Characterisation Procedure for TELLIE LED Drivers

E. Leming, J. Maneira, M. Mottram, S. Peeters, J. Sinclair and J. Waterfield
SNO+
University of Sussex

May 22, 2015

Figure 1: *Please remember to try and stay positive thoughout these tests...*

Introduction

This document details the characterisation procedure for the Push-Pull LED drivers and the LEDs used by the TELLIE system. These procedures were originally run at Sussex before shipping TELLIE to SNOlab. Some revisions have since been made, in particular updates to the software and data taking / processing procedures, for re-running the calibration on site a SNOlab.

TELLIE driver testing

Overview

The driver characterisation setup is shown in Figure 2. This setup runs the drivers as they will be at site, using a monitoring PMT and oscilloscope to characterise the time profile and

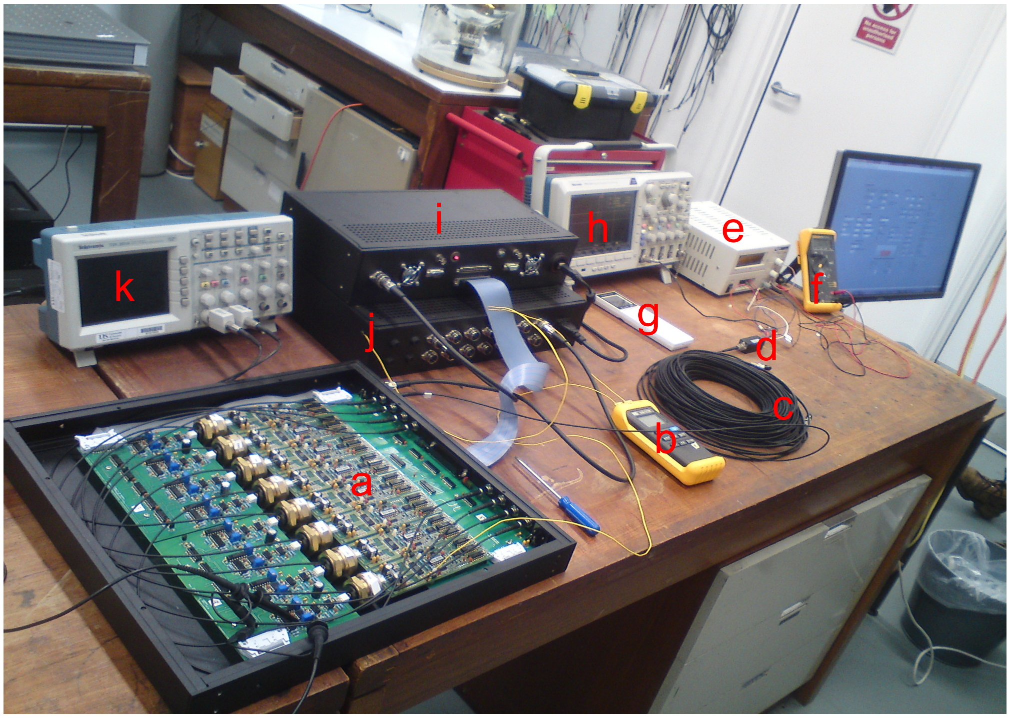


Figure 2: *PIN and driver characterisation setup. Driver box (a), Temperature monitor (b), Optical fibre (c), PMT (d), PMT power supply (e), Multimeter (f), Air-conditioning thermostat (g), Oscilloscope (h), Control box (i), Power supply for driver and control boxes (j), Oscilloscope (k).*

amplitude of their LED pulses. The measured amplitude can then be compared with the internal PIN diode's response. The driver box (a) is connected via the optical fibre (c) the PMT (d). The PMT power supply (e) is monitored by the multimeter (f). The oscilloscope (h) is used to measure the rise time, fall time, width and the amplitude of the PMT pulse. The control box (i) for the drivers is powered by the same supply as the driver box (j). The second oscilloscope (k) is used to set up the PIN's amplifier, shown in detail in Figure 4(b). The temperature within the driver box is measured with the temperature monitor (b), the air-conditioning thermostat (g) is used to keep the lab temperature constant.

At site the driver boxes will be mounted in a rack on deck, each connected via 2 m of optical fibre to the patch panel. The patch panel then connects via 45 m fibres to the PSUP nodes. Hence, for these tests the drivers will be connected to the monitoring PMT in a similar arrangement using a 2 m + a 45 m fibre.

The stack of driver boxes are operated through the *control box*. The control box contains a PIC-18F452 Peripheral Interface Controller (PIC) chip, pre-loaded with a set of C functions which can be called to control the system. During SNO+ data running functions will be called by the DAQ system (currently ORCA), but during these tests the control box will be interfaced with a set of python scripts running off a the TELLIE DAQ laptop. The laptop plugs directly into the control box via USB. The python scripts themselves can be found at [1].

Each of the twelve *driver box's* contains eight driver channels ($8 \times 12 = 96$ channels in total). A single driver channel is defined by a Push-Pull driver and it's associated LED and a PIN diode to monitor the light response, which are held within a brass "cone". The light is coupled out of the "cone" by a 0.45 m length of PMMA optical fibre. These fibres are drilled into the LED's plastic casing one end, terminating at the box output with Thorlabs ST connectors. Each of the Push-Pull driver boars are mounted on a motherboard which is connected via a ribbon cable to the control box.

The Push-Pull drivers were designed with three current variables which could be used to manipulate the driver's pulse shape: IBI, IOP and IPW. IBI sets the maximum length of time that a potential difference can be applied across the LED, IBI is fixed on this version of the drivers. IOP sets the size of the potential difference across the LED. IPW sets the time that the potential difference is applied for, controlling the width, and to an extent the intensity. As IPW approaches IBI the pulse goes to zero. It had been shown in tests carried out by Dr. Matt Mottram and James Waterfield that the IPW has the greatest effect on the photon intensity. In order to minimise uncertainties added by correlations between the IOP and IPW parameters, the IOP is held constant and only IPW is varied for these measurements. Further, detailed discussion of the drivers can be found here [2].

Connections

For all measurements the lab temperature is maintained at approximately 21° C using the air-conditioning, with the thermostat mentioned above. For the re-calibration to be done at SNOlab, the procedures are to be carried out on the deck (or in the control room) where the temperature is constantly controlled.

Before any fibre connections are made, each end is to be cleaned with the Thorlabs fibre connector cleaner, (c) in Figure 11(a). The 'fitted connections' at the output of each driver box should also be sprayed with pressurised air to remove any dust.

The 'TELLIE DAQ' MacPro used to run the python scripts [1] connects to the control box via a standard A-B USB. The TELLIE DAQ MacPro is stored underground at all times so will be available for the re-calibrations at SNOLab. A back-up is also available underground in case of any major failures.

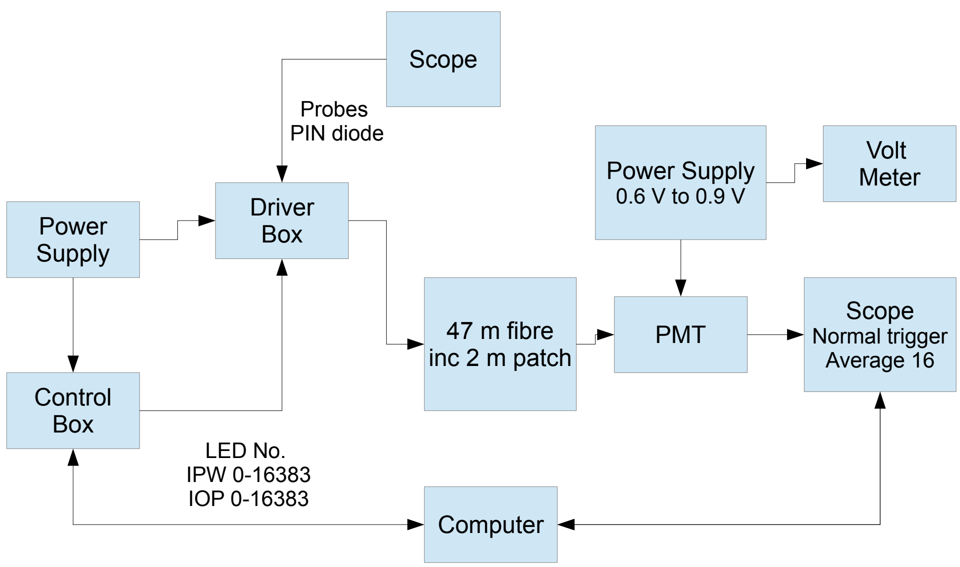
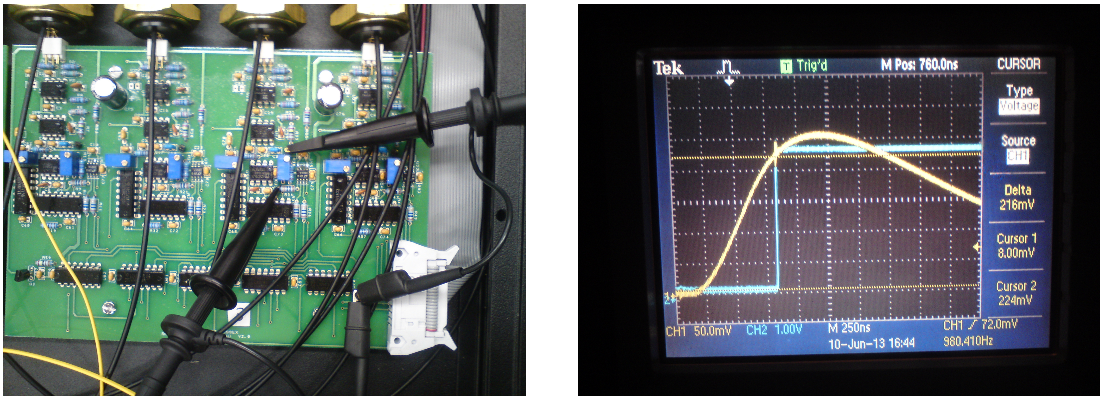


Figure 3: *Basic setup of PIN and driver characterisation.*



(a) *Probe connection points for PIN diode.*

(b) *Scope displaying PIN pulse (CH1) and sample point (CH2), the gain determines the aptitude of the pulse.*

Figure 4: PIN diode probes and pulse shape

The driver box is connected to the control box via a ribbon cable, onto which seven driver boxes can be daisy-chained. As the control box has ports for two robin cables, fourteen driver boxes can be connected to one control box.

On the motherboard of the driver box, there is a Board-Select jumper next to the control box connection, which identifies the driver box to the control box, as box 1 to 7. Driver box 1 contains channels 1 to 8, box 2 channels 9 to 16 and so on, until channel 112 in box 14. To address a specific channel, only its number is needed.

For the initial test to optimise IOP and find the cutoff value for IPW, each driver is placed into a reference channel so that the same LED and PIN diode are used on every driver board, providing a measurement of the board to board fluctuations in performance. For the characterisation of the operational range of IPW the driver is place into it's own channel, where the PIN diode first has to be optimised for the individual driver-LED combination.

Each PIN diode has two potentiometers, shown in Figure 4(a), that allow the PIN's gain and sample point to be set. Probes can be connected to the 330 Ω resistor and the probe point also shown in Figure4(a). The gain is set to allow maximal sensitivity to the low light levels without saturating at higher levels, the value varies for each board, normally approximately 2.5 V. The sample point is set to approximately 10 to 15% below the peak on the leading edge of the PIN pulse, see Figure 4(b).

PIN model number and amplification circuit... .. ask Dick tomorrow

The PMT used here is a Hamamatsu H10721-210, with the control voltage set to 0.6 V or

0.8 V depending on pulse intensity. Hamamatsu calibrated the PMT on the 30/08/2010, at 1 V its gain was 1.80×10^6. Using the plot of gain as a function of control voltage, see Figure 5(b), an offset can be found by calculating the difference between the gain at 1 V and the calibrated value, this can be crudely used to find the gain at any control voltage. A value of the control voltage is recorded manually for each dataset.

The QE can be calculated from the cathode radiant sensitivity,shown in Figure 5(a), which is $77.4\,\mathrm{mA\,W^{-1}}$ at 501 nm, as follows:

$$\frac{77.4 \times 10^{-3}\,\mathrm{CJ^{-1}}}{1.6 \times 10^{-19}\,\mathrm{C}} = 4.84 \times 10^{17}\,\mathrm{eJ^{-1}}$$

$$\frac{1}{E_\gamma} = \frac{\lambda}{hc} = 2.52 \times 10^{18}\,\gamma\mathrm{J^{-1}}$$

$$\frac{4.84 \times 10^{17}}{2.52 \times 10^{18}} = 19.2\%$$

Software

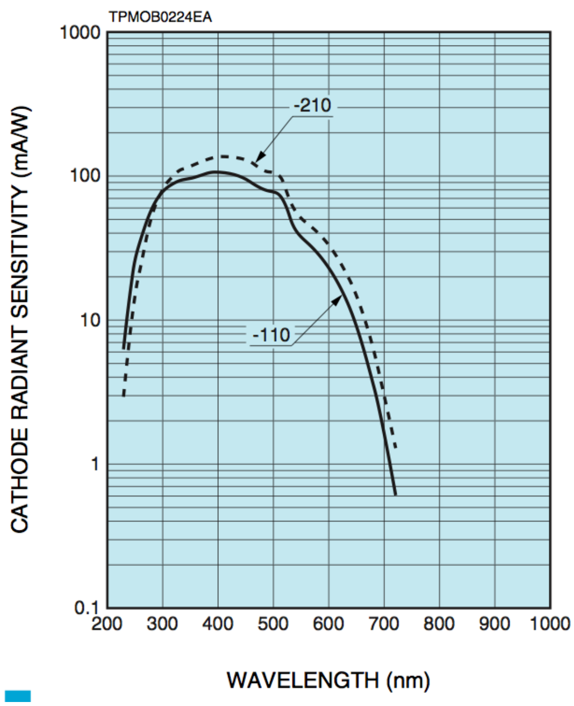
System Requirements: Python is needed (written and tested with python 2.7.3). In addition, the FTDI driver and the pySerial module must be installed to communicate with the control box. pySerial can be downloaded from here: http://pyserial.sourceforge.net/ and the FTDI driver from there: http://www.ftdichip.com/Drivers/VCP.htm.

For the characterisation of the drivers, they are controlled by python scripts within a LabView program, which also records the results. The LabView front end is shown in Figure 6. [4] describes the details of the control software, including how the driver settings are applied.

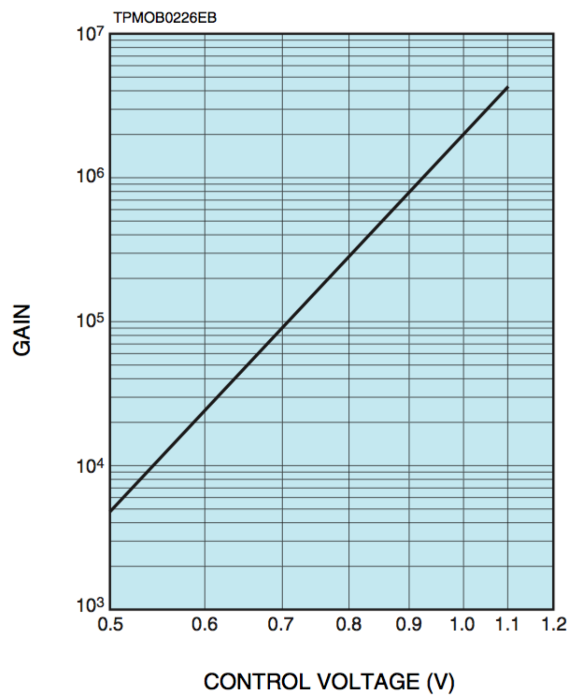
For each new scan the user selects the channel number and the ranges of IOP and IPW, as well as the step size of the scan in each parameter. The curser positions have to be set[1], defining the window within which pulses will be analysed.

The following settings should be left at the given values for each run: Number of readings = 1, VISA resource name = USB1, Trigger Slope = 1 (falling), Trigger coupling = 0 (DC), Fibre = 0.5 (short), Gating = 1 (between cursors), Acquisition type = 4 (averaging), Number of averaged waveforms = 16, Pulse delay hi = 0, Pulse delay lo = 250, Number of pulses hi

[1] The positions are determined manually by running the driver continuously at the maximum settings to be used in the scan



(a) *Cathode radiant sensitivity, from which quantum efficiency can be derived.*



(b) *PMT gain as a function of control voltage. PMT used here has is calibrated to have a gain of* 1.8×10^6, *at 1 V.*

Figure 5: PMT calibration plots taken from [3].

Parameter Name	Hi/Lo Range	Internal Calculation	Range	python function
Pulse Width	Hi 0 - 63 Lo 0 - 255	par=(Hi*256)+Lo	0 - 16383	setPulseWidth(par)
Pulse Height	Hi 0 - 63 Lo 0 - 255	par=(Hi*256)+Lo	0 - 16383	setPulseHeight(par)
Number of Pulses	Hi 0 - 255 Lo 0 - 255	par=Hi*Lo	0 - 65025	setPulseNumber(par)
Delay between Pulses	ms 0 - 255 μs 0 - 255	par=ms+(μs*4)	0 - 255 ms 0 - 1020 μs	setPulseDelay(par)
Fibre Delay	ns 0 - 255	par=ns*0.25	0 - 63.75 ns	setChannelDelay(par)
Trigger Delay	ns 0 - 255	par=ns*5	0 - 1275 ns	setTriggerDelay(par)

Table 1: TELLIE parameter range, along with the python function to set each of them.

= 10, Number of pulses lo = 100. Table 1 shows how the parameters are interpreted by the python script.

The filename format is important as the macros used to process the data read important information from it, the format should be as follows:

BoardNo_TestType_Date_Temperature_GainVoltage.

Board number is three digits long, date is YYMMDD, temperature is three digits long with no decimal point and control voltage is in mV, e.g. 013_IOPopt_130605_226_596 refers to driver 13, in an IOP optimisation run, on the 5/6/2013, at a temperature of 22.6° C, with PMT control voltage of 0.596V.

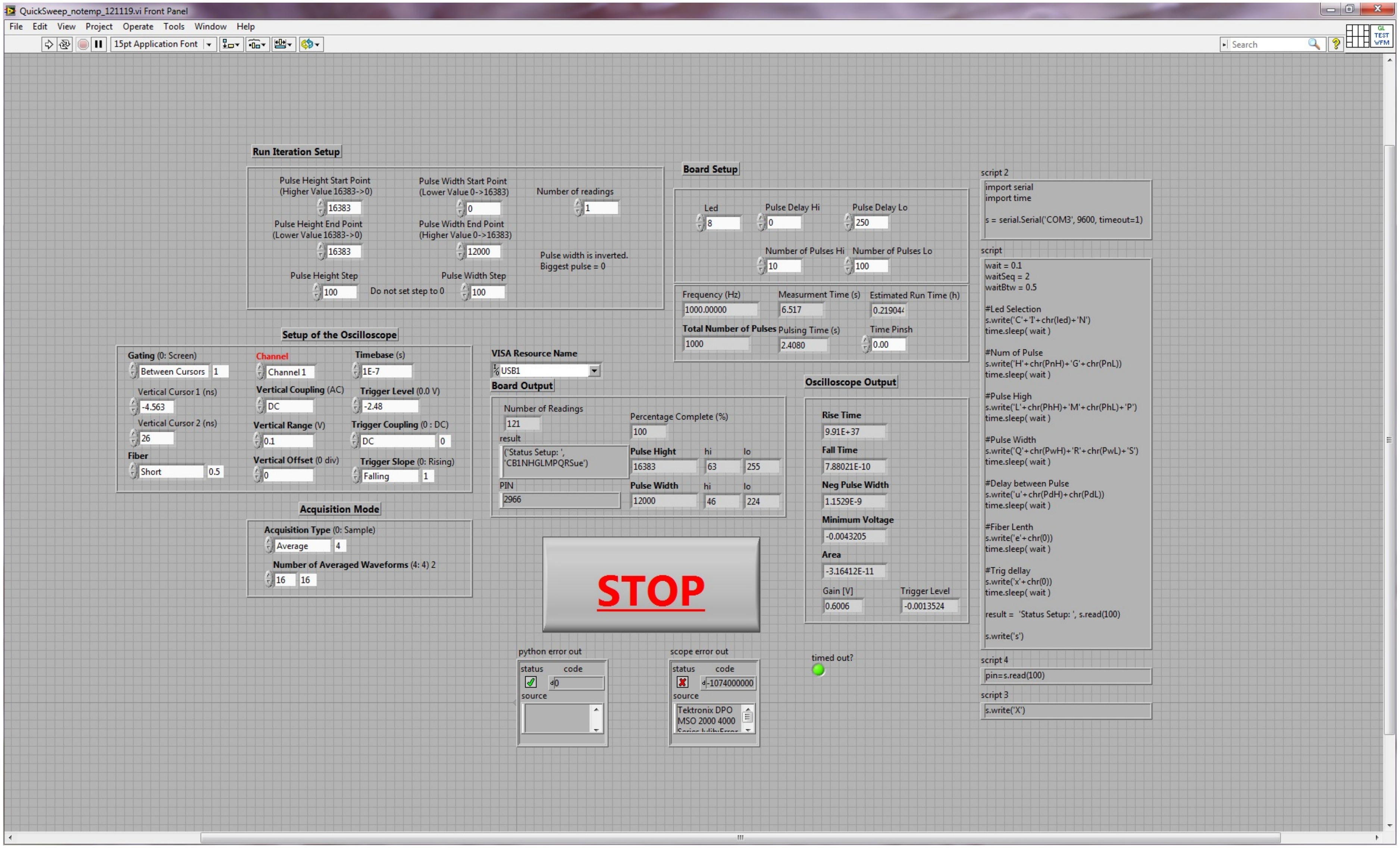


Figure 6: *The front end of the LabView program used for characterisation of drivers.*

IOP Optimisation

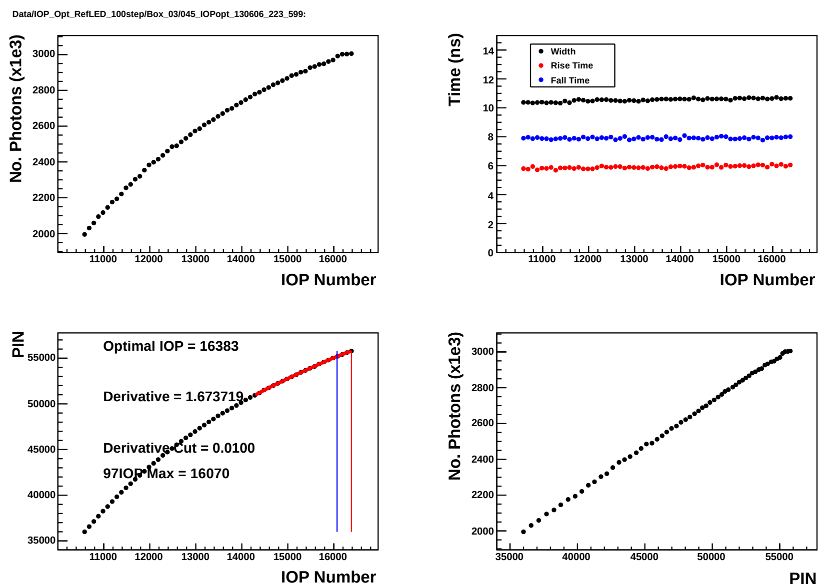


Figure 7: *IOP scan with reference LED.*

Using the reference LED, IPW is set to 0 (widest possible pulse) as IOP is varied from 10583 to 16383, in steps of 100. If number of photons plateaus before IOP reaches its maximum then optimal values of IOP is set at start of plateau, but if the number of photons is still increasing at the maximum IOP then the maximum (16383) is set as the optimal value, see top left of Figure 7.

Finding range of IPW

Again, using the reference LED, IOP is set to its optimal value as IPW if varied from 0 to 12000, in steps of 100. As mentioned the LED pulse will go to zero, as IPW approaches IBI, this vale of IPW is selected as the cutoff point, the red line in Figure8.

Optimise PIN diode

Using the individual LED, repeating the initial IPW scan with, IOP is set to its optimal value as IPW if varied from 0 to 2500, in steps of 100. If the PIN response saturates (plateau a IPW goes to 0) and the relationship between PIN and number of photons seises to be

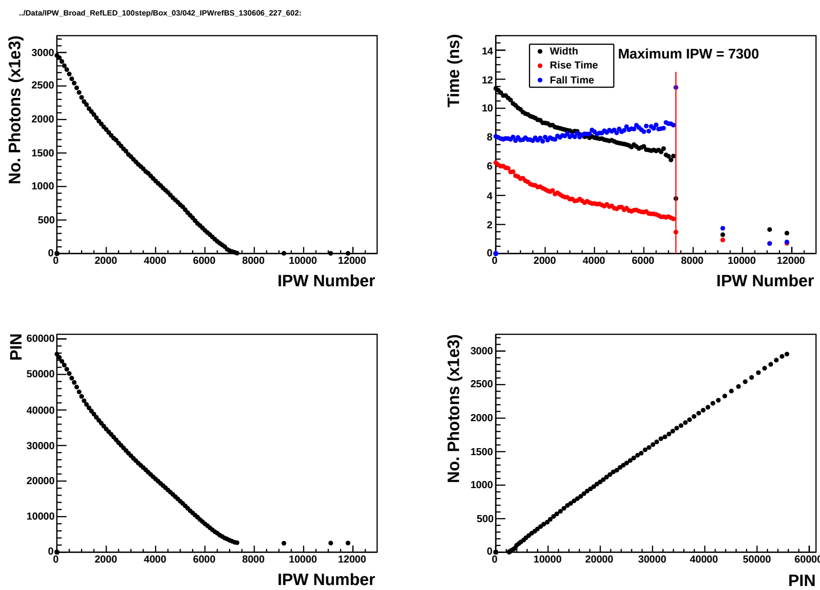
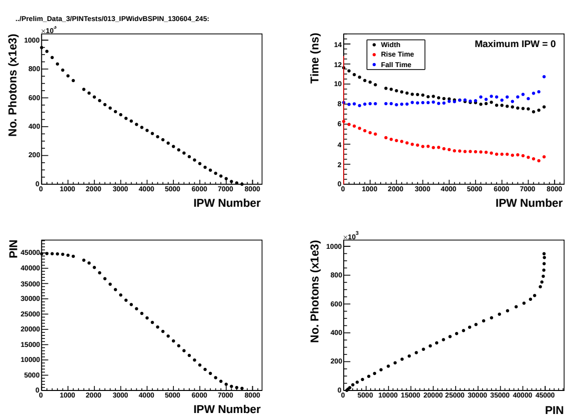
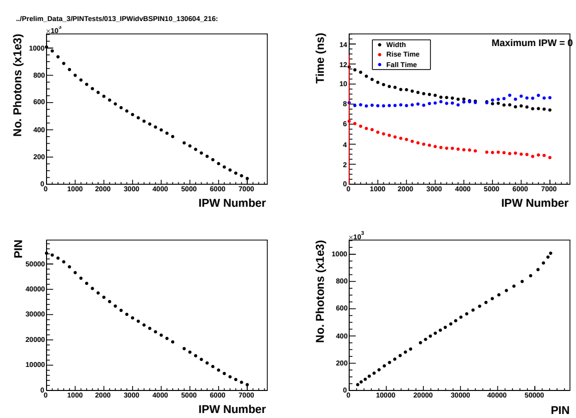


Figure 8: *IPW scan with reference LED showing cutoff, upper right.*

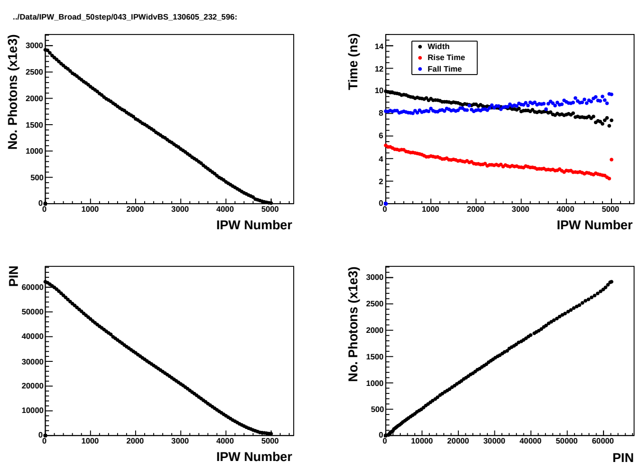


(a) *Saturating PIN diode, lower left. Non-linear relation in PIN vs photons, lower right*

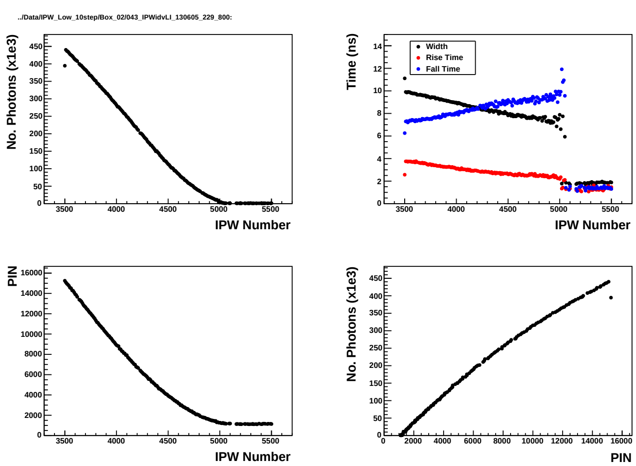


(b) *Non-saturated PIN diode. Linear PIN vs photons*

Figure 9: PIN optimisation plots.



(a) *Broad scan of IPW with IOP optimised*



(b) *Low end scan of IPW with IOP optimised*

Figure 10: IPW characterisation plots for same channel.

linear for larger numbers of photons, see Figure 9(a), then PIN diode has to be tuned as described above, with IPW at 0 and IOP at 16383. The scan is repeated until a linear relationship between PIN and number of photons can be observed across the range of IPW, as in Figure 9(b).

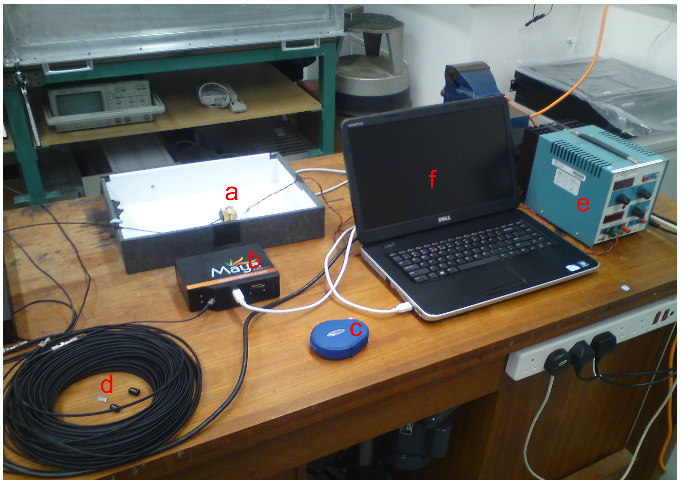
IPW characterisation

Using the individual LED, two scans are taken of IPW, one across the full range and a detailed one of the low end. For the full range, IOP is set to its optimal value as IPW is varied from 0 to its cutoff + 500, in steps of 50, see Figure 10(a). For the low end IOP is set to its optimal value as IPW if varied from cutoff-1500 to cutoff+500, in steps of 10, see Figure 10(b).

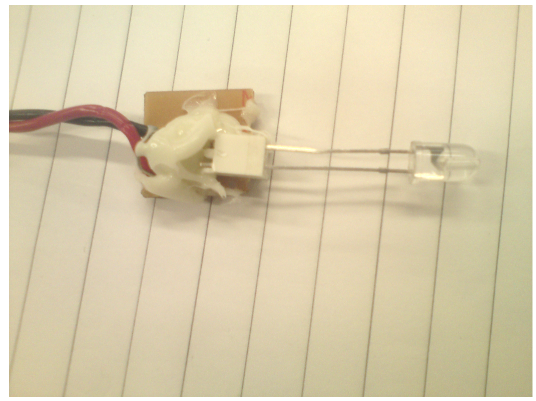
LED test set-up

Overview

The LED spectral measurement setup is shown in Figure 11(a). There is a simple circuit (a) used to drive the LED at a constant voltage provided by the power supply (e). The same



(a) *LED spectral test setup.*



(b) *Close up of driver used for spectral measurements.*

Figure 11: LED test setup

optical fibre (d) used in the measurements described above is used to connect the LED to the spectrometer (b) which is controlled by the laptop.

Connections

The measurement can only be carried out on LED that have fibres glued into them. They do not have to be mounted in cones or in the driver box, but if they are not then measures should be taken to prevent background light affecting the measured spectrum, in Figure 11(a) a box covered in a black sheet is used.

The circuit used to drive the LED is simply a 550 Ω resistor in series with the LED, supplied with a constant 2.48 V. To prevent the LED legs shorting on the components of the, by being instead too far into the connection, the Circuit has been coated in an layer of insulations.The anode (long leg) of the LED has to be mounted in the connection closest to the red wire.

The spectrometer used is an Ocean Optics Maya 2000 Pro. The LED is connected to the spectrometer with 45 m of optical fibre, the 2 m patch is not used, however there is a 0.2 m patch used at the spectrometer to match the ST fibre connection to the SMA spectrometer connection.

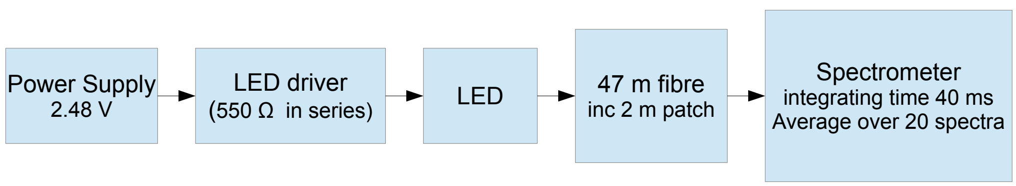


Figure 12: *Basic setup of LED spectral measurements.*

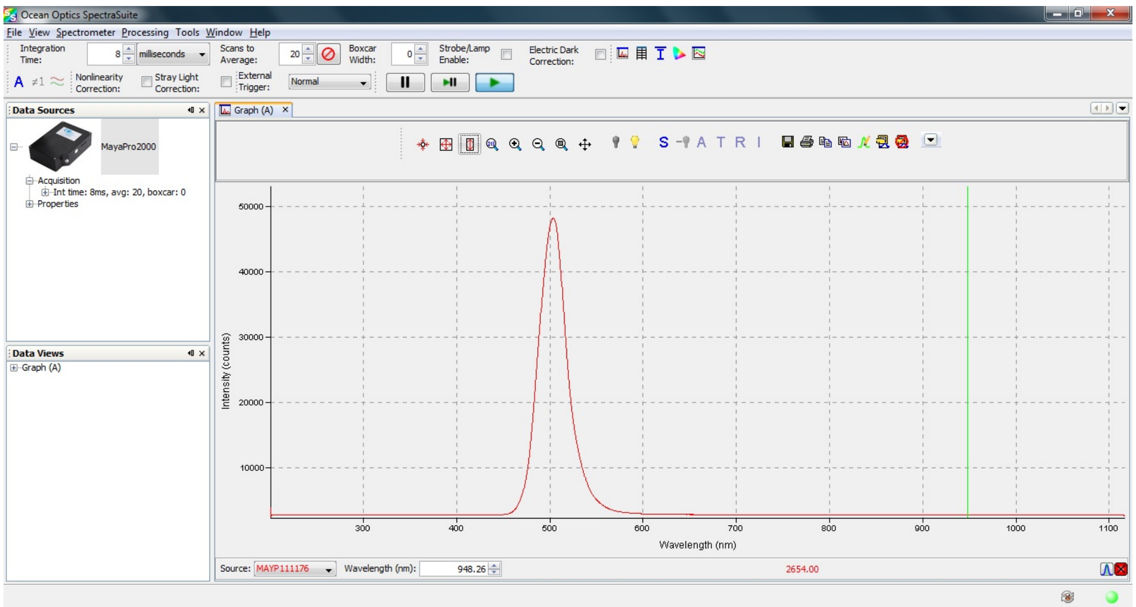


Figure 13: *The front end of the spectrometer software, sowing an LED spectrum.*

Software

Ocean Optics SpectraSuite, shown in Figure 13, is used to record the LEDs spectra. The integrating time is set to 40 ms and each spectrum is averaged over 20 scans.

LED characterisation

After the LED's spectrum has been recorded, a macro is used to fit it. The peak wavelength is stored along with the intensity, since the spectra are not Gaussian, the half-width at 50% and 20% of the maximum are recorded for data to the left and right of the peak. The integration of the tail (25% of maximum to the right of the peak) is taken and the ratio of that to the total intensity is stored.

Data structure

The data is stored in an individual root file. The root file has a TDirectory structure as indicated in figure 14. The ID number labelled on each component is stored for each individual box as a TVector. These components are a mother board, MB; a left and right PIN board as

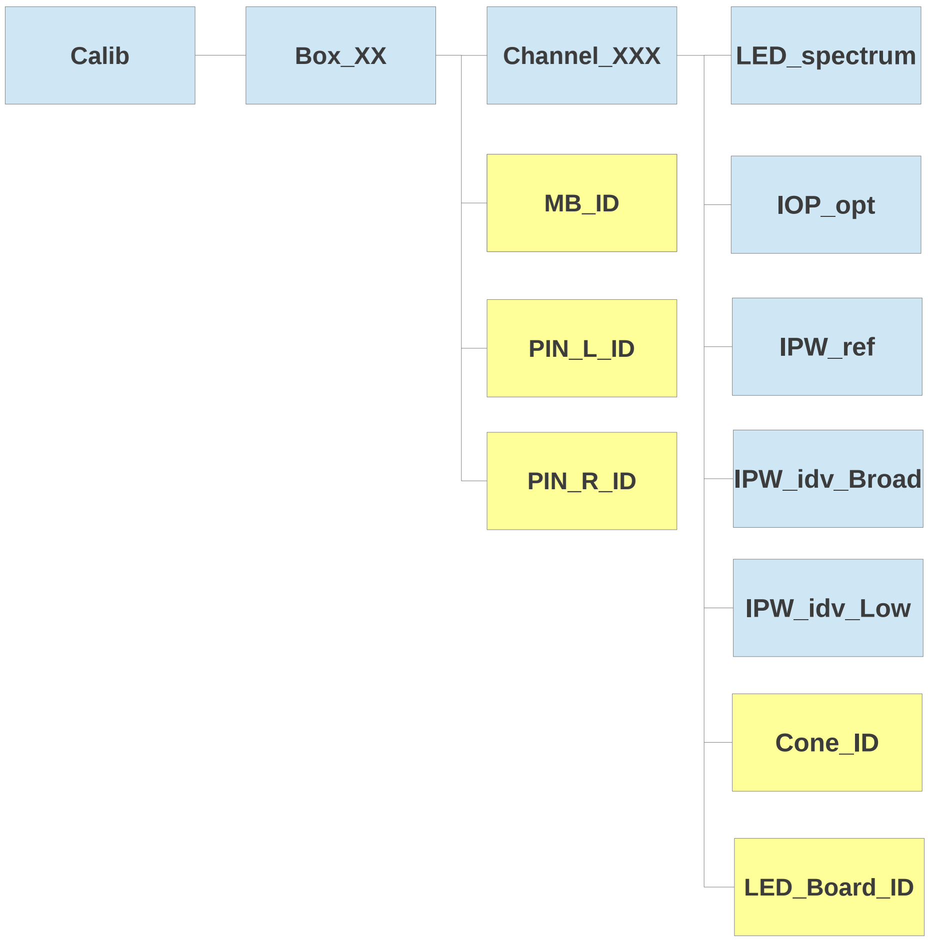


Figure 14: *Data structure of the root file "PushPullTests.root". The root file has a TDirectory structure. The blue boxes indicate a subdirectory where the 'X's indicate an integer which identify the boxes ranging from 01 to 13 and channel number from 001 to 120. The yellow boxes are TVectors which are an ID number which identifies the various components of the box. MB is the mother board; PIN_L is the left PIN board as viewed facing the channels; PIN_R is the right PIN board; LED_Board is the push-pull board and Cone is the LED which corresponds to that board.*

viewed as facing the channels, PIN_L and PIN_R; 8 push-pull driver boards, LED_Board and a cone which houses the LED for the respective board which is referred to in the root file as Cone. Within the LED_Spectrum the data of the LED characterisation is stored as TGraph in the form of counts as a function of wavelength in nm. In each of the IOP and IPW tests the temperature and PMT gain voltage will be stored as TVectors and a set of TGraphs will be stored. These are:

- photons_v_IPW or IOP - Number of photons as a function of IPW or IOP.
- rise_v_IPW or IOP - Rise time in seconds as a function of IPW or IOP.
- fall_v_IPW or IOP - Fall time in seconds as a function of IPW or IOP.
- width_v_IPW or IOP - FWHM of the pulse as a function in seconds as a function of IPW or IPW.
- PIN_v_IPW or IOP - PIN readout as a function of IPW or IOP.
- photons_v_PIN - Number of photons as a function of PIN readout.

Figure 15: *...well done, you look tired, go home and rest.*

Bibliography

[1] Leming. `https://github.com/Sussex-Invisibles/TELLIE_calibration_code`.

[2] R. Alves, S. Andringa, S. Bradbury, J. Carvalho, D. Chauhan, K. Clark, I. Coulter, F. Descamps, E. Falk, L. Gurriana, and et al. The calibration system for the photomultiplier array of the sno+ experiment. *Journal of Instrumentation*, 10(03):P03002P03002, Mar 2015.

[3] Hamamatsu. *Metal package PMT photosensor modules H10720/H10721 series.*

[4] G. Lefeuvre. Ellie control integration. *SNO+ DocDB*, (1682), 2012.