave` running and access the most recent measurement from the server as needed.

## D.1 Enabling the server

The server is accessed via TCP/IP on the host computer that is running `mogwave`. When `mogwave` starts, it will establish a TCP/IP socket on the host computer. The TCP/IP port is defined in the `default.cfg` file (see C.2); if 0, the server is disabled. A typical port value would be 7803.

## D.2 Communication with the server

Communication with the server is via TCP/IP, with a simple command/response `telnet` protocol. For example, if the host computer running `mogwave` has IP address of 10.1.1.21:

```
telnet 10.1.1.21 7803
help
```

Here the `telnet` command line specifies IP address 10.1.1.21 and port 7803. You can also use a GUI `telnet`, in particular on Windows<sup>TM</sup> we recommend PuTTY available at [www.putty.org](http://www.putty.org). Specify the Host Name as 10.1.1.21, Connection type as Telnet, and the Port which is normally 22, should be changed to 7803.

The above example sends the `help` command to `mogwave`, which will report back something like:

```
mogwave running wavemeter W2001101140930
Wavelength range 766.2 to 788.9nm, difference 22.7nm
Valid commands: wave, last, close, start, stop, die, kill
```

### D.3 mogwave commands

The `mogwave` server is controlled via simple text commands of the form `wave`. The commands must match exactly, with no abbreviations or extensions. For example, `wave` is good, but `wav` and `waver` are not.

Each command will respond with “OK:” if valid, or “ERR:” if invalid or some error occurred, followed by values or further information.

#### START `start`

Start continuous measurement.

#### STOP `stop`

Stop continuous measurement.

#### WAVE `wave[,units]`

Returns most recent measurement if acquiring; initiates a single new measurement if not.

**units** Units for return value; can be `nm vac`, `nm air`, `nm raw`, `THz`, `cm`. If no units specified, return value is raw measurement in nm.

The response includes the time of measurement, for example:

```
last,nm vac
OK: 16:59:08.8 779.984977
```

#### WIDTH `width[,units]`

Returns peak width of most recent measurement if acquiring; initiates a single new measurement if not. Note that the values returned by `wave` and

`width` are not necessarily associated with the same instantaneous spectral measurement. See above for optional units specification.

#### UNCERT `uncert[,units]`

Returns the statistical uncertainty for the most recent measurement if acquiring; initiates a single new measurement if not. Note that the values returned by `wave` and `uncert` are not necessarily associated with the same instantaneous spectral measurement. See above for optional units specification.

#### SPECTRUM `spectrum[,units]`

Returns most recently acquired spectrum, in text format, tab-separated wavelength and intensity values. Use the `wave` command to acquire a spectrum if not running acquisition continuously. See above for optional units specification.

#### EXPOSURE `exposure[,value]`

Control the sensor exposure time. If `value` is omitted, returns the current exposure duration in milliseconds.

`value` `auto` or `fixed` to enable/disable auto-exposure, or numerical value to set the exposure time in milliseconds.

#### PEAK `peak`

Return position and height of peak. Height is the largest integer value in the raw spectrum, and position is the integer number of the highest value (use `wave` to find calibrated wavelength).

#### FAST `fast[,state]`

Switch between fast and normal modes. Fast mode operates without showing the small 2d images of diffraction spots.

`state` State should be `True` or `1` to enable fast mode, `False` or `0` to return to normal mode. If no state specified, defaults to `True`.

**TWOD** `twod[,state]`

Switch between high accuracy 2d mode and faster 1d mode. See above for definition of state.

**CLOSE, EXIT** `close` or `exit`

Close the TCP/IP socket (i.e. disconnects).

**KILL, DIE** `kill` or `die`

Terminate mogwave.

# E. Upgrading firmware

1. Connect power and ethernet connection.
2. Use a web browser (e.g. `firefox`) to load the device webpage, based on the IP address shown on the device display.
3. On the web browser, select `Function, Update firmware`.
4. Click on `Reboot to firmware upload mode`.
5. Watch the device display until it shows the IP address, then open your web browser at that address.
6. At the prompt, enter your user ID and password. Initially these are `moglabs` and `wavemeter`.
7. Select `Browse` and then select the new `MOG-WM.bin` firmware file.
8. Click on `Upload`.
9. Watch the device display, and when all packets have been uploaded, reset the device (disconnect from power and reconnect).





# Bibliography

- [1] J. D. White and R. E. Scholten. Compact diffraction grating laser wavemeter with sub-picometer accuracy and picowatt sensitivity using a webcam imaging sensor. *Rev. Sci. Inst.*, 83:113104 (2012). <http://dx.doi.org/10.1063/1.4765744>
- [2] P. J. Fox, R. E. Scholten, M. R. Walkiewicz and R. E. Drullinger. A reliable, compact and low-cost Michelson wavemeter for laser wavelength measurement. *Am. J. Phys.* 67:624-630 (1999).





