User Guide for the MUX

MultiplexÔ

2019-07-20

Contents

Chapter 1

Reasons for the MUX

The first prototype of the system was built in 2010. The idea was that current water quality sensors are expensive and they are designed to be installed to obtain high frequency data at a particular station. In the field of hydrology, agrohydrology, ecohydrology, and limnology, to only name a few, there is a tight coupling between concentration variations in time and those in space. In other words, concentrations may vary by several folds within minutes and meters. To be able to track constituents and their fate over short distances (~< 50 m), one would ideally need not ***one*** but ***several*** sensors... But then the price of such monitoring system would be prohibitive in the vast majority of cases. Not to mention that maintenance and chances of failure would be multiplied by the number of sensors deployed.

A more promising idea was to create a system that would be able to provide access to high enough frequency data for ***multiple points*** in the vicinity of ***one sensor***. This solves the affordability problem as one sensor only is necessary, and it also solves the problem of multiplying maintenance on multiple sensors. Our solution is the MultiPlexed, MultiPoint Sampler or MPS, which we have now decided to call the *MUX*, because it is just easier to say! And since you have the instrument with you now, you probably already have a pretty good idea of how it works.

Our solution is thus to bring water to the sensor, rather than the opposite. And once this idea became a promising solution, then expanding the 'bringing of water to the sensor' to multiple points was a natural extension of the idea. All the MUX is, is a peristaltic pump for pumping and purging, a bunch of three-way solenoid valves (we chose 12 for now) that dictate which sampling point is activated, and a micro controller system to control and log all the MUX activities in synchrony with the sensor (Figure 1.1). The MUX sequentially pumps water from all the desired point to the sensor, and once all the points have been sampled, the sequence starts over again.

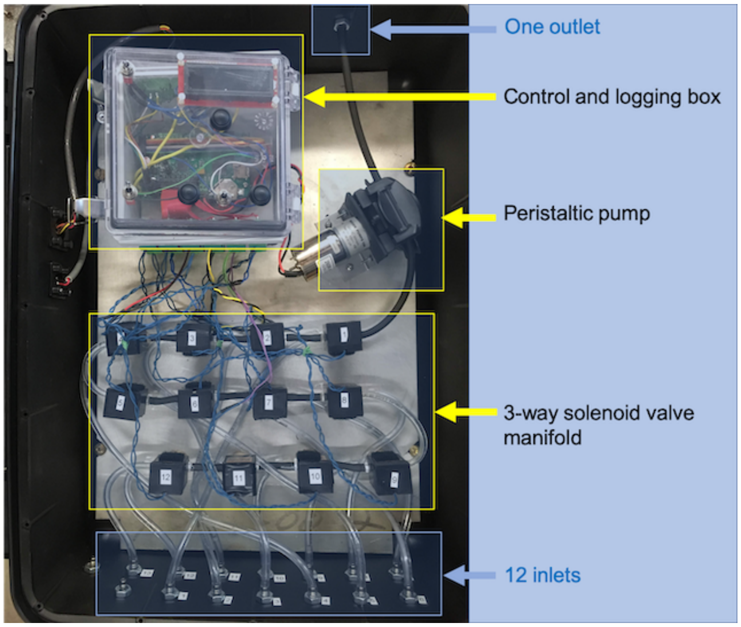


Figure 1.1: General Layout of the MUX, including a control box, a peristaltic pump, 12 3-way solenoid valves, 12 inlets, and one outlet

In theory, it is very simple. In practice, we have discovered that it takes a lot of attention to details to have a system that is robust enough to work over long periods of time reliably. We feel the version we have now is robust enough for others to use, although we are quite aware that there is still room for improvement, and we are dedicated to keep improving our system. We have published the details of the design and the performance of the MUX in (2016).

1.1 The versions available

Right now, we have three versions available. One specifically dedicated to work with the S::CAN field spectrophotometer called Spectro::lyser, one version that works with any sensor, and, a third version coupled with a synchronous syringe based sampler designed to sample rather small volumes of water at very low pumping rates (~around 1 ml/min).

1.2 Downloads

- Download the SpectroMUX.pdf here

- Download the epub version here

Chapter 2

How to hook things together

2.1 The different parts

The version you have received works with a S::CAN spectro::lyser, a Con::nect box, a power supply cable connected between a battery or a AC to DC transformer, a 12V battery (+ solar panel for long term deployment) or a AC to DC transformer, a 6-pin and a 2-pin cables provided with the MUX, and the MUX itself. You need all these parts for things to work properly.

We use only 4 of the cables of the 6-pin cable. This cable is used to catch the cleaning or valve signal from the Spectro::lyser, which we later use trigger the MUX sampling sequence, and, to obtain the measurement data through the RS485 connection and communication protocol.

We use the 2-pin cable to power the MUX. Honestly, we could have used the 6-pin cable and use all 6 cable to do, power, cleaning signal, and the data transfer, but by default, we like to keep a cable dedicated to power.

2.2 Connecting with the Con::nect Box

All wires to the MUX are connected to the Con::nect Box. To use with the MUX, we use the left panel of the Con::nect Box (Figure 2.1).

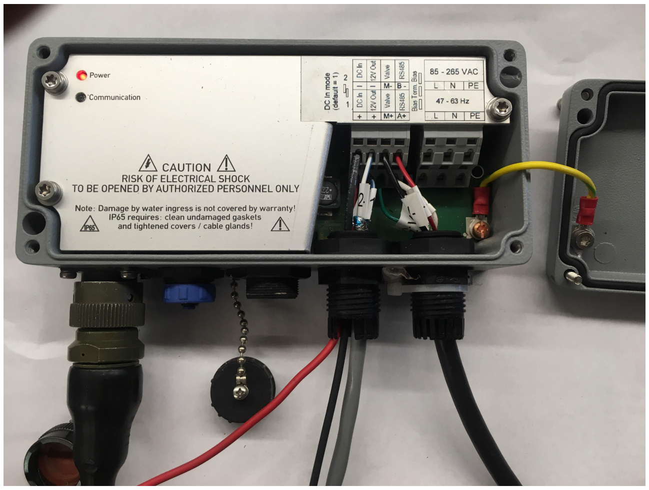


Figure 2.1: picture of the connecting capabilities inside the Con::nect Box

2.2.1 Powering the Con::nect Box

At the most left side, the user connects the wires either connected a 12V battery, or to an AC to DC adapter, which delivers 12V. This is referred to as 'DC In' on the Con::nect Box label (Figure 2.1). By convention, the positive wire is normally red and the negative wire, black. The black or negative wire is at the top or right of the label, and the red or positive wire is at the bottom (not really visible on Figure 2.1). Make sure you use at least 14 gauge for these wires, to be on the safe side.

All the Con::nect Box connections are done using spring-loaded terminal. You need to use a fine screwdriver and insert it on the top part of the terminal (yellow arrows on Figure 2.2), gently push or rotate down to open the spring-loaded connection located below (magenta arrows on Figure 2.2). While open, insert the wire as deep as you can to obtain the best and secure connection, while still leaving the plastic sheathing out. Make sure the free wires are not longer than 4 mm, however, otherwise free wires will be stay out and that is not ideal. After you have inserted the wires, make sure you pull on the wire. If it comes out easily, do it again until it is hard to pull on it. Also, always insert the bottom wires first.

The three other connection types include '12V out', 'Valve', and 'RS485', and we use all three with the MUX.

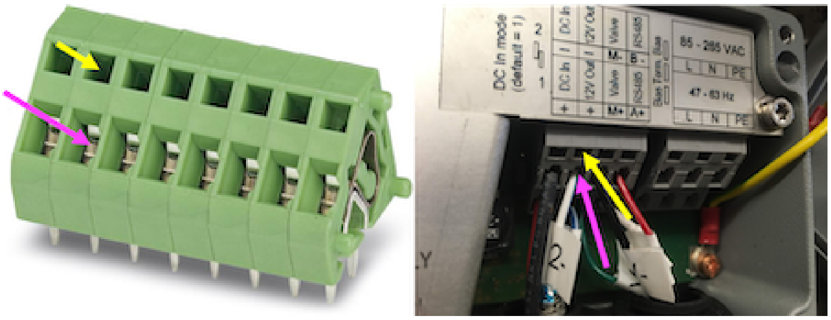


Figure 2.2: Spring-loaded connections for the Con::nect Box

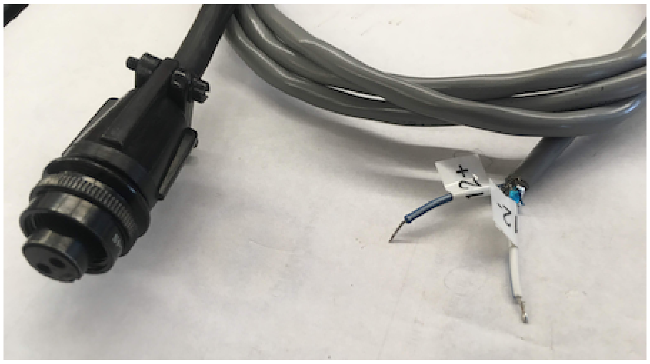


Figure 2.3: 2-pin cable to provide 12V DC power to the MUX, with the blue wire as ’+’ and the white wire as ’-’

2.2.2 Powering the MUX

The MUX is powered by the ‘12V out’ connection. The Con::nect Box provides a source of 12V power to other instruments that might be connected to it and we are taking advantage of this system to power the MUX. For that you will use the 2-pin cable as illustrated in Figure 2.3 below.

The blue wire corresponds to the positive or “+” and the white wire corresponds to the negative or ‘-’. As illustrated in Figure 2.4, insert the blue “+” wire at the bottom or left connection, and the white “-” wire at the top or right connection of the ‘12V out’ terminal.

2.2.3 Connections to Trigger the MUX

The Con::nect box has a connection through which it can send a 12V signal for a specified amount of time. Actually, it is the spectro::lyser that sends the 12V. Originally, this was used to open an air valve to send a burst of compressed air

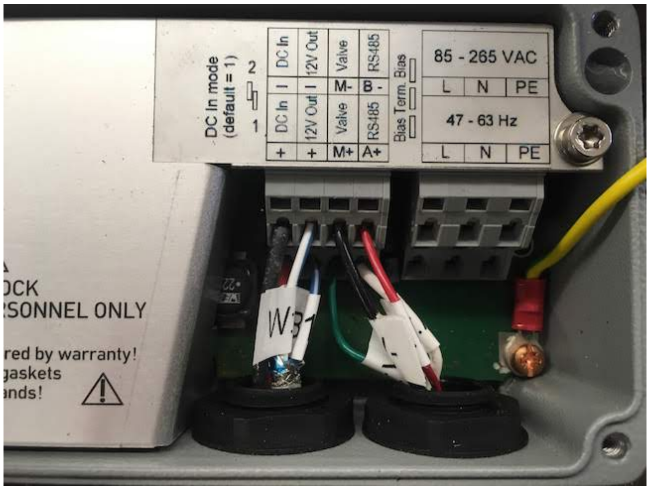


Figure 2.4: 2-pin cable to provide 12V DC power to the MUX, with the blue wire as '+' and the white wire as '-'

to clean up the optics. Later it has been used to activate an automatic brush, also to clean the optics. We are using this feature to start a sampling sequence from the MUX. When at idle, the MUX is listening to such a 12V signal, and upon receiving this signal the MUX starts its sampling sequence.

For this we use the black and green wires of the 6-pin cable (Figure 2.5).

As illustrated in Figure 2.4, insert the green “M+” wire at the bottom or left connection, and the black “M-” wire at the top or right connection of the ‘Valve’ terminal.

2.2.4 Communications between the MUX and the Spectro::lyser

With this version, the MUX essentially takes control of the Spectro::lyser, by giving impoing the measurement intervals, the 12V trigger signal duration, and its time before measurements, sending it into ‘logger mode’, and writing the fingerprint values onto an SD card. For this we use the RS485 protocol and connections available through the Con::nect Box.

As illustrated in Figure 2.4, insert the white “A+” wire at the bottom or left connection, and the red “B-” wire at the top or right connection of the ‘RS485’ terminal.

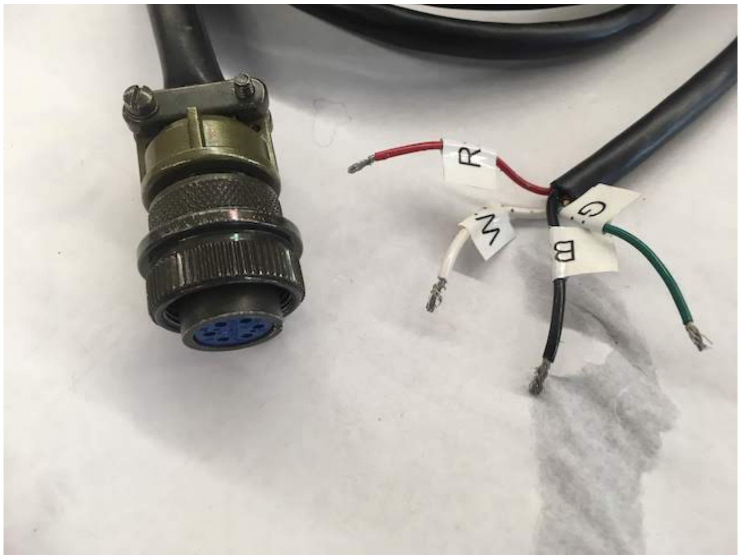


Figure 2.5: 6-pin cable to trigger a sampling sequence to the MUX, using the black (M-) and green (M+) wires, and, to communicate between the MUX and the spectro::lyser using the white (A+) and the red (B-) wires through the RS485 protocol

2.2.5 The MUX external plugs

The 6-pin and 2-pin wires are plugged in the MUX box itself via the “Auxilliary” and “12 VDC” plugs located on the hinge side of the MUX box as illustrated in Figure 2.4 below.

2.2.6 Control box plugs

Inside the MUX, the control box (Figure 1.1) itself is connected via plugs to the 2-pin and 6-pin wires, and to the solenoid valves. The peristaltic pump is directly connected via its wires thanks to a screw terminal. Normally, there is no need to ever touch at any of these plugs. But should the user need it, it is possible to undo the control box from the rest, and or to change a solenoid valve and easily replace it if needed.

For illustration, the MUX power plugs, as well as the valve and the RS485 plugs on the control box. The 12V coming from the Con::nect Box are plugged in the control box as illustrated in Figure 2.7 below.

The Signal input and the communication wires coming from the Con::nect Box are plugged in the control box as illustrated in Figure 2.8 below.

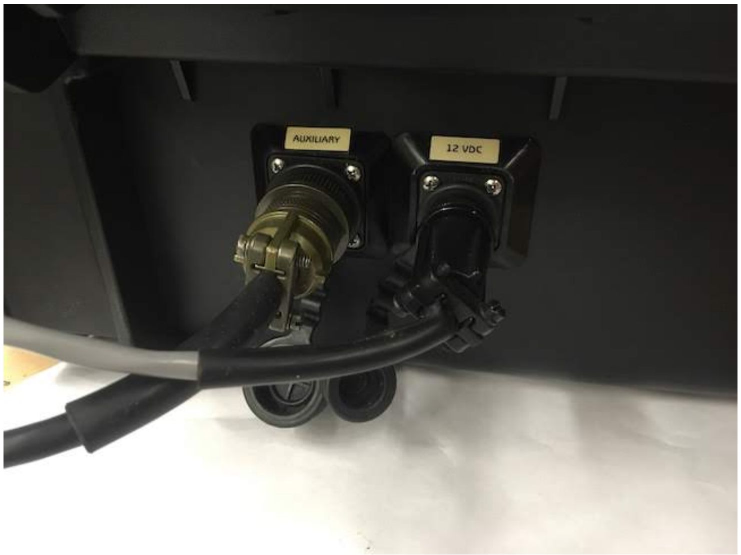


Figure 2.6: External plugs for the 2-pin and 6-pn cables

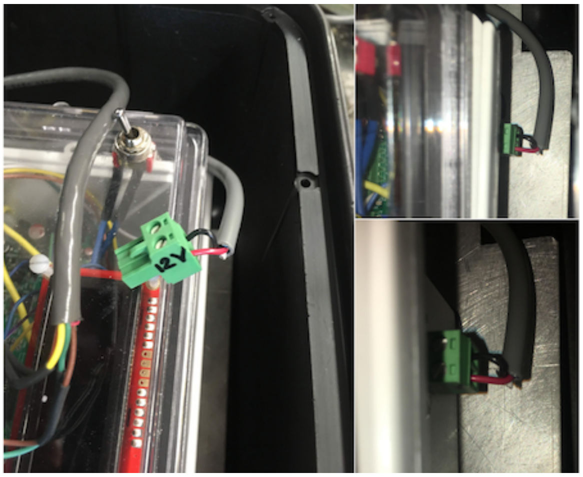


Figure 2.7: 12 V Connection between the control box and the external plug

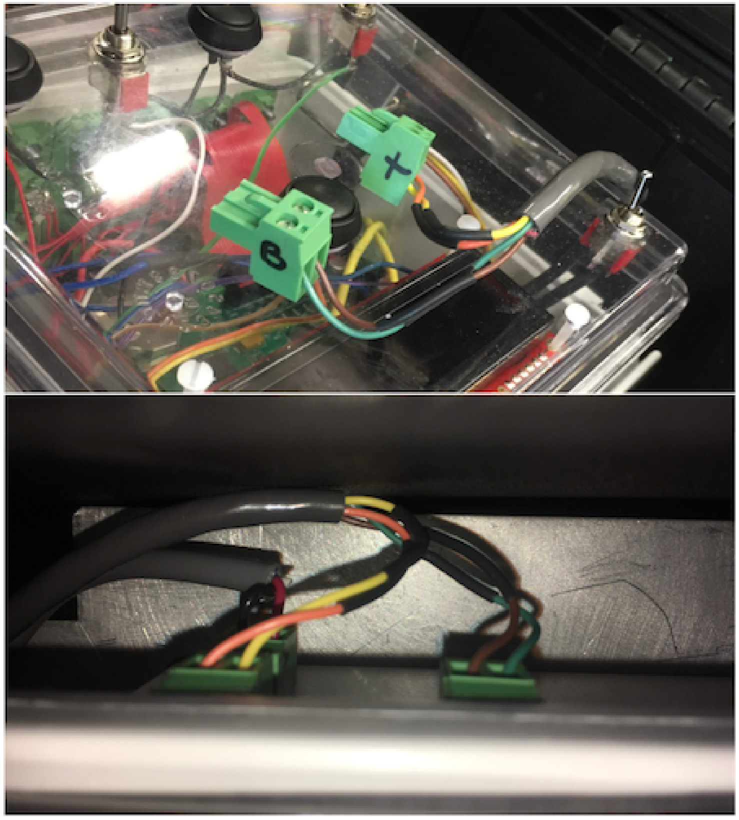


Figure 2.8: 12 V Connection between the control box and the external plug

2.3 How it works

peristaltic pump

change clock time

Manual pump

General functioning on/off

2.4 How to start

2.4.1 SD card

2.4.2 Config file

2.4.3 Rinsing options

2.5 How to change the clock

2.6 How to pump manually

2.7 How to trigger the MUX manually

2.8 The files you get

2.9 Trouble shooting

Chapter 3

The MUX-sensor functioning

3.1 The general design principles for the MUX

An exert from our first paper (Birgand et al., 2016) summarizes the philosophy behind the MUX:

> To remain affordable, the system uses a single high frequency automatic water quality probe as the central analytical instrument to which water from different sampling sites (hereafter, point sources) are pumped via [the MUX]. Our design criteria when constructing the [MUX] system were: (1) to have the capacity of obtaining hourly or sub-hourly samples for measuring multiple point sources, (2) to be able to pump water from the point sources to the probe to overcome at least 3 m of head difference, (3) to be able to run on 12 volts (V) direct current (DC) power for field deployment, and (4) that the coupled [MUX]-water quality probe system functioned entirely automatically. An acceptable compromise for these criteria was to design [the MUX] that enabled sampling from up to 12 point source sites located within [30] m of the central probe. [...].
>
> As the in situ field spectrophotometer can only collect one measurement at a time on fixed time intervals, we designed and built [the MUX] to sequentially pump and purge water from each point source to the probe, in synchrony with the probe measurements, and cycle through the sequence of measurements to obtain at least hourly time resolution of data collection at each source. To maintain the water quality probe's capability of measuring small suspended particle concentrations, we chose 3.18 mm internal diameter flexible tubing as

> a sampling conduit of water from the point source to the probe, fitted with 1.5 mm mesh screens at the source [...]. Consequently, the coupled [MUX]-water quality probe system is well suited for applications where the source sampling volume is not limited and does not affect the process or system studied.

3.2 The different parts of the MUX-sensor system

3.2.1 Sampling ports and rinsing solutions

In continuous flow laboratory analytical equipment, the flow in the tubing is unidirectional, and the measuring cuvette is rinsed by the carrier solution and color reagent between samples to essentially eliminate cross-contamination between samples. In our case, although we are using the Spectro::lyser as an *in situ* laboratory instrument, there is no carrier fluid for rinsing between samples. To minimize cross contamination between samples, the idea we developed was to reverse flow and purge the instrument cuvette from its water instead. This is the reason for using a peristaltic pump. Now water can be purged either to the sampling port or to a port dedicated to waste.

This addresses the cross contamination, but not the fouling that may occur as is often the case on the instrument optics. For this reason, the MUX has an embedded rinsing procedure where it is possible to use acid or other solutions for rinsing. The overall general MUX-sensor setup is summarized in Figure 3.1 below.

In Figure 3.1 above, n sampling ports are activated, while two of the ports are dedicated to a *rinsing solution 1* and a *rinsing solution 2*. The MUX has 12 sampling ports and a 13th port for waste if desired. The number or rinsing solutions used, therefore reduces the number of available ports for measured points. In the case of Figure 3.1 above, a maximum of 12 - 2 = 10 ports remain available for point sampling.

Sampled water from any sampling point, including the rinsing solutions, can be purged to any port, including the dedicated waste port (port 13 in the MUX box), or the same port used for sampling, or any other port of the user's choice.

3.2.2 Flow through cuvette

At the instrument, water is pumped through a flow-through cuvette. This allows to pump water beyond the cuvette and allow for rinsing of the whole manifold and of the cuvette with the new sample. We have found that a good practice to minimize cross contamination is to pump water at least 4 times beyond the

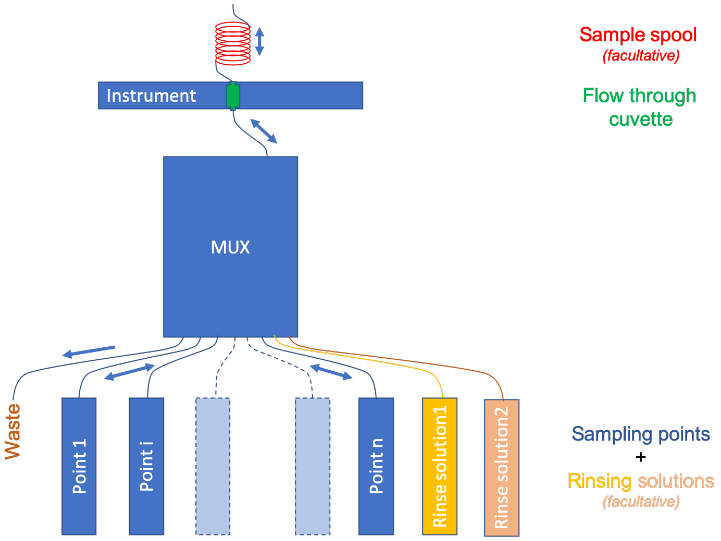


Figure 3.1: The MUX in relationship with the sampling points, rinsing solutions, the instrument and its flow-through cuvette, and a sample loop

volume of the cuvette. For use with the Spectro::lyser with a 35 mm opening, we recommend to use the cuvette 1-Q-10/SBTX2-8/10X20 from Starnacell®. It has a 10 mm path length, is made of quartz, which lets the UV light through, and has a large collar, which make cleaning much easier.

3.2.3 Post-cuvette sample spool

Eventually during a measurement campaign, it is necessary to compare the instrument measured concentrations to those of the laboratory. For this, it is sometimes a good idea to add a *sampling spool* made of tubing curled together. We recommend that the spool allows for large enough a volume corresponding to the one needed for lab analyses, but small enough such that the initial water that has served to rinse the MUX and the cuvette be beyond the spool such that when purging only 'clean water' be used for lab sampling.

3.3 Sampling sequences

In the MUX *config.txt* file (see further details later), the user specifies the desired sampling interval. During initialization, the MUX puts the Spectro::lyser in *logger mode*, which means that the instrument will, with no live connection to a computer, automatically take measurements at the indicated time interval.

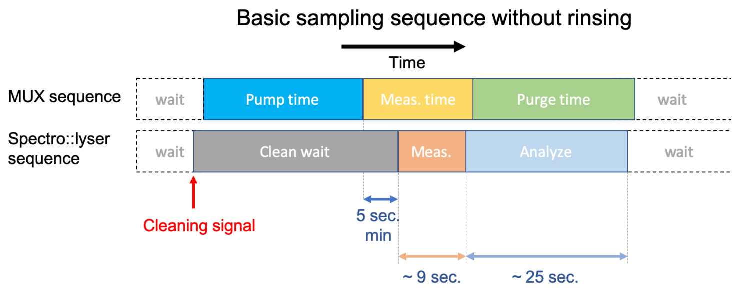


Figure 3.2: The MUX in relationship with the sampling points, rinsing solutions, the instrument and its flow-through cuvette, and a sample loop

The Spectro::lyser sequence of actions (Figure 3.2) is thus to first send a 12V signal measurable at the 'valve' output in the Con::nect box. Again, this is the way the Spectro::lyser deals with fouling. After a given time, referred to as *clean wait* in the *config.txt* file and in Figure 3.1, the instrument takes a measurement. A measurement consists in two series of light bursts accompanied by a series of beeps. There are about 9 seconds between the first and the last beep. After that, it takes the Spectro::lyser about 25 seconds to analyze its measurement and make available its absorbance fingerprint and its calculated concentrations (Figure 3.2).

3.3.1 Pump and measurement times

The idea with the MUX is to have water to be analyzed in the measuring chamber or cuvette when the Spectro::lyser does its measurement. Once the MUX 'hears' the *cleaning signal* from the Spectro::lyser (Figure 3.2), it starts what we refer to as its *sampling sequence*. The basic *sampling sequence* consists in pumping, waiting for the Spectro::lyser measurement, and purging the MUX.

The MUX starts pumping for a *pump time* defined in the *config.txt* file for each valve or sampling port. The *pump time* corresponds to the time it takes for water to go from its sampling point to the instrument cuvette, to which one must add the number of seconds necessary to rinse the equivalent of at least 4-5 times the cuvette volume (Birgand et al., 2016). For some applications, there is no limit on the available volume of water to be analyzed. In these cases, one might want to add 5-6 seconds more on the *pump time*. For other applications, the available volume might be limited, and it might be necessary to limit the pump time to the strict minimum prescribed here.

The *pump time* and the *clean wait* times are thus intrinsically linked. The *clean wait* time has to be equal to the pump time plus at least 5 seconds to let water in the cuvette 'rest' and let all micro air bubbles to stop moving to obtain a reliable and consistent reading by the Spectro::lyser. Keep in mind that there

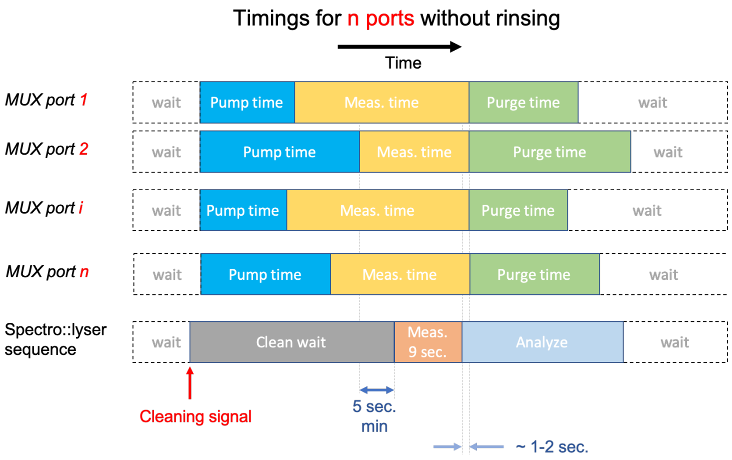


Figure 3.3: Adjusting the *pumping times* and *measurement times* of the MUX to match the sequence of events of the Spectro::lyser

are about 1-2 seconds delay between the *cleaning signal* and the time when the pumping starts (Figure 3.2).

There are two important additional things to realize here:

1. In reality, the *pump time* may vary from sampling point to the next as it is preferable to pump the least amount of time per port to save energy and pump tubing. The *pump time* for each port is defined in the *config.txt* file. However, the *clean wait* time corresponds to a parameter from the Spectro::lyser, and the same value applies to all sampling points. As a result, the *clean wait* time **must correspond to the longest *pump time* + 5 seconds** (Figure 3.3).

2. Because it always takes the same amount of time between the *cleaning signal* and the last beep of the Spectro::lyser measurement, and, because the *pump time* may vary from one sampling point to the next, the (on the MUX side) **sum of *pump time* + the MUX *measurement time*** should also be the same and be equal to (on the Spectro::lyser side) **the *clean wait* time + the 9 sec. of measurement time + at least 2 seconds past the last measurement** (Figure 3.3).

As a result, the *pumping time* and the *measurement time* from the MUX must be adjusted (in the *config.txt* file) accordingly for each sampling point as illustrated in Figure 3.3

3.3.2 Purge port and times

The MUX *purge time* corresponds to the time needed to purge, at least, the cuvette and the MUX valve manifold (Figure 1.1). The purge valve and the *purge time* is defined in the *config.txt* file. Depending on the application, the user might want:

- to purge the line that was just used for sampling all the way, in which case, we recommend to use the *pump time* and add 3-4 seconds to make sure the line is rid of as many droplets as possible.
- to purge to a *waste* port (port 13 is the dedicated port for that but any port or valve can be used). In that case, the *purge time* may be much shorter as in all likelihood the line to waste will be much shorter. The idea here is to save energy and the lifetime of the peristaltic pump tubing.

3.3.3 Sampling sequence with automatic rinsing

Our experience with the Spectro::lyser has suggested that a rinsing system be put in place to keep optics or cuvette fouling at a minimum. For this we have designed an embedded automatic *rinsing sequence* that takes place right after the purging of the *basic sampling sequence*. For this it is possible in the *config.txt* file to define one or several *rinsing sequence*. For each *rinsing sequence*, one defines in the *config.txt* file, the port from which the rinsing solution will be sampled and will be purged, the *pumping time*, and the *purging time* for the defined rinsing sampling and purging ports, and the number of seconds the rinsing solution will stay in the cuvette.

We have found that when water is sampled from areas where reduced conditions prevail, chemical fouling in the form of iron and manganese precipitate may occur. In these cases, we have found that rinsing using 5% HCl acid solution or an oxalic acid solution (~15 mg L^{-1}) left in the cuvette for 1 min greatly reduced the amount of fouling. We have also found that in some cases rinsing with DI water is enough. In this case, the amount of time DI water stays in the cuvette does not matter as much and can be just a few seconds. We recommend that a *rinsing sequence* that uses acid should be followed by a second rinsing sequence that uses DI water.

The timings for the *rinsing sequence* are defined in the *config.txt* file and the number of *rinsing sequences* are also defined in the same file. The *rinsing sequence(s)* automatically take(s) place after the *purging* of the *basic sampling sequence*, and does not require a *cleaning signal* from the Spectro::lyser to take place as illustrated in Figure 3.4 below.

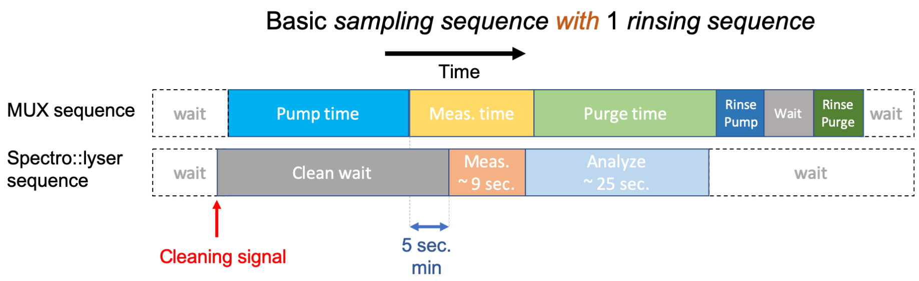


Figure 3.4: The MUX *basic sampling sequence* followed by an automatic *rinsing sequence*

3.4 Sampling cycle

A *sampling cycle* is the name we have given for a series of *sampling sequences* that occur on a recurrent basis. For example, if the user has ports 1, 2, 4, and 5 being sampled, a *sampling cycle* will have a *sampling sequence* corresponding to port 1, followed by a *sampling sequence* corresponding to port 2, etc., until the last sampled port, in our example, a *sampling sequence* for port 5. This would make for the basic *sampling cycle*. However, we do recommend that the user adds in the *sampling cycle*, either air or DI water as an easily recognizable 'tracer' measurement in the time series of fingerprint from the Spectro::lyser. In our example, a *sampling cycle* would thus be: "port 1, port 2, port 4, port 5, air" or "port 1, port 2, port 4, port 5, DI". This *sampling cycle* repeats itself indefinitely (Figure 3.5). The example below only considers 4 sampled points, because there was not enough space for illustration. However, with one rinsing solution, and using air as a measurement tracer, a *sampling cycle* can have up to 11 sampled ports (12 minus the port used for rinsing). With two rinsing solutions, and using air as a measurement tracer, a *sampling cycle* would have up to 10 sampled ports (12 minus the 2 ports used for rinsing). In the latter case, one of the rinsing solutions would likely be DI water, which could also be used as the measurement tracer instead or air in the *sampling cycle*. In that case, a *sampling cycle* would still have up to 10 sampled ports (12 minus the 2 port used for rinsing, knowing that DI is used both for rinsingn and as a measurement tracer).

The advantage of inserting air or DI water in the *sampling cycle* is that in addition to the aid it provides to identify each cycle in the fingerprint time series, it also gives an estimate of the fouling of the cuvette or the optics. To insert and *air* measurement in the *sampling cycle*, the user can use 'port 13' in the *config.txt* file as it is used for waste and is therefore open to the atmosphere. To insert an *DI* measurement, the user has to have a dedicated port sampling DI from an container with enough DI volume to cover the field visit intervals.

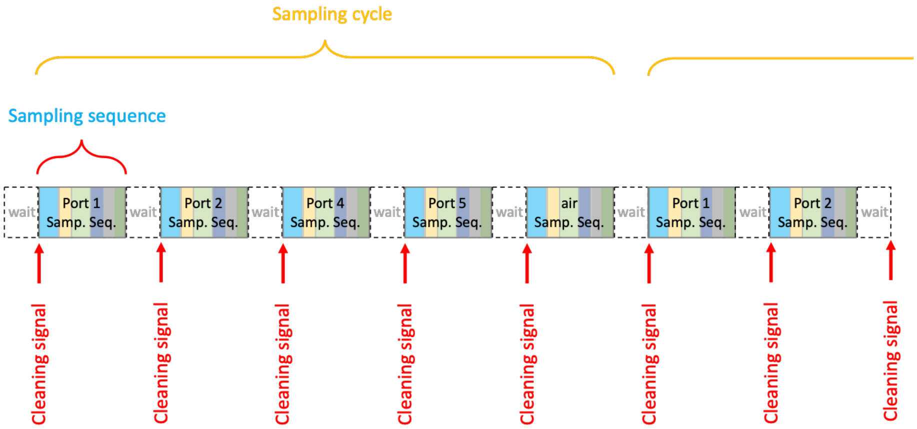


Figure 3.5: The MUX *sampling sequences* for ports 1, 2, 4, and 5, included into *sampling cycles* in which an *air* measurement has been inserted

Chapter 4

Programing the MUX

4.1 The SD card in the control box

All the instructions for programing the MUX are consigned in the *config.txt* file, itself located on the micro SD card. The SD card can be retrieved by opening the control box. One may use the macro SD card adapter provided to read the SD card on a computer (Figure 4.1).

4.2 The *config.txt* file

All the instructions for the sampling sequences are provided in the *config.txt*. A typical basic *config.txt* file looks like in Figure 4.2 below. In this particular example, there are 12 valves activated in a *sampling cycle* ranked in increasing order from 1 to 12. The order of the valves can be completely interchanged and the *sampling cycle* will go followin the prescribed order.

4.2.1 Programming the MUX *sampling sequence* without automatic rinsing

The commands for the *sampling sequences* for each *sampling cycle* are given in the first 6 lines of the *config.txt* file (Figure 4.3). Each variable between `<...>` defines the name of the variable used in the program, and all numbers between `[...]` define the array of values to be used. The values in one array correspond to all the values necessary for a *sampling cycle*.

- `<VALVES PUMPING>` defines the list and its order of the valves or port which will be activated during *pumping* of the *sampling cycles*. In the example

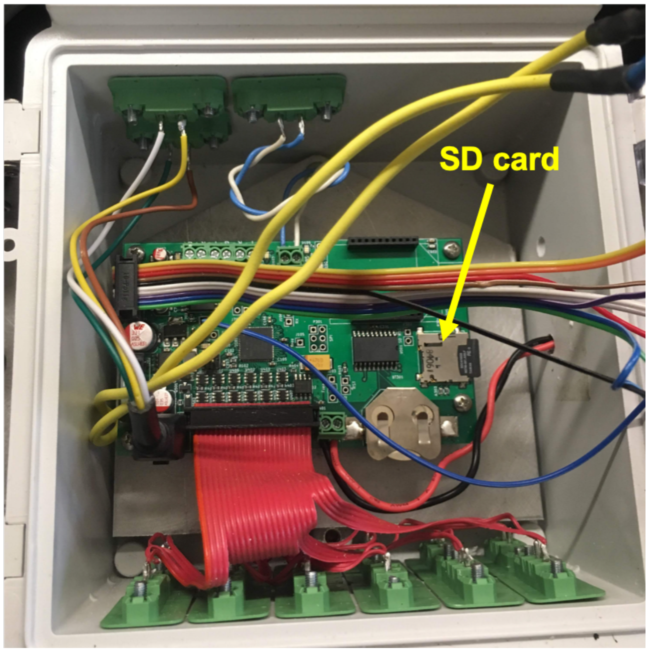
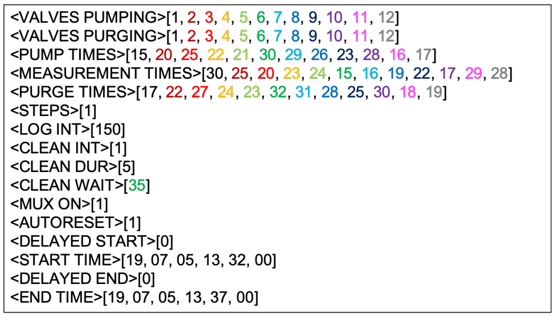
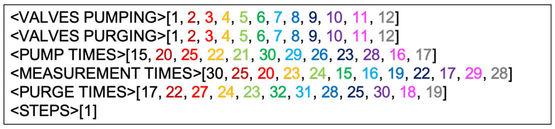


Figure 4.1: Location of the SD card in the control box of the MUX



```
<VALVES PUMPING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<VALVES PURGING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<PUMP TIMES>[15, 20, 25, 22, 21, 30, 29, 26, 23, 28, 16, 17]
<MEASUREMENT TIMES>[30, 25, 20, 23, 24, 15, 16, 19, 22, 17, 29, 28]
<PURGE TIMES>[17, 22, 27, 24, 23, 32, 31, 28, 25, 30, 18, 19]
<STEPS>[1]
<LOG INT>[150]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[35]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 05, 13, 32, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 05, 13, 37, 00]
```

Figure 4.2: Typical *config.txt* file where 12 ports are activated, with no automatic rinsing



```
<VALVES PUMPING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<VALVES PURGING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<PUMP TIMES>[15, 20, 25, 22, 21, 30, 29, 26, 23, 28, 16, 17]
<MEASUREMENT TIMES>[30, 25, 20, 23, 24, 15, 16, 19, 22, 17, 29, 28]
<PURGE TIMES>[17, 22, 27, 24, 23, 32, 31, 28, 25, 30, 18, 19]
<STEPS>[1]
```

Figure 4.3: The commands for the MUX sampling sequence timing are provided in the first 6 lines of the *config.txt* (12 ports are activated, with no automatic rinsing)

of Figure 4.3, the *sampling cycles* inclues all valves from 1 to 12 in that order.

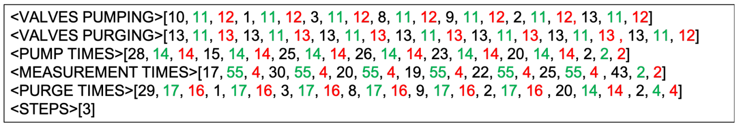
- `<VALVES PURGING>` defines the list and its order of the valves or port which will be activated during the *purging* phase of each *sampling sequence* of the *sampling cycle*. In the particular example of Figure 4.3, all water is purged back to *its source* as the same valve is used during *pumping* and *purging*.
- `<PUMP TIMES>` defines the array of values in seconds that will be used for the *pumping* in each *sampling sequence* corresponding to the *pumping* valve activated.
- `<MEASUREMENT TIMES>` defines the array of values in seconds that will be used for each *sampling sequence* and corresponding to the activated valve.
- `<PURGE TIMES>` defines the array of values in seconds that will be used for the *purping* in each *sampling sequence* corresponding to the *purging* valve activated.
- `<STEPS>[1]` means that there is no automatic rinsing. Technically, it defines the fact that the program will read each array and each cleaning signal will activate the next `[1]` (one) value in the array. In other words, the program will read the next value in the array to be used upon a *cleaning signal* (more details later).

For illustration purposes, the text of the *config.txt* file have been colored. In the current example (Figure 4.3), in the *sampling cycle*, the first *sampling sequence* starts with pumping from valve 1 for 15 seconds, waits for 30 sec for the Spectro::lyser to do its measurement, then purge to through the same valve 1 for 17 seconds, i.e., 2 seconds more than pumping to remove as many drops as possible. Because the `<STEPS>` is set to 1, the MUX waits for the next Spectro::lyser *cleaning signal* to start its next *sampling sequence*.

By analogy, the 7th *sampling sequence* starts with pumping from valve 7 for 29 seconds, waits for 16 sec for the Spectro::lyser to do its measurement, then purge to through the same valve 7 for 31 seconds, i.e., 2 seconds more than pumping to remove as many drops as possible.

- Notice that for all activated valves, `<PUMP TIMES> + <MEASUREMENT TIMES>` is always equal to 45 seconds, and the 45 seconds corresponds to the longest *pumping time* (30 sec, corresponding to valve 6 in the example), plus 5 seconds of no pumping for all gas bubbles to settle, plus 9 seconds for the Spectro::lyser measurement to take place. The current setting suggests that the MUX would start its purging not even one second after the last beep of the Spectro::lyser measurement... That would work, but honestly there is advantage to do that. So adding several seconds more in the measurement timings would probably be more judicious and safer to make sure water is still in the cuvette for the full measurement.

- Also notice that the `<PUMP TIMES> + <MEASUREMENT TIMES>` is equal to the `<CLEAN WAIT>` (Figure 4.2) value minus 5 seconds corresponding to



```
<VALVES PUMPING>[10, 11, 12, 1, 11, 12, 3, 11, 12, 8, 11, 12, 9, 11, 12, 2, 11, 12, 13, 11, 12]
<VALVES PURGING>[13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 13 , 13, 11, 12]
<PUMP TIMES>[28, 14, 14, 15, 14, 14, 25, 14, 14, 26, 14, 14, 23, 14, 14, 20, 14, 14, 2, 2, 2]
<MEASUREMENT TIMES>[17, 55, 4, 30, 55, 4, 20, 55, 4, 19, 55, 4, 22, 55, 4, 25, 55, 4 , 43, 2, 2]
<PURGE TIMES>[29, 17, 16, 1, 17, 16, 3, 17, 16, 8, 17, 16, 9, 17, 16, 2, 17, 16 , 20, 14, 14 , 2, 4, 4]
<STEPS>[3]
```

Figure 4.4: Programing for 6 ports and an air measurement, purging to waste, and two automatic rinsing

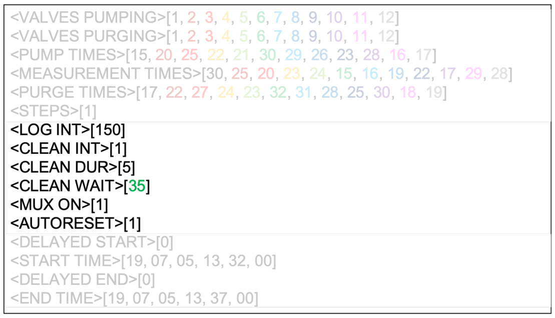
the minimum of time for bubbles to settle for the longest pumping time (more details below)

4.2.2 Programming the MUX *sampling sequence* with two automatic rinsing, and a waste port

The next example illustrates more complicated options. The user wants to use points 10, 1, 3, 8, 9, 2 in that order in a *sampling cycle* and wants to add an *air tracer measurement*. The user does not want to purge water back to the sampling point, but rather, wants to purge all water to a waste port, which incidently can be used for lab sampling purposes. The user also wants to use two automatic rinsing, the first one with acid where acid would stay in the cuvette for 55 seconds to leave enough reaction time, and the second one with DI to rinse the acid off.

The MUX could be programed as illustrated in Figure 4.4 below.

- Notice that `<STEPS>` value is set to 3. This means that the MUX *sampling sequence* will be activated by the *cleaning signal* every 3 values of the arrays (the text in black in Figure 4.4). In between, the green and red values, are automatically activated during the *sampling sequence*, and correspond here to the two rinsing desired. The green values corresponds to the values for the acid rinse, and the red values correspond to the DI rinse.
- The activated sampling ports (other than rinsing) of the *sampling cycle* are defined in the `<VALVES PUMPING>` array and appear in black in Figure 4.4. Their list is thus 10, 1, 3, 8, 9, 2, and 13. The last one, 13, corresponds to the waste port open to the atmosphere, which means that it can be used for 'air tracer measurement'.
- The port used for purging of the sampled points (other than rinsing) also appears in black in the `<VALVES PURGING>` array, and is always 13.
- The pump times, measurement times, and purge times for the sampled points are illustrated in black, respectively, in the `<PUMP TIMES>`, `<MEASUREMENT TIMES>`, and `<PURGE TIMES>` arrays (Figure 4.4)
- The acid rinse will be pumped and purged from and to port 11 everytime. There often no need to purge acid to waste. Although, the low strength of the acid would not damage the environment, it is preferable to purge back to its bottle to save acid. The dilution by water droplets will not



```
<VALVES PUMPING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<VALVES PURGING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<PUMP TIMES>[15, 20, 25, 22, 21, 30, 29, 26, 23, 28, 16, 17]
<MEASUREMENT TIMES>[30, 25, 20, 23, 24, 15, 16, 19, 22, 17, 29, 28]
<PURGE TIMES>[17, 22, 27, 24, 23, 32, 31, 28, 25, 30, 18, 19]
<STEPS>[1]
<LOG INT>[150]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[35]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 05, 13, 32, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 05, 13, 37, 00]
```

Figure 4.5: The parameters in the *config.txt* to pilot the Spectro::lyser

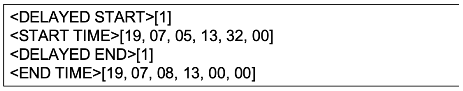
affect acid strength and anti-fouling efficacy.

- To leave acid in the cuvette for 55 seconds, the user add 55 sec. in the `<MEASUREMENT TIMES>` array.
- The DI water rinse is always pumps from port 12 in the example from a designated DI water jar. In the example, the DI water is purged to the waste port 13. This implies that a lrge enough DI water container be used to have enough between field service intervals.
- The amount of time DI water stays in the cuvette does not matter as much as acid here and a value of 4 seconds has been chosen in the example.
- During the air measurements, the pumping and purging times are kept to a minimum (2 sec.) to minimize energy spending and peristaltic pump wear. The measurement time (black value) is set to 43 seconds to stay in sequence with the other *sampling sequences* although it could have been shortened as well.

4.2.3 The Spectro::lyser values in the *config.txt* file

The next five lines in the *config.txt* file give instructions to the Spectro::lyser on the values to use when going into the *logger mode* (Figure 4.5).

- `<LOG INT>` defines the interval in seconds, at which the Spectro::lyser will take a measurement and logs its value. It also corresponds to the interval at which the Spectro::lyser sends a *cleaning signal*, which in the end corresponds at the time interval between two *sampling sequences*. The time interval on each sampled point will be the recurring interval of a *sampling cycle*.
- `<CLEAN INT>` defines the number of *logging intervals* or measurements from the Spectro::lyser there are before it sends a *cleaning signal*. This does not apply to our use, as we want to have a *cleaning signal* sent everytime the Spectro::lyser takes a measurement. ALWAYS set it to 1.
- `<CLEAN DUR>` defines the number of seconds the 12V cleaning signal coming from the Spectro::lyser is on.



```
<DELAYED START>[1]
<START TIME>[19, 07, 05, 13, 32, 00]
<DELAYED END>[1]
<END TIME>[19, 07, 08, 13, 00, 00]
```

Figure 4.6: Example of delayed start on the *config.txt* file

- `<CLEAN WAIT>` defines the number of seconds between the beginning of the *cleaning signal* and the first beep of the Spectro::lyser measurement. **The duration of this value is linked to the *pumping times*, and must be equal, at least, to the *longest* pumping time of the *sampling cycle*, plus 5 seconds to leave time for air bubbles to settle**.
- `<MUX ON>` has to be set at 1. It corresponds to a feature obsolete with the current version of MUX. `<AUTORESET>` this option defines whether the user wants the system to autorest at midnight everyday. We recommend to use this option (set to [1]), as it allows the MUX to clean up its memory. If the user prefers not to use this option, then set to [0].

4.2.4 Delayed start

In many situations, it might be of advantage not to start sampling when the users are in the field, but rather start at a later time. Similarly, the user might want to stop the system before being able to come to the field to retrieve data. This what the delayed start and end are for. If the user does not want the delayed start and end, set `<DELAYED START>` and `<DELAYED END>` to [0].

When the user wants a delayed start, set **both** `<DELAYED START>` and `<DELAYED END>` to [1]. The start time is defined in `<START TIME>` as [YY,MM,DD,HH,mm,SS]. When the user does not want the system to end until the next field visit, set the `<END TIME>` to several years later. An example is given below in Figure 4.6.

4.3 Examples of *config.txt* files

4.3.1 Sampling cycles with no rinsing

- 12 sampling ports in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 order.
- same purging ports
- Five minutes *sampling sequence* intervals
- 60 time resolution for each sampled point

```
<VALVES PUMPING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<VALVES PURGING>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
<PUMP TIMES>[15, 20, 25, 22, 21, 30, 29, 26, 23, 28, 16, 17]
```

```
<MEASUREMENT TIMES>[30, 25, 20, 23, 24, 15, 16, 19, 22, 17, 29, 28]
<PURGE TIMES>[17, 22, 27, 24, 23, 32, 31, 28, 25, 30, 18, 19]
<STEPS>[1]
<LOG INT>[150]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[35]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 05, 13, 32, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 05, 13, 37, 00]
```

4.3.2 Sampling cycles with one rinse

- 4 sampling ports in 1, 3, 5, 7 order, no tracer measurement (not recommended)
- one automatic (e.g., DI water) rinse on port 2
- same purging ports
- Three minutes *sampling sequence* intervals
- 12 time resolution for each sampled point

```
<VALVES PUMPING>[1, 2, 3, 2, 5, 2, 7, 2]
<VALVES PURGING>[1, 2, 3, 2, 5, 2, 7, 2]
<PUMP TIMES>[10, 5,10, 5,10, 5,10, 5]
<MEASUREMENT TIMES>[12, 3, 12, 3, 12, 3, 12, 3]
<PURGE TIMES>[10, 3, 10, 3, 10, 3, 10, 3]
<STEPS>[2]
<LOG INT>[180]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[12]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 06, 06, 08, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 06, 13, 37, 00]
```

4.3.3 Sampling cycles with one rinse

- 4 sampling ports in 1, 3, 5, 7 order, one air tracer measurement (recommended)

- one automatic (e.g., DI water) rinse on port 2
- All purging on port 13
- Three minutes *sampling sequence* intervals
- 15 time resolution for each sampled point

```
<VALVES PUMPING>[1, 2, 3, 2, 5, 2, 7, 2, 13, 2]
<VALVES PURGING>[13, 2, 13, 2, 13, 2, 13, 2, 13, 2]
<PUMP TIMES>[10, 5,10, 5,10, 5,10, 5, 2, 2]
<MEASUREMENT TIMES>[12, 3, 12, 3, 12, 3, 12, 3, 2, 2]
<PURGE TIMES>[10, 3, 10, 3, 10, 3, 10, 3, 2, 2]
<STEPS>[2]
<LOG INT>[180]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[12]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 06, 06, 08, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 06, 13, 37, 00]
```

4.3.4 Sampling cycles with one rinse

- 4 sampling ports in 10, 1, 3, 8, 9, 2, and one air (13) tracer measurement (recommended)
- two automatic, e.g., acid or port 11, followed by DI water rinse on port 12
- All purging on port 13, but for acid back to port 11
- Because of the two rinses, the interval between two *sampling sequences* has to be a bit longer, in fact longer than the summation of the (pumping time + measurement time), the acid (pumping, stay in cuvette, purging) times, and the DI water (pumping, stay in cuvette, purging) times. That sum is equal to 164 seconds. So technically, one could set the log interval to 180 seconds. However, in practice, this would mean that the MUX would be pumping and purging a lot of the time, drawing lots of power and wearing out the peristaltic pump tubing. The user must also assess whether 30 min or hourly data is in fact plenty enough.
- Five minutes *sampling sequence* intervals
- 35 time resolution for each sampled point

```
<VALVES PUMPING>[10, 11, 12, 1, 11, 12, 3, 11, 12, 8, 11, 12, 9, 11, 12, 2, 11, 12, 13
<VALVES PURGING>[13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 13, 13, 11, 1
<PUMP TIMES>[28, 14, 14, 15, 14, 14, 25, 14, 14, 26, 14, 14, 23, 14, 14, 20, 14, 14, 2
<MEASUREMENT TIMES>[17, 55, 4, 30, 55, 4, 20, 55, 4, 19, 55, 4, 22, 55, 4, 25, 55, 4 ,
<PURGE TIMES>[29, 17, 16, 1, 17, 16, 3, 17, 16, 8, 17, 16, 9, 17, 16, 2, 17, 16 , 20,
<STEPS>[3]
```

```
<LOG INT>[300]
<CLEAN INT>[1]
<CLEAN DUR>[5]
<CLEAN WAIT>[33]
<MUX ON>[1]
<AUTORESET>[1]
<DELAYED START>[0]
<START TIME>[19, 07, 06, 06, 08, 00]
<DELAYED END>[0]
<END TIME>[19, 07, 06, 13, 37, 00]
```

Chapter 5

The files saved on the SD card

5.1 The MUX log files

All activities from the MUX are saved on a file named *YYMM-MUX.txt*, with YY for the year, and MM for the month. A typical example of the content of the *MUX.txt* file is illustrated below:

```
2019/7/4 10:32:36, 10, Forward, 10, 20, 5, Automatic - Start!
2019/7/4 10:33:06, 8, Reverse, 10, 20, 5, Automatic
2019/7/4 10:33:12, 2, Forward, 3, 15, 3, Automatic
2019/7/4 10:33:30, 2, Reverse, 3, 15, 3, Automatic
2019/7/4 10:35:35, 1, Forward, 15, 10, 8, Automatic
2019/7/4 10:36:00, 8, Reverse, 15, 10, 8, Automatic
2019/7/4 10:36:09, 2, Forward, 3, 20, 3, Automatic
2019/7/4 10:51:36, 10, Forward, 10, 20, 5, Automatic - Start!
2019/7/4 10:52:06, 8, Reverse, 10, 20, 5, Automatic
2019/7/4 10:52:12, 2, Forward, 3, 2, 3, Automatic
2019/7/4 10:52:17, 2, Reverse, 3, 2, 3, Automatic
2019/7/4 10:52:20, 7, Forward, 3, 2, 4, Automatic
2019/7/4 10:52:25, 3, Reverse, 3, 2, 4, Automatic
2019/7/4 10:53:36, 1, Forward, 15, 15, 8, Automatic
2019/7/4 10:54:06, 8, Reverse, 15, 15, 8, Automatic
2019/7/4 10:54:15, 2, Forward, 3, 2, 3, Automatic
2019/7/4 10:54:20, 2, Reverse, 3, 2, 3, Automatic
2019/7/4 10:54:23, 7, Forward, 3, 2, 4, Automatic
```

On each line of the file, the first value is the time stamp of any pumping (*Forward*) or purging (*Reverse*) action. The next value is the valve or port that is activated. The next value is whether the pump pumped forward or purged for purging. The next three values correspond to . The next value tells whether the MUX actions were automatic, or whether the pumping and purging were done manually. When the MUX is started by turning the on/off button, or when the MUX is automatically reset at midnight, these actions are flagged by a '- Start!' at the end of the line.

This log file is very useful when trouble shooting and to corroborate information stored on the *.fp* file.

5.1.1 The fingerprint files

The fingerprint files are a copy on the SD card of the fingerprint (.fp) stored by the Spectro::lyser. Until very recently, there was very limited memory space on the Spectro::lyser, leaving about three days of autonomy for 3-min measurements... As a result, the MUX automatically interrogates the Spectro::lyser after each *sampling sequence* and asks the Spectro::lyser to give the last absorbance spectrum stored in its memory, and appends it at the end of an *.fp* file on the SD card.

An example of the such *.fp* file is illustrated below. The first line defines which Spectro::lyser is used. The second line defines the Date/time, then status (OK or not), and then the wavelengths at which the absorbances are measured. In our case here, from 200.00 to 750.00 nm. The next lines provide the Spectro::lyser time stamp, the status of the probe, the absorbance values corresponding to each wavelength, AND, at the end of the line, the valve used for pumping and the time stamp from the MUX. The `...` replace all the 200 and some intermediate values on each line.

```
17240013_50_0x0101_spectro::lyser_RIV100NFV2
Date/Time   Status_0    200.00  202.50  ... 745.00  747.50  750.00
2019.07.04  10:51:59    Ok  42.7532 41.6454 ... nan nan nan 10  19/7/4 10:52
2019.07.04  10:51:59    Ok  42.7532 41.6454 ... nan nan nan 1   19/7/4 10:54
2019.07.04  10:53:59    Ok  42.8591 41.7240 ... nan nan nan 3   19/7/4 10:56
```

There is an identified bug, which we have not resolved satisfactorily on an automatic basis, yet. However, when aware of it, it can be easily resolved manually. At the end of its sampling sequence, the MUX inquires about the last measurement stored by the Spectro::lyser. In the meantime, in the Spectro::lyser sequence of actions (Figure 3.2), the time for analysis of the measured absorbance takes about 25 seconds. If the inquiring time, the Spectro::lyser has not finished analyzing its results, then the absorbance spectrum available on the Spectro::lyser memory is NOT the current measurement, but rather the *previous* measurement. This is illustrated in the *.fp* exert above. The fingerprint time stamp and values are exactly the same in third and fourth line of the file, although the MUX time stamp at the end of the lines have changed. This is because on the 10:54 measurement, the Spectro::lyser was still analyzing its measurement, and the available fingerprint on memory was the previous one. It is relatively easy to manually reassign the correct fingerprint to the correct port.

Now, when the end of the *sampling sequence* occurs **after** analysis of the Spectro::lyser, then the fingerprint available in the memory is the one corresponding to the MUX time stamp. In these cases, the Spectro::lyser fingerprint and MUX time stamps are synchroneous.

Chapter 6

How to start

6.1 Make all necessary connections

Needless to say, the MUX-sensor system will not work unless the connections are done in the Con::nect Box! Follow The instructions in part 2.

6.2

Bibliography

Birgand, F., Aveni-Deforge, K., Smith, B., Maxwell, B., Horstman, M., Gerling, A. B., and Carey, C. C. (2016). First report of a novel multiplexer pumping system coupled to a water quality probe to collect high temporal frequency in situ water chemistry measurements at multiple sites. *Limnol. Oceanogr. Methods*, 14(12):767–783.