# A LABVIEW INTERFACE

# BTECH 451 (A)

RINA GAO

1857858

# CONTENTS

# ABSTRACT

BTech 451A is the first third of the BTech451 course, where the research phase of the LabVIEW Interface project is carried out. During this period, I investigate the real-time all-fibre spectroscopic device named the Optrode and the scope of the project. The Optrode measures fluorescence in microbiological substances, and is in development by the Physics department at the University of Auckland. The LabVIEW Interface project has two aims: code upgrade, and interface improvement. The existing Optrode code is incompatible with recent versions of LabVIEW, so the code must be upgraded. The user interface must also become more intuitive to become a commercially viable product. The current interface is complex; hence it is difficult and visually unappealing for users to learn and operate. This report presents the purpose and operation of the Optrode, then relevant literature review for the dual aims of the project, the course of action taken through BTech451A, a summary of the tasks remaining, and concludes with a plan of action for BTech 451B.

# 1 INTRODUCTION

## 1.1 COURSE OVERVIEW

BTech 451 is a full-year course honours project, designated 15 points (or one course credit) in the first semester under the course name BTech 451A, and 30 points (two course credits) in the second semester under the course name BTech 451B. This report summarises the findings and progress I have made in the first half of the year, and proposes a plan of action for the remainder of the year. The senior lecturer developing this Optrode is Frederique Vanholsbeeck [1], and she is also the owner of this LabVIEW Interface project. My project focuses on a LabVIEW [2] program developed for a spectroscopic system called the Optrode [3, 4, 5]. The Optrode's software and hardware components were designed in the Physics department of University of Auckland, modified and improved by past students [6, 7, 8, 9] and have produced results for a number of academic papers by biologists and physicists around the world [3, 4, 5, 6, 7, 8, 9, 10, 11].

## 1.2 PROJECT BACKGROUND

The Optrode and its LabVIEW interface have been incrementally improved on by a number of students of the Physics department at the University of Auckland over the past six years as mentioned in the course overview. As a result, the system has been patched in a number of places with little coherence, and simply made-to-work. Consequently, it faces a number of crucial hardware, software, and visual upgrades in preparation for prospective commercialisation. First, the function library for the MCC DAQ card [12] used in the existing Optrode code is Universal Library (UL) [13]. However, a new library named ULx [14] has since been released by MCC. Thus for commercialisation, the code which is written using UL must be translated and updated to use ULx. There are a number of functions that require implementation or adjustment within the existing code, adding to the code upgrade segment of the project. Finally, the operating system hosting the Optrode has upgraded from 32-bit to 64-bit Windows, resulting in path dependency issues that need to be resolved. Lastly, the Optrode interface needs to be simplified and become more 'intuitive', as the existing interface has not been developed with user-friendliness in mind.

This BTech 451 project aims to achieve the following goals: **(i) upgrade the code behind the Optrode; (ii) improve the user interface to be more intuitive**.

To reach the aforementioned goals of the project, a number of literatures are reviewed. These relate to the Optrode; LabVIEW; UL and ULx; user interface design; and usability studies. The remainder of this report is structured as follows: Chapter two will present my research and understanding of the Optrode and the code and user interface problems that the project aims to solve. Chapter three will detail the development I have made on the Optrode during BTech 451A with the problems I faced. Chapter four will conclude the report with a summary of the tasks remaining in this project, and a plan of action for BTech451B.

# 2 LITERATURE REVIEW

## 2.1 OPTRODE

In order to improve the Optrode, it is imperative to first explore the theory behind why and how the Optrode operates. A number of studies have been published using the results of experiments conducted with the Optrode over the last few years [3, 4, 5, 10, 11]. The primary function of the Optrode is to deliver and detect light, in order to measure the fluorescence of substances through an optical fibre. The Optrode can use single fibre to both deliver light and collect a signal, or dual fibre probes to deliver and collect respectively. Patel et al [5] optimised the performance of the Optrode in 2013, by increasing the collection volume and system sensitivity. This allowed the Optrode to reliably acquire accurate quantitative measurements from the test subject. Recent applications of the Optrode include measuring cardiac activity [11] and microbiology monitoring such as counting the bacterial population in water of different origins [3, 4].

The Optrode, shown in Fig. 1, is described by Bogomolny et al. (2014) [4] as follows. The optical shutter is controlled by the data acquisition (DAQ) card. It is placed in front of the fibre coupling stage to synchronise the exposure time of the sample and the acquisition time of the spectrometer. Half the excitation light from the laser is guided to the sample via one of the output arms of the coupler through the SMA connector, while the remaining is directed to a photodiode power metre to monitor the laser noise. Through the tip of the probing fibre, a fraction of the fluorescence is collected from the illuminated sample and guided to the 16-bit spectrometer through the fibre coupler. The remaining excitation light is blocked from the spectrometer by a long-pass filter. The spectrometer is connected via a USB to the computer used for data acquisition and spectral analysis.
The Optrode interface developed on LabVIEW allows users to process, view, and save the necessary data for experiments conducted with the Optrode. It also allows users to monitor the stability of the laser throughout experiments [9].

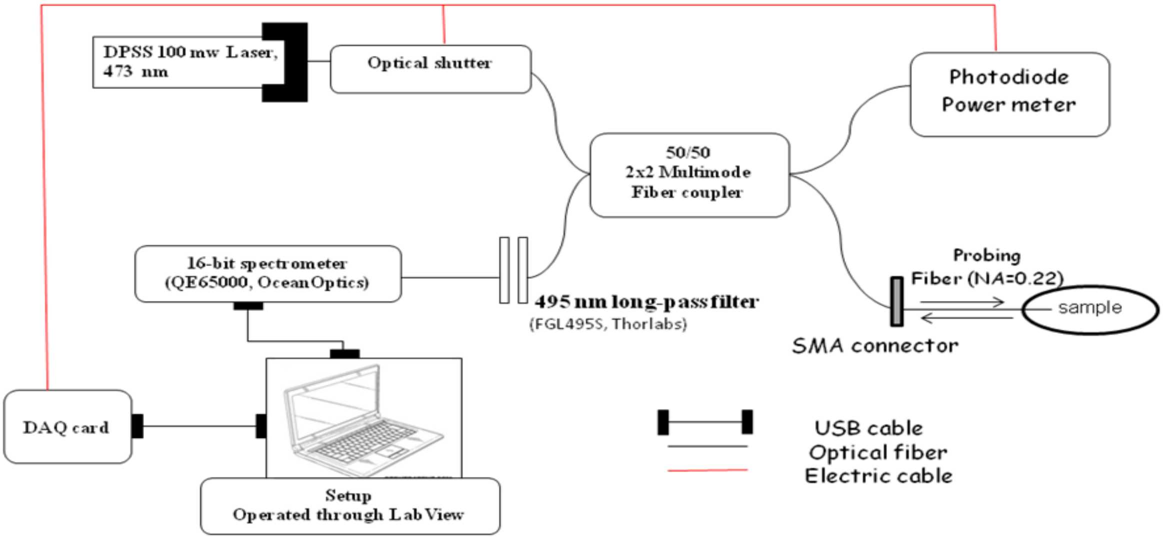


Fig. 1 Schematic diagram of the Optrode set-up [4]

The overarching process of an experiment as controlled through LabVIEW can be split into four main steps:

- Initiating the system
- Power calibration
- Background signal acquisition
- Signal acquisition

Each step has one screen, and the use of each screen is walked through below. The description of each screen is followed by a brief list of suggested changes. The suggestions are collated from observations made on new users during demonstrations of the Optrode, and knowledge of usability heuristics and guidelines [15, 16].


Fig. 2 Existing Shutter Control panel

1. The laser chosen for the rest of the experiment needs to be chosen in the upper left corner, and be visible with any panel throughout the experiment. The default colour is blue, so this control will not need to be changed unless the green laser is used.
2. Before a session begins, the system user must open (turn ON) the shutter for the appropriate laser colour by clicking either the button for Green or Blue, corresponding to the Laser Color control.
3. Any errors that occur with the shutters will be displayed respectively here.
4. The stop button is used for when the user wishes to adjust the hardware positioning and does not want the laser to be emitting light in the process.

First impression:

- The laser colour control is outside tab page, which does not help the user identify it as a control that needs to be used in this (or any) particular step of the experiment.

This control is often not touched, as the default color is blue and all the experiments I have observed all used the blue laser. Because the laser colour control is rarely used, users may get the sequence of events wrong, or forget about modifying the laser colour completely which would be very troublesome. Frederique, as an experienced user, has encountered this issue in the past so it is a likely occurrence, particularly for new users.

- A laser shutter ON button needs to be pressed at the start of a session, but the STOP button does not have to be pressed at the end of a session. The purpose and appropriate moment of use of these controls (particularly the STOP) are not very intuitive. Description labels, and visual separation to differentiate them pre-linguistically, should be applied if there are no friendlier solutions feasible within LAbVIEW constraints.
- Error messages can be moved into dialogues as they will rarely be displayed with an error message to reach a minimalistic design.
- Only the 'Laser Color' and shutter button controls are used at the initiating stage of the session, therefore, it may be more intuitive if these controls are displayed as a dialog that only appears once when the Optrode is launched. The stop button is used in the power calibration stage of the system setup, so it should appear while the user is on that screen rather than on this laser initiating screen.
- As the laser shutter buttons are for the purposes of opening and closing a shutter, it would be more natural if the buttons are changed to Boolean switches which can be clicked to slide open and close, like the action of shutters.

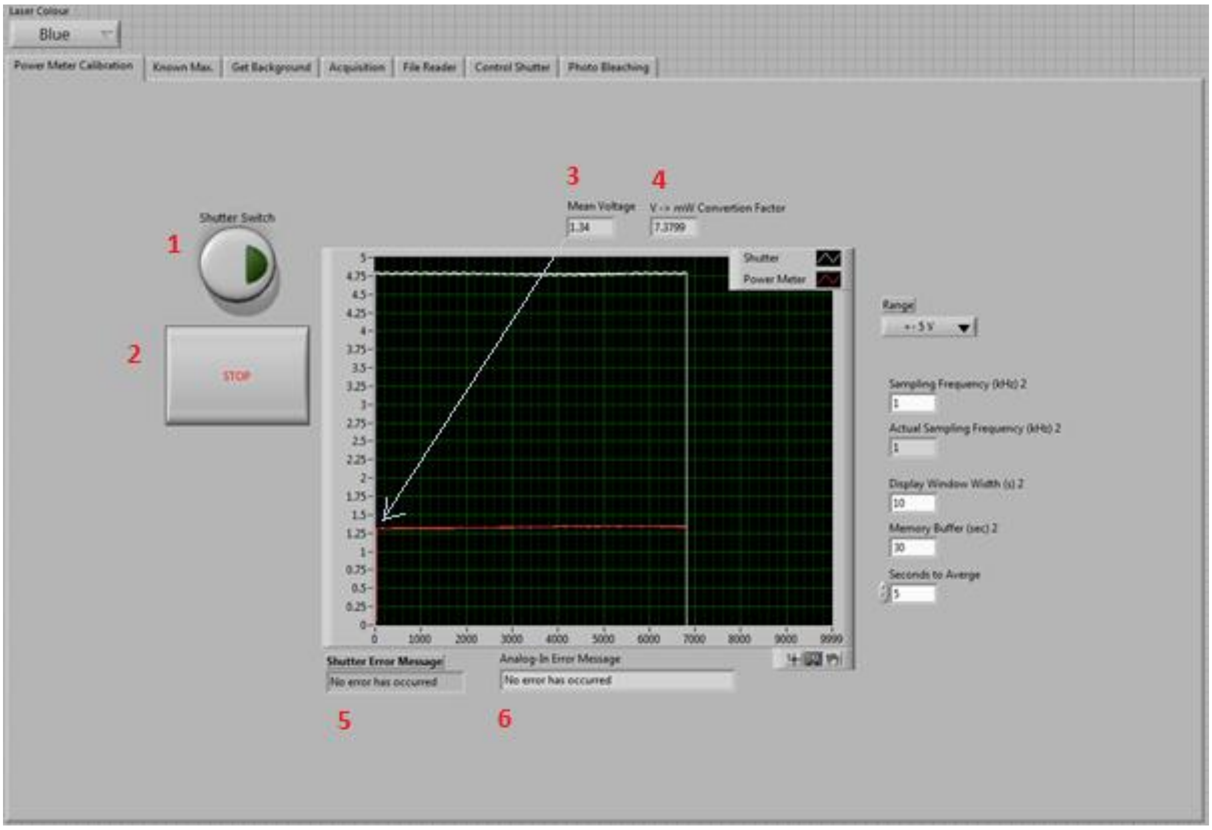


Fig. 3 Existing Power Calibration Panel

Preliminary: Place the Optrode's probe in front of the external power meter.

1. Turn on the laser using the Shutter Switch. This switch should be lit ON during the entire reading.
2. Press run () in LabVIEW control. When the voltage measured appears roughly constant over a period of 5 to 10 seconds, press STOP
   A window will pop up asking for the power to be entered, in which the user inputs the reading displayed by the external power meter.
3. The mean voltage is displayed here.
4. The power, automatically converted from average voltage using an algorithm, is displayed in mW. The expected value is around 10mW.
5. Any error with the shutter will be displayed here.
6. Any error with the data input (voltage) collection will be displayed here.

Note: Any error caused by the analog-in on the DAQ card will result in a dialog pop up requesting that the user restarts LabVIEW.

First impression:

- The best solution to remove the popup asking for the user to input a power reading is by connecting the external power meter to an AC input on the DAQ, and reading that value using LabVIEW. This way, the power value can be recorded without user interference and eliminates potential error, and also simplifies the process for the user. Frederique will be assisting me in implementing this change to the hardware.
- The display is very compacted together, and does not follow the flow of action. The pane should be divided into areas that intuitively lead the flow of use for the user.
- The range can be removed as it is never changed, but the values below it may be necessary so they are to be put in a pane which can be hidden away by default, and opened in the rare occurrence that the values need to be edited.
- In order to reduce clutter, error messages could appear as dialog pop ups.
- Perhaps the calibration could be timed and stop automatically when there is little fluctuation over the past 5 seconds. This will reduce manual labour and make the program easier to use for people unfamiliar with the backend logic of the Optrode.
- IF it is programmatically possible, call restart on LabVIEW within the analog-in dialog rather than requesting for the user to restart the software to reduce manual labour work and increase efficiency.
- Power reading results will be inaccurate if the shutter is closed before STOP is pressed. There should be a prevention mechanism implemented to avoid such an error.

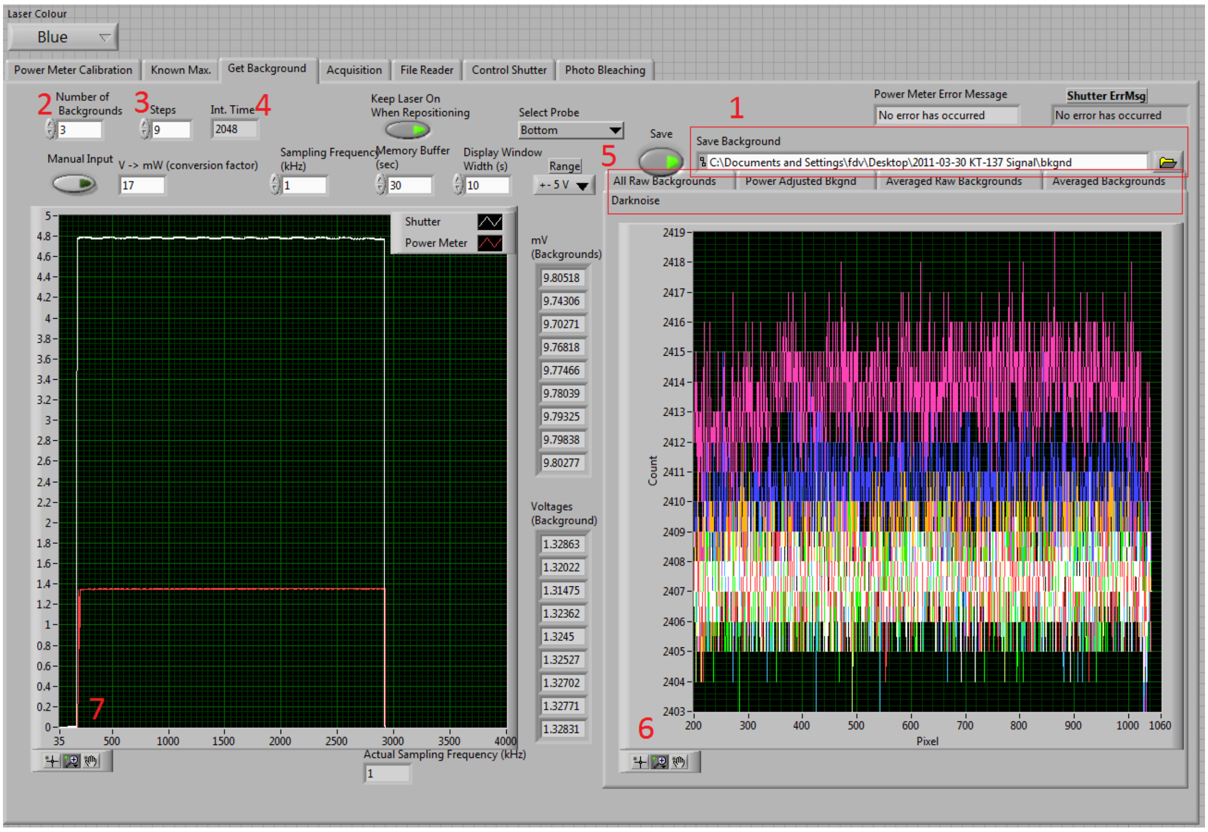


Fig. 4 Existing Background Panel

Preliminary: Insert the fibre into a control sample that is not fluorescent. Ensure that the Save button is lit so acquired data will be saved at the end of acquisition.

1. A location for data file storage can be typed, or explored. When the background samples are acceptable, they will be saved for future reference.
2. The number of backgrounds is preset to 3, and usually does not need to be changed.
3. The step is preset to 9, and usually does not need to be changed. The steps should be set to at least 9, which equates to an acquisition integration time that is unlikely to be reached in an experiment. This way, no matter how many signal acquisitions are actually made, the background noise will be accounted for.
4. Integration time is how long the sample is exposed to the laser light, and is not modifiable. It will increase with every measurement, as the integration time is set to double each time. Once 1 and 2 are set, the user will turn off all lights in the room to

reduce external noise, press run (in LabVIEW control), and wait until background measurements have been taken.

5. User checks background and dark noise plots to determine if the readings are acceptable. If not, hardware fixes will need to be made before running LabVIEW to measure the background noise again.
6. There is one graph per tab. They can be viewed within the Background Panel.
7. This graph appears the same as the power calibration panel, however it shows the shutter and power meter reading as the background readings are made.

Note: The Optrode should be turned on 10-15minutes prior to background being measured. This way, background will be measured at the same temperature as the actual signal acquisition.

First impression:

- Rather than having a save path on the background and acquisition pages, it would be more intuitive to ask for a path when the experiment is first started, and then auto create the file names within the folder through the rest of the experiment. This prompt will need to be reproducible on command, in the case that multiple substances are tested in one sitting.
- The controls "Sampling Frequency", "Memory Buffer", "Display Window Width", and "Range" are not applicable for the purposes of the Optrode hardware set up that I am working on, and may be removed.
- Error messages can be moved into dialogues as they will rarely be displayed with an error message to reach a minimalistic design.
- "mV (Backgrounds)" and "voltages (Background)" and "Actual Sampling Frequency (kHz)" can be moved into a retractable pane which is initially hidden, as they will assist in evaluating whether the Background is correct, however they are often unnecessary.
- The mixed use of plural and singular in the labels mentioned in the previous point is confusing.
- The panel should be arranged in a way that assists the natural flow of use by the user.

- “Number of backgrounds”, “Steps”, “Manual Input” button and “V -> mW (conversion factor)” are never changed so they can be removed from the screen.

## SIGNAL ACQUISITION

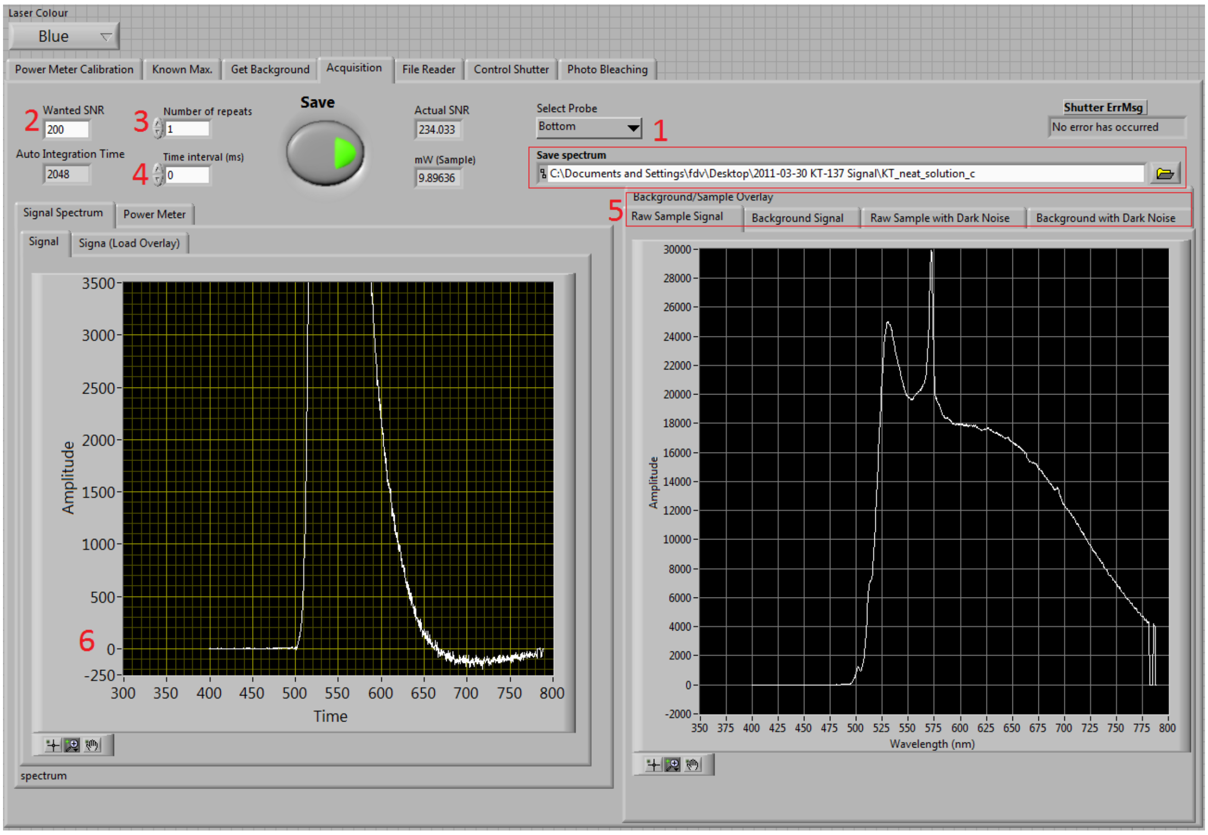


Fig. 5 Existing Acquisition Panel

Preliminary: Clean the fibre to remove any residue, and insert the fibre into the sample for testing. Ensure that the Save button is lit so acquired data will be saved at the end of sample acquisition.

1. User selects a location where the readings will be saved by typing or browsing
2. User sets the SNR. It is usually placed around 10 or 20, but can be increased if the sample is weak and will require a longer acquisition time.
3. Number of repeats is preset to 1, but can be increased.
4. Time interval is preset to 0. It is not usually changed.
5. User can check various adjusted graphs in these tabs.

6. Raw signal can be read from these graphs on the left, and compared with the corrected signals on the right.

First impression:

- The pane should be divided into areas that intuitively lead the flow of use for the user.
- I believe the spelling of “Signa (Load Overlay)” may be unintentionally incorrect.
- The graph display adjustment buttons on the lower left of each graph is misaligned for some panels so the panels appear inconsistent.
- Although the “Wanted SNR” requires some understanding of the logic behind the Optrode, Frederique wishes to keep this control as it is (without abstraction). The label may be re-named to be more intuitive.
- Like the background acquisition pane, it would be better to remove the save path from here, and have it set together with the background path when the experiment is first started.
- Error messages can be moved into dialogues as they will rarely be displayed with an error message to reach a minimalistic design.
- As the Background Signal graph pane is exactly the same as the one on the Background acquisition tab, it can be removed from this pane.
- The probe may need to be changed, so it should be moved into a retractable pane, together with the “Number of Repeats” and “Time Interval”.
- “Actual Integration Time”, “Actual SNR”, and “mW (sample)” are not necessary, and can be removed to reduce clutter and confusion.

LabVIEW [2] is a system-design platform, and a development environment for a visual programming dataflow language developed by National Instruments. The first version of the Optrode [4] was built on LabVIEW 2009 [17] on a 32-bit Windows OS. It uses a USB-1608FS DAQ card [12] produced by Measurement Computing, which must be installed with the Universal Library [13] in order to read and write data to the DAQ card in LabVIEW. The Universal Library also includes a number of functions which are pre-made virtual instruments (VIs) that are essentially external subroutines which assist with coding. The version I am modifying is a newly built Optrode running on the latest LabVIEW 2013 [18], on a 64-bit Windows OS. Since the original code was written and upgraded on the old setup, an updated driver library named ULx [14] has been released. The Universal library is compatible with versions of LabVIEW up to 8.2.1 but not later, and ULx is compatible with versions 8.5 to 2013. Hence, the existing UL code cannot be read, run, or modified on LabVIEW 2013. In order to operate the Optrode on the new operating system, the Optrode code must be translated from using UL to ULx. This new driver library brings about a number of changes in coding, as VIs in ULx are more modular than VIs in UL. This means that each function offered by ULx can be used for multiple purposes, and have different inputs and outputs to what UL functions accepted. Thus, ULx VIs are applied differently within LabVIEW compared to VIs in UL, and can essentially be considered two different languages. LabVIEW also offers a number of core functions that can be used across different devices. Although these functions have not changed, the way they are used in conjunction with UL and ULx functions differ greatly, and so LabVIEW can be considered a third language.

Different versions of LabVIEW can be installed on the same operating system. As a result, the current 64-bit desktop computer has both 32 and 64-bit versions of LabVIEW installed. Theoretically, it is possible to run both 32-bit and 64-bit LabVIEW and even older versions of LabVIEW on the same computer. However, National Instruments removes the previous version of LabVIEW from their website when a new version is released, so to re-install old versions, one must request an installation CD from National Instruments directly. Due to the difficulty of coding in ULx without the ability to run the UL code, I requested a copy of

LabVIEW 2009 which can run UL. However, after a lengthy delay, National Instruments replied that they ‘do not have’ an installation for LabVIEW 2009 that they can send to me.

As LabVIEW and associated hardware devices are used for industrial automation and similar purposes, there are few resources online for troubleshooting and general discussion on its operation. Oftentimes, researching an error will come up with only one or two results which are sometimes not applicable to my problem. In the end, it is more convenient to email National Instruments or Measurement Computing with questions directly, though there is a lengthy wait time of a few days between emails.

The spectrometer developed by Ocean Optics measures the wavelength of light, which is then fed into LabVIEW and used to plot graphs. To correctly interpret the data sent in by the spectrometer, the OMNIDriver and SPAM package [19] by Ocean Optics is installed.

## 2.3 USER INTERFACE AND USABILITY STUDY

A number of research papers have been reviewed in order to identify how usability of the Optrode interface can be improved. The Optrode is a device with a specific purpose, many system constraints, and is only used by a small number of people in a niche market. This BTech project is also under a heavy time constraint, so I believe that employing multiple iterations of user testing will yield the best results in this instance compared to conducting an expert evaluation [20] followed by a user test [21]. The rest of this section will explain the study of usability, and explain past approaches to conducting usability improvement and lessons learnt. Some may also be able to be modified for the purposes of use on the Optrode.

Usability can be split between functional and traditional forms [22]. Functional aspects of usability include navigation, controls, and design of the interface. Traditional aspects of usability can be encompassed by effectiveness, efficiency, and satisfaction in use. As Bevan (2009) [23] explains, effectiveness is defined by accuracy and completeness, while efficiency is defined in the amount of resources expended. Satisfaction, defined as comfort and acceptability of use, is divided into likability (cognitive satisfaction), trust (satisfaction with security), pleasure (emotional satisfaction), and comfort (physical satisfaction). All elements of usability should be considered in a usability study, and can be metrics evaluated on likert scales through which users can express their satisfaction towards the usability of the Optrode interface as they test it.

McCarthy and Wright (2004) [24] explained sense-making as a six step process: anticipating, connecting, interpreting, reflecting, appropriating, and then recounting. With regards to the Optrode, it is particularly important that sense-making is optimised for the user. The phases of connecting and interpreting are particularly crucial for the Optrode as the system is likely to be used by one person at any time, with little assistance available. Connecting refers to the immediate, pre-conceptual and pre-linguistic reaction, which means that the interface needs to be naturally easy to navigate. Interpreting is about what a user understands about a situation, and how they feel about it. In this instance, the interface should be simple to use for any user, but also allow users to adjust specific settings if they wish to do so. This will allow basic users to easily use the system, while not limiting advanced users with preset default settings.

Karapanos et al. (2009) [25] found that the temporality of experience consists of three main forces: familiarity, functional dependency, and emotional attachment. The beautification of a system interface is often not extremely important for long term use compared to the product's ability to serve the user's purpose. This holds particularly true for the Optrode, which is a result-driven product rather than experience-driven product such as a PC game. I will be referencing the usability characteristics by Nielsen [16]: learnability, efficiency of use, recounting, frequency and seriousness of errors, and subjective user satisfaction. These are guidelines widely accepted in the study of usability, and for the basis upon which Nielsen's 10 heuristics [15] were penned. The heuristics listed in Fig. 6 will also serve as a guideline to which the user interface of the Optrode will be evaluated.

1. Match the real world
2. Consistency & standards
3. Help & documentation
4. User control & freedom
5. Visibility of system status
6. Flexibility & efficiency
7. Error prevention
8. Recognition, not recall
9. Error reporting, diagnosis, and recovery
10. Aesthetic & minimalist design

Fig. 6 Nielsen's 10 Heuristics

The process of improving a product through iterations of alternately optimizing and testing is widely accepted in software development [22], and is appropriate in this project. There are no existing products of similar purpose to model or reference, so the only way to confirm whether the design has improved is through user approval and measurement of user tests against KPIs.

National Instruments has published a guide on building effective user interfaces in LabView [26]. These include adding decals to buttons to differentiate buttons with minimal text use, recoloring graphs to increase visibility, customising the run-time menu, moving rarely used functionality into dialogues, organising the items on each page into panes, utilising the 'busy' cursor to keep users informed on the status of the system, editing panel backgrounds, and creating decorations with Microsoft Powerpoint. These tips will be a primary source of reference for making changes on the front panel within the limitations of LabVIEW.

With regards to the process of a usability study, Fierley and Engl (2010) [22] composed a number of key points to be aware of. A usability test can be conducted with an uninterrupted trial of the program first, followed by an interview. For a Business to Business (B2B) product that is mostly used by a single user, it is appropriate to arrange the test room as a study or work place. The context of the test such as location, ambience, and aide are all factors that should mimic a real use situation. During the trial phase, direct observations made on the subject's natural reaction to the system provide valuable insight. Gestures, facial expression, verbal expression, and emotional reaction to the system and situation should be noted.

There are several ways to conduct a successful test. Popularly adopted methods include moderator and observer teamwork, consistent written notation and structure for notes, and post video analysis. Thinking aloud is a technique which is valued for functional evaluation of a system. For the Optrode, it would be effective to encourage and observe spontaneous remarks from the subject as they navigate through the task, rather than asking questions as the subject works through the test case. Their reactions can then be recorded and explored in detail during the interview phase of the test, in order to avoid distracting the subject.

After the system trial period of the usability test, a questionnaire and/or interview is conducted. Common questionnaires include SUMI [27], Mental Effort [28], and Attrak-Diff [29]. Satisfaction of use can be measured through psychometrically designed questionnaires. Some questionnaires are also platform-specific [22], and can be modified effectively for the Optrode. During the interview period of the usability test, a video analysis method can be used to help users reflect and explain how they felt at certain points in the system trial. For the Optrode, the authoritative standardised SUMI [27] will definitely be used. The mental effort scale by Zijlstra [28] can also be included in the questionnaire. Attrak-Diff [29] is used to evaluate the attractiveness and usability of an interactive application for the user, which is entirely applicable to the Optrode and can be used as an additional test to reinforce the results of the other tests. As the Btech 451 project is rather small, and the Optrode is a much specialised professional instrument which is not frequently used, there is no need to test extensively for comfort in use.

As the Optrode is developed and being used in two countries (New Zealand and Belgium), there are best practices to note when conducting usability development on its user interface. As some existing users are overseas, tests may need to be conducted online or through email correspondence. Ideas and mock-ups should be shared early and regularly, as with all rapid recursive development. Bit also, idea clarifying and thought documenting is particularly important where face-to-face meetings are difficult to realise [30].

Throughout the first semester, I witnessed 3 run-throughs of the system. However, despite taking notes and audio recording, I was still very confused about the exact process, and the logic behind it. The code in LabVIEW appears to be hacked together through the past upgrades, and is incredibly complex. It also does not help that the LabVIEW software is not very user friendly. For example, the coding screen is an unlimited large canvas, but the user cannot zoom in or out so only a section of the code can be seen at any one time. Universal Library has a substantial amount of resources to learn from online as it has been used for a long time, however ULx is rather new with few resources, and is therefore harder to learn straight away without knowledge of how to use UL in LabVIEW. However, LabVIEW is a visual programming software and is very different and (personally) convenient as I am a visual learner. The difficulty with the first goal of this project is primarily in learning to program in LabVIEW and understanding the operation of the existing code.

I felt that the front panel control of the Optrode is simple and each control is easy to understand for its purpose within the system. However, it is not visually appealing, and much of the time many controls (such as graph adjustment controls) are not utilised. Label names should be standardized through the interface, taking note of American vs. UK English, abbreviation, spelling, and plurality. My primary concern with regards to usability is in the layout of each tab panel and the order of tabs. The operation of each panel does not follow any flow and appears to be at random, and the use of the system skips between the panels in a random order while certain tabs are never used. This will make it very difficult for a new user of the system to remember the steps for operating the Optrode. Currently, there is a user manual with instructions on how to use the Optrode, but like the explanation guide in chapter 2.1, there is little flow or coherence. At the end of this project, Frederique requests that a new user manual be made for the Optrode, with well-flowing and user friendly instructions.

# 3 OPTRODE DEVELOPMENT, SEMESTER 1

## 3.1 CODE UPGRADE

First, because the new Optrode set up has been moved to a 64 bit Windows operating system, I had to re-install LabVIEW, MCC DAQ driver and Ocean Optics Spectrometer drivers for the 64-bit OS. With this upgrade, I made the decision to upgrade from LabVIEW 2009 (which was UL compatible) to the newest LabVIEW 2013 which only operates with ULx VIs. In order to install the DAQ driver, MCC's InstaCal had to be installed first. InstaCal [31] is an installation, configuration, and test utility developed by MCC. I installed both the Universal Library 6.33 application [13] and the 64-bit ULx for NI LabVIEW 2.05 [14]. Finally, I installed the Ocean Optics Spectrometer OmniDriver + SPAM package [19], because those were the applications that were previously installed on the computer for LabVIEW 2009. Extensive research on how the hardware, software, and drivers relate to each other was conducted prior to making any changes on the computer system.

The first problem that arose in this system upgrade step is that the Universal Library application for MCC DAQ could only be installed in the x86 Program Files folder in C drive. I installed a 32-bit version of LabVIEW 2013 in the x86 Program Files in order for the existing program written with UL to run. The 32-bit LabVIEW is theoretically able to run separately to the 64-bit according to National Instruments [32], however when I attempt to launch the 32-bit version the 64-bit version runs, and the UL Vis create errors. As it turns out, Universal Library is not compatible with LabVIEW 2013 regardless of whether the installation is 32-bit or 64-bit, even though I was able to install Universal Library into the folder without error. Hence, the existing code cannot be run until it has been adapted for ULx.

Another problem I faced is that the dependencies of the Optrode's Vis were previously linked to Universal Library installed within another student's project rather than an independent folder. Consequently, VI dependency issues appeared as a consequence of current library VIs residing in a different location to where they were previously. If the program is moved onto another computer, this issue will occur again unless an independent location for dependency files is specified and consistently used, or dependency files are contained within the folder of the Optrode's code files. Because I feel that it would be excessive to install LabVIEW and its related drivers within the Optrode's program folder as it uses a large amount of memory, I decided to install LabVIEW within Program Files and link all the dependencies there. However, due to UL not being able to run on LabVIEW 2013, the program could not run even though all the dependencies were linked properly. When the program is upgraded to use ULx, I will ensure that the dependencies link to the LabVIEW installation folder so that the same issue won't surface if the program is moved to a new computer.

As mentioned in chapter 2.2, there are few resources online, and few people in New Zealand who could provide help to me with regards to LabVIEW, and all the other components used in the Optrode. As a result, I spent a large amount of time learning about how LabVIEW works, and how the Optrode works. To put the difficulty into context, the Core 1 LabVIEW qualification training costs 2700 New Zealand Dollars [33], and there are four core courses. I worked to overcome this issue by consulting other users of LabVIEW and the MCC 1608FS DAQ card on forums, and watching Youtube tutorials for generic codes. This allowed me to gain a basic understanding of how LabVIEW functions operated, the flow of functions during execution, and how UL functions were used in conjunction with LabVIEW functions. ULx offers a number of sample codes in ULx help, however they require a good understanding of physics to differentiate the different examples as they all looked similar to me. Thus, I made the most of the examples by walking through many of them to understand the general construction of code using ULx.

The last problem I faced is that ULx does not acknowledge the existence of the DAQ card. Therefore, it is not possible to select physical channels and program the source and destination of data. This is the biggest problem, as it has prevented me from re-coding the entire program. Without being able to collect data through the DAQ, it is not possible to test whether the program works. I had looked through all the forums I could find on National Instruments and Measurement Computing, but none of the solutions I found fixed this issue on the Optrode. I have since contacted forum commenters to ask if they could help provide information on how they solved their problem, and also to ask solution providers in the forums for ideas they may have for my issue. I am currently still working on fixing this problem, in contact with two people from Measurement Computing.

## 3.2 USABILITY UPGRADE

Through the first semester, I studied the questionnaires available for professional usability studies, behaviour and fine details to note about conducting a study, and generally how to conduct a usability study. This has been documented in chapter 2.3, and will be the knowledge upon which my user study (roughly outlined in chapter 4) will be formed.

# 4 SEMESTER TWO PLAN OF ACTION

The second part of BTech 451 will focus on action. Firstly, I will work to improve the code. This will include correcting the following issues:

- Laser shutter closes automatically when the laser is not in use (implemented, but not tested)
- Background and power measurements saved in separate respective files (background saving partially done, power saving not done, and none are tested)
- Acquisition data to be saved as a float rather than an integer. (implemented, but not tested)

Secondly, the user interface will be improved in three iterations. Usability studies will be carried out between development iterations to take user feedback into account. The following issues will be addressed, upon request from the project owner:

- Labels are hard to understand and not intuitive
- Pages are messy
- Pages are ordered in a difficult to follow way. The overall operation of the Optrode is unintuitive
- The user guide must be updated to reflect the actual state of the Optrode

Together with Frederique, we have penned a list of KPIs for the usability improvement to measure whether the aim is achieved by the end of the project:

1. An experienced user should be able to run a full experiment (including set up) within 10 minutes.
2. First time users should be able to successfully complete an experiment fully within 15 minutes with the help of the user guide.
3. New users should be able to prepare the Optrode for an experiment within 5 minutes with the help of the user guide.
4. Each of the new functions to be implemented should work as expected.

# REFERENCE

[1] University of Auckland. (2014). Frederique Vanholsbeeck. Retrieved from http://www.engineering.auckland.ac.nz/people/f-vanholsbeeck

[2] National Instruments. (2014). LabVIEW. Retrieved from http://labviewwiki.org/LabVIEW#Learning_LabVIEW

[3] Bogolmony, E., Swift, S., Vanholsbeeck, F. (2013). Total viable bacteria count using a real time all-fibre spectroscopic system. *Analyst*, 2013, **138**, pp. 4112-4119.

[4] Bogomolny, E., Swift, S., Cheng, M., Van Binsbergen, S., Vanholsbeeck, F. (2014). Dead/alive bacteria detection using an all-fibre optical system. Proc. SPIE 8933, Frontiers in Biological Detection: From Nanosensors to Systems VI, 89330F, pp. 1-8.

[5] Patel, A., Bogomolny, E., Cheng, M., Swift, S., and Vanholsbeeck, F. (2013). Sensitivity Improvement of an all-fibre computerized optical fluorescence setup using dual fibre probes. Proc. SPIE 8798, Clinical and Biomedical Spectroscopy and Imaging III, 87980F (June 18, 2013), pp. 1-5.

[6] Chen, A. Y. H. (2011). Development and characterization of fibre-based systems for biomedical imaging (PhD manuscript). University of Auckland, Auckland, New Zealand. https://researchspace.auckland.ac.nz/handle/2292/13361 (accessed online on the 5th August 2014).

[7] Kuo, J. Y. C. (2012). Optimisation of a Real Time All Fibre Spectroscopic Fluorescence Probe (master's thesis). University of Auckland, Auckland, New Zealand. (no online resources, soft copy only).

[8] Patel, A. (2013). Optimisation of an all-fibre spectroscopic system for in situ bacteria monitoring (master's thesis). University of Auckland, Auckland, New Zealand. (no online resources, soft copy only).

[9] Cheng, M. (2014). In-situ fluorescence monitoring of microorganisms using fibre optics (master's thesis). University of Auckland, Auckland, New Zealand. (no online resources, soft copy only).

[10] Hewitt, B. M., Singhal, N., Elliot, R. G., Chen, A. Y. H., Kuo, J. Y. C., Vanholsbeeck, F., Swift, S. (2012). Novel Fiber Optic Detection Method for in Situ Analysis of Fluorescently Labeled Biosensor Organisms. Environmental Science & Technology, 46 (10), pp. 5414-5421.

[11] Chen, A. Y. H., Vanholsbeeck, F., Tai, D.C.S., Svrcek, M., and Smaill, B. H. (2010). Time-resolved all fibre fluorescence spectroscopy system. Optical Fibres and Sensors for Medical Diagnostics and Treatment Applications X, 7559, pp. 23-28.

[12] MCC. (2014). USB-1608FS. Retrieved from http://www.mccdaq.com/usb-data-acquisition/USB-1608FS.aspx

[13] MCC. (2014). Universal Library. Retrieved from http://www.mccdaq.com/daq-software/universal-library.aspx

[14] MCC. (2014). ULx for NI LabVIEW. Retrieved from http://www.mccdaq.com/daq-software/universal-library-extensions-lv.aspx

[15] Nielsen, Jakob. (1994). Usability Engineering. (pp. 115-148). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[16] Nielsen, Jakob. (1992). The Usability Engineering Life Cycle. Computer, 25 (3), 12-22.

[17] National Instruments. (2014). NI LabVIEW 2009. Retrieved from http://www.ni.com/support/lv2009.htm

[18] National Instruments. (2014). LabVIEW 2013 Features and Changes. Retrieved from http://zone.ni.com/reference/en-XX/help/371361K-01/TOC1.htm

[19] Ocean Optics. (2014). OmniDriver Programming Guide. Retrieved from http://www.oceanoptics.com/technical/OmniDriver_Programming_Manual.pdf

[20] Usability. (2014). Heuristic Evaluations and Expert Reviews. Retrieved from http://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html

[21] Usability. (2014). Usability Testing. Retrieved from http://www.usability.gov/how-to-and-tools/methods/usability-testing.html

[22] Fierley, R. and Engl, S. (2010). User Experience Methods and Games: Lessons Learnt. Proceedings of the 24th BCS Interaction Specialist Group Conference, pp. 204-210. Swinton, UK: British Computer Society.

[23] Bevan, N. (2009). Extending Quality in Use to Provide a Framework for Usability Measurement. Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009, pp. 13-22.

[24] McCarthy, J. and Wright, P. (2004). Technology as Experience. Interactions – Funology, 11 (5), 42-43.

[25] Karapanos, E., Zimmerman, J., Forlizzi, J., Martens, J.B. (2009). User Experience Over Time: An Initial Framework. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp 729-738.

[26] National Instruments. (2012). 10 Tips and Tricks for Building Effective User Interfaces in LabVIEW. Retrieved from http://www.ni.com/white-paper/14131/en/

[27] Kirakowski, J. (1996). The Software Usability Measurement Inventory: Background and Usage. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. WeerdMeester (Eds.), Usability Evaluation in Industry, pp. 169-176. Great Britain: CRC Press.

[28] Zijlstra, F.R.H. & Van Doorn, L. (1985). The construction of a scale to measure perceived effort. Delft, The Netherlands: Department of Philosophy and Social Sciences, Delft University of Technology.

[29] Hassenzahl, M. (2003). The thing and I: Understanding the relationship between user and product. In M. A. Blythe, A. F. Monk, K. Overbeeke, & P. C. Wright (Eds.), *Funology: From usability to enjoyment*, pp. 31-42. Norwell, MA, USA: Kluwer Academic Publishers.

[30] Yiu, C. (2013). UX Design with International Teams: Challenges and Best Practices. Proc. CHI '13 Extended Abstracts on Human Factors in Computing Systems, pp. 2333-2336.

[31] MCC. (2014). InstaCal. Retrieved from http://www.mccdaq.com/daq-software/instacal.aspx

[32] National Instruments. (2014). LabVIEW 64-Bit vs. 32-Bit Applications FAQ. Retrieved from http://digital.ni.com/public.nsf/allkb/71E9408E6DEAD76C8625760B006B6F98

[33] National Instruments. (2014). LabVIEW Core 1. Retrieved from http://sine.ni.com/tacs/app/overview/p/ap/of/lang/en/pg/1/sn/n5:selfpacedonline,n24:12725/id/1582/