1 – External devices

In many electrochemical experiments, it is often required to control external devices like stirrers, water baths, sample changers, etc… While most of these external devices can be controlled directly, using dedicated hardware or software, it is convenient to allow NOVA to control these devices directly, during an electrochemical experiment.

NOVA supports a number of communication protocols, which can be used to control external devices. The following communication protocols are available:

- Metrohm Serial Bus (MSB)
- TTL triggers (DIO)
- Recommended Standard 232 (RS232)

Scope of the tutorial

The aim of this tutorial is to explain how to setup and use external devices in NOVA. Particular attention is given to Metrohm devices compatible with this version of the software. DIO triggering and the RS232 protocol is also explained at the end of the tutorial.

2 – Metrohm Serial Bus (MSB)

The MSB interface is the standard interface for Metrohm dosing devices, sample processors and stirrers. It is present on most of the current Metrohm devices. Please note that the Autolab PGSTAT does not have MSB ports that can be used to control MSB devices. Instead, a USB controlled Metrohm device providing MSB ports must be used (see next section for more information).

2.1 – Metrohm devices support

This version of NOVA provides support of the following Metrohm devices:

- 800 Dosino
- 801 Magnetic stirrer
- 802 Rod stirrer
- 803 Titration stand with stirrer and pump
- **814 USB Sample processor**
- **815 Robotic USB sample processor**
- **846 Dosing interface**
- **858 Professional sample processor**
- 786 Swing head

Note: the instruments indicated in bold lettering in the list above are controlled through USB and provide MSB ports that can be used to control other supported Metrohm instruments.

Metrohm devices can be controlled in NOVA, with or without the Autolab potentiostat / galvanostat connected to the computer.

2.2 – Metrohm drivers installation

In order to control the supported Metrohm devices, the required Metrohm USB drivers must be correctly installed on the computer. The drivers are installed during the installation of NOVA. Make sure that NOVA is installed on the computer before connecting Metrohm devices.

When a Metrohm device is connected to a USB port for the first time, the Found new hardware window should appear. When prompted to do so, select the *Install the software automatically* option and click the next button (see Figure 1).

Note: the drivers are automatically installed on Windows Vista and Windows 7.

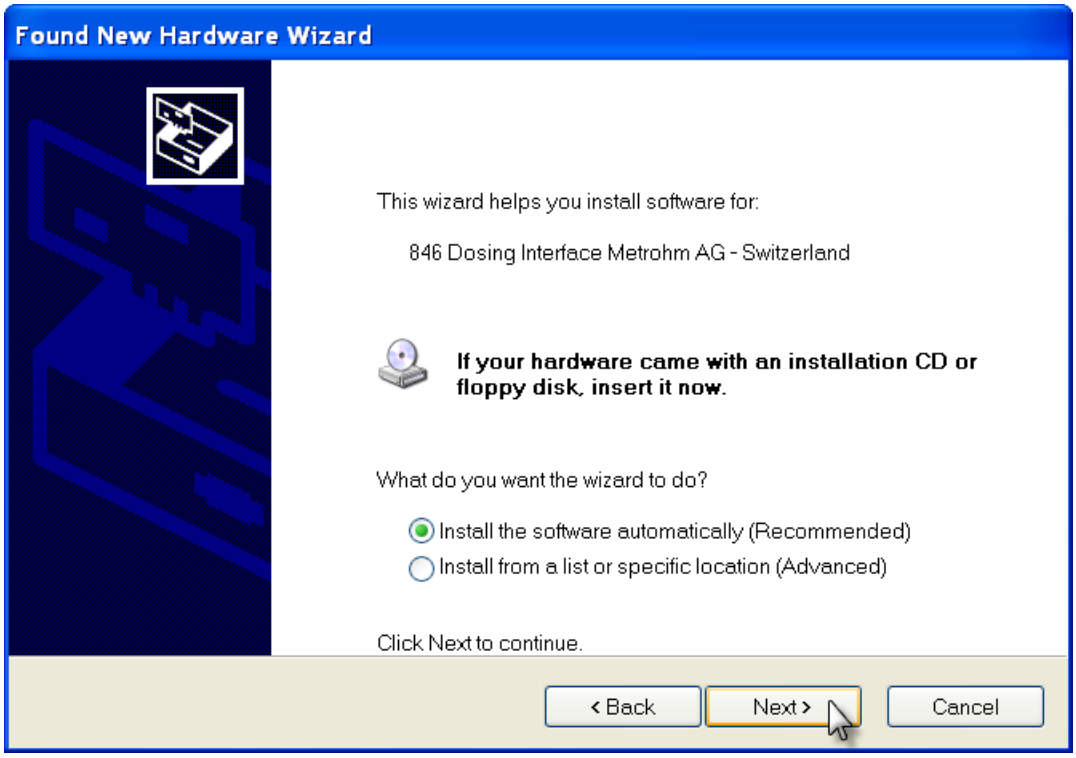

Figure 1 – The found new hardware window

The drivers should be installed automatically. If the Windows logo testing warning is displayed during the installation process, click the *Continue Anyway* button.

Note: contact your Metrohm distributor for more information about device drivers and installation guidelines.

Warning: make sure that the Metrohm devices are connected to the computer and are switched on **before** starting the NOVA software.

2.3 – Dosino 800 control

The current version of NOVA supports the Metrohm 800 Dosino through the MSB interface[1]. Dedicated commands are available in the Control Metrohm devices group of the command browser (see Figure 2)

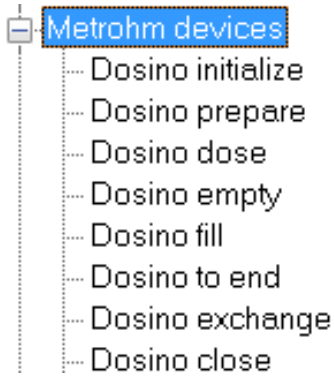

Figure 2 – The Dosino related commands are located in the Control Metrohm devices group

The following Dosino control commands are available[2]:

- **Dosino initialize** – used to setup the Dosino and initialize the control.
- **Dosino prepare** – fills the dosing cylinder and the tubings.
- **Dosino dose** – delivers the specified volume through the dosing port.
- **Dosino empty** – empties the dosing cylinder and the tubings.
- **Dosino fill** – fills the dosing cylinder completely.
- **Dosino to end** – empties the dosing cylinder completely.
- **Dosino exchange** – prepares the dosing unit for exchange.
- **Dosino close** – terminates the Dosino control.

Note: all the Dosino control commands are tagged as *Intermediate* commands, except the Dosino to end and the Dosino exchange commands that are tagged as *Advanced* commands[3].

[1] Metrohm devices with MSB connectors cannot be connected to the computer directly. These devices must be connected to a Metrohm instrument with USB interface or to a dosing interface with USB. Refer to section 1 for more information.
[2] Refer to the Metrohm Dosino user manual for more information.
[3] Please refer to the Command list document available from the Help menu of Nova for more information.

2.3.1 – Dosino initialize

To control a Metrohm 800 Dosino with NOVA, the dosing unit must be initialized. This is done using the Dosino initialize command (see Figure 3).

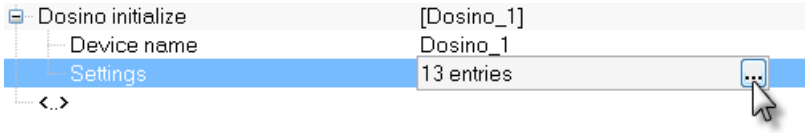

Figure 3 – The Dosino initialize command

Using the Dosino initialize command, it is possible to specify a name for the dosing unit (default: Dosino_1), under device name. This name can be defined by the user. It is also possible to define the Settings of the dosing unit by clicking the ... button (see Figure 3).

A new window will be displayed, allowing the specification of the Dosing unit serial number, dosing ports, tubing volume and dosing rates (see Figure 4).

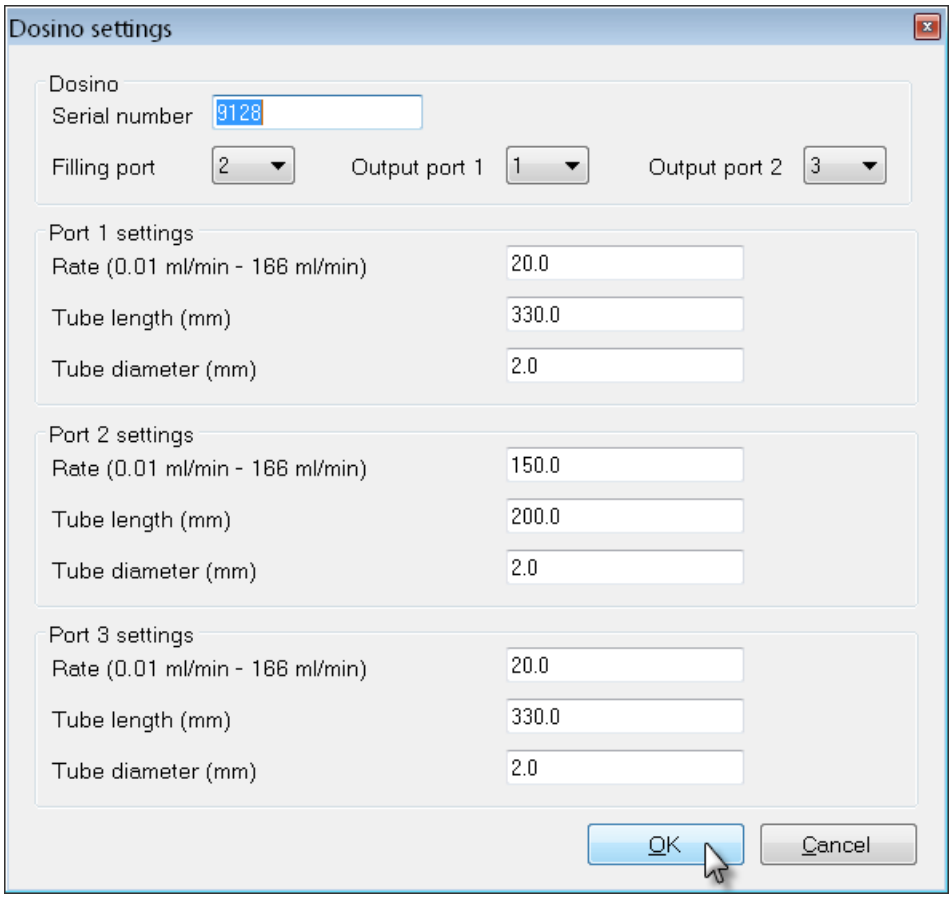

Figure 4 – Overview of dosing unit settings

Note: if one of the ports is not required, set the tube diameter and the tube length to 0 mm.

Figure 5 provides an overview of the ports located on the dosing unit.

The rates are defined in ml/s, the tube length and diameter are defined in mm. The serial number consists in the five last digits of the serial number printed on the MSB cable[4].

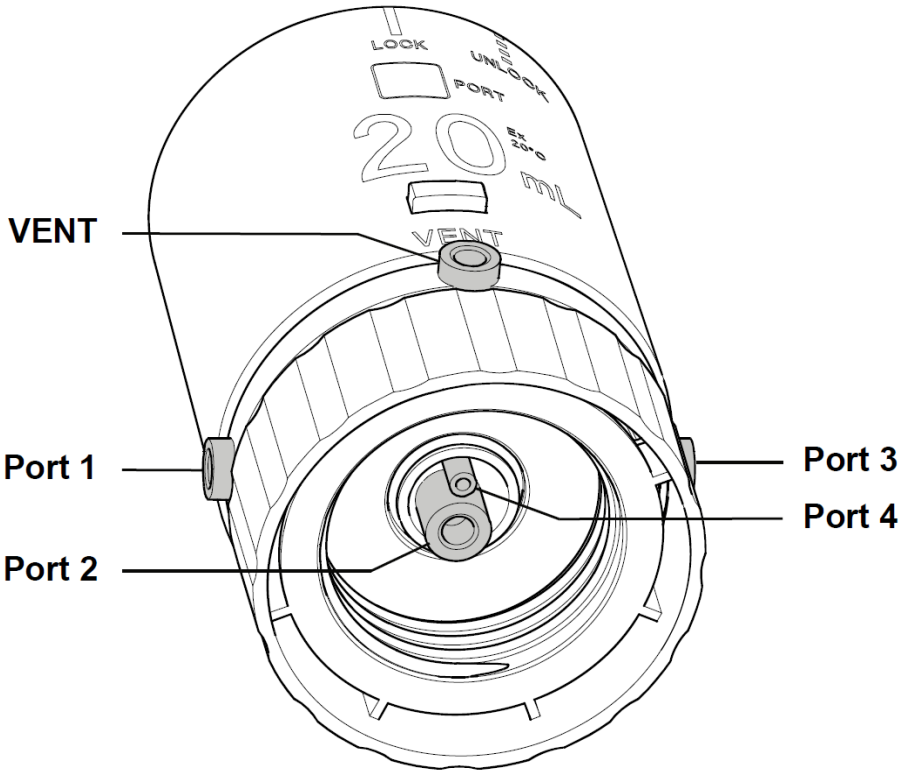

Figure 5 – Overview of the ports located on the dosing unit

Tubing settings can be defined for each port of the device, along with filling and dispensing rates. The volume of the dosing unit is automatically recognized.

Note: it is possible to control more than one dosing unit through NOVA. However, for each dosing unit, a separate Dosino initialize command must be added to the procedure. The device name must be unique.

Once a dosing unit has been initialized, the remaining commands available in the Control Metrohm device group can be used. When the initialize command is used in a procedure, the dosing unit is automatically filled.

[4] In case the first of the five digits is a zero, then use the last four digits instead.

2.3.2 – Dosino prepare

This command prepares the dosing unit by rinsing and filling the connected tubing and the dosing cylinder. The tubing system of the Dosino should be freed from air bubbles at least once per day by carrying out a preparation cycle. This is a process that could take some time. It is recommended to use this command at the beginning of the procedure.

During the preparation process the dosing cylinder as well as the connected tubings are completely filled. Several filling and dosing processes can be carried out. The volumes required for this are calculated from the settings of the Dosing unit, i.e. from the tubing lengths and diameters.

The content of the dosing cylinder is ejected via the pre-defined port. Figure 6 shows an overview of the Dosino prepare command used in a procedure.

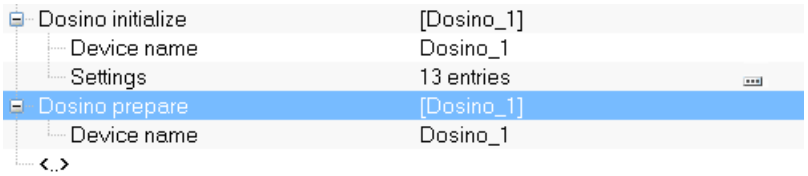

Figure 6 – Using the Dosino prepare command

Note: the Device name parameter must be the same as one of the initialized dosing units.

Tip: it is possible to link the Device name parameters in the Dosino commands to avoid using an undefined Dosino (see Figure 7).

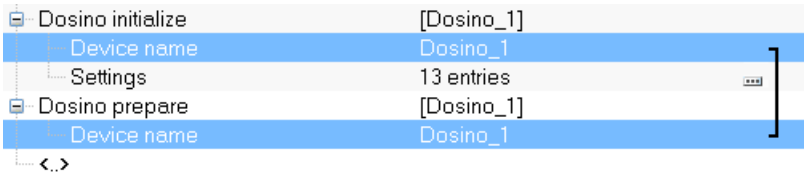

Figure 7 – The Device name parameter can be linked

2.3.3 – Dosino dose

This command delivers the specified volume (in ml) through the pre-defined port. If the volume exceeds the dosing unit volume, the Dosino will be refilled before the dosing command is resumed. If a negative value is defined, the volume will be aspirated.

Figure 8 shows how to use the Dosino dose command.

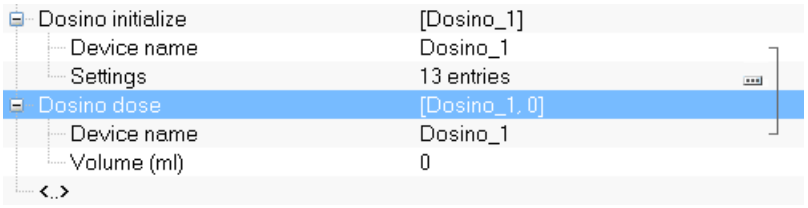

Figure 8 – Using the Dosino dose command

While the Dosino is dispensing the requested volume through the pre-defined port, a window indicating the dosing progress will be displayed (see Figure 9).

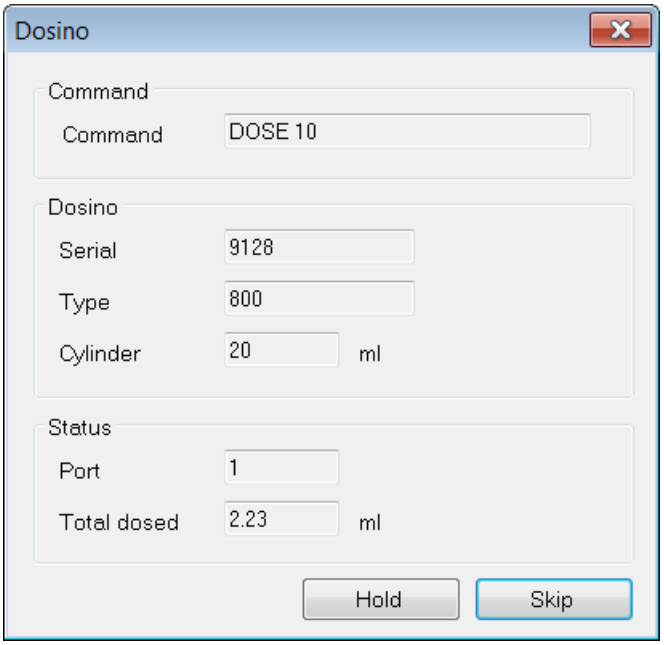

Figure 9 – The dosing progress

Pressing the hold button will interrupt the dosing command, while pressing the skip button will interrupt the whole dosing command and continue with the next command in the procedure.

Note: like the previous commands, the name of the Dosino specified in the Device name parameter must correspond to one of the initialized Dosinos.

Tip: it is possible to add a link between the volume used in the Dosino dose command and another NOVA command, like an Input box (see Figure 10).

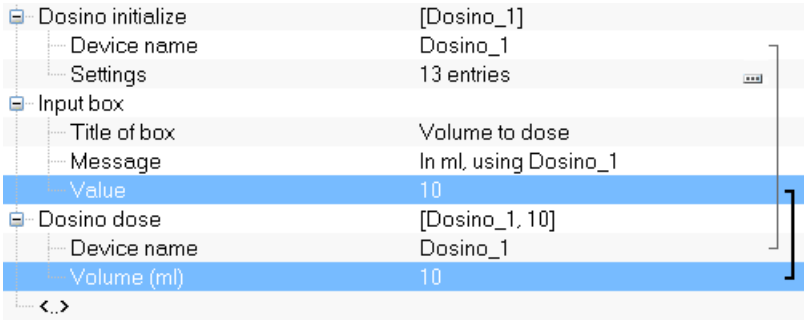

Figure 10 – Linking the Dosino dose command to an Input box

2.3.4 – Dosino empty

With this command, the tubing system and Dosing unit cylinder can be completely emptied. The liquid in the dosing cylinder is ejected via the dosing port specified in the Dosino initialize command. The air required to displace the liquid from the tubing is aspirated via the given port. This command performs the opposite as the Dosino prepare command.

Figure 11 shows how to use the Dosino empty command.

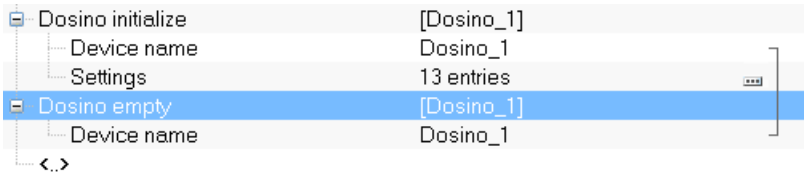

Figure 11 – The Dosino empty command

2.3.5 – Dosino fill

This command can be used to fill the dosing unit cylinder completely. The liquid is aspirated via the given port.

Figure 12 shows how to use the Dosino fill command.

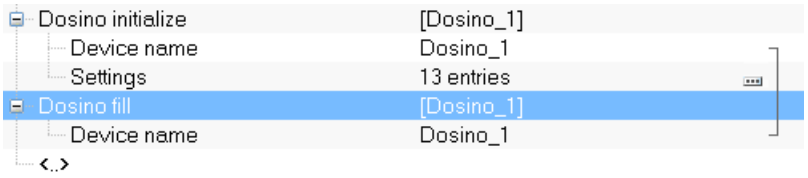

Figure 12 – The Dosino fill command

2.3.6 – Dosino to end

With this command, the content of the dosing cylinder is ejected via the pre-defined port. The piston stops at the specified end volume. This is useful for pipetting functions or for removing air bubbles from the dosing cylinder.

Figure 13 shows how to use the Dosino to end command.

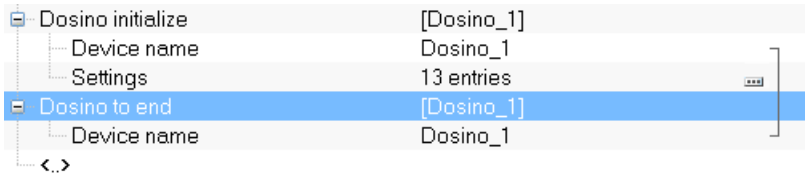

Figure 13 – Using the Dosino to end command

2.3.7 – Dosino exchange

Before changing the dosing unit, the Dosino exchange command must be used to fill the dosing cylinder and move the stopcock to the exchange position. The cylinder is filled by aspirating the necessary volume via the specified port.

Figure 14 shows how to use the Dosino exchange command.

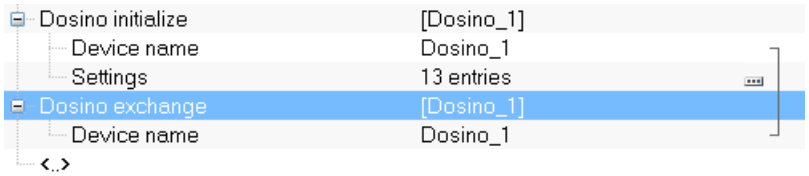

Figure 14 – The Dosino exchange command

2.3.8 – Dosino close

This command must **always** be used to release the connection to the Dosino. To close a connection, add the Dosino close command to the procedure and define the Device name to terminate the remote control of that device (see Figure 15).

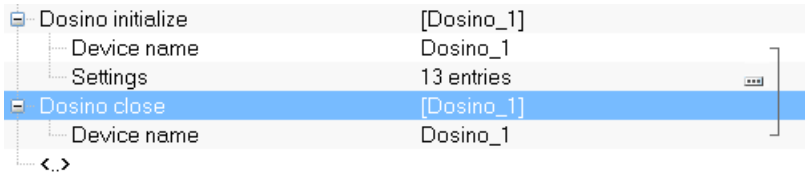

Figure 15 – The Dosino close command must always be used to release the control of the Metrohm Dosino

2.4 – Stirrer 801 control

The current version of NOVA supports the Metrohm 801 Stirrer through the MSB interface. Dedicated commands are available in the Control Metrohm devices group of the command browser (see Figure 16).

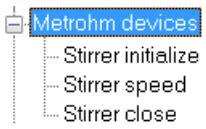

Figure 16 – The Stirrer related commands are located in the Control Metrohm devices group

The following Stirrer control commands are available[5]:

- **Stirrer initialize** – used to setup the Stirrer and initialize the control.
- **Stirrer speed** – defines the rotation rate and direction of the stirrer
- **Stirrer close** – terminates the Stirrer control.

Note: all the Stirrer control commands are tagged as *Intermediate* commands[6].

2.4.1 – Stirrer initialize

To control a Metrohm 801 Stirrer with NOVA, the stirrer must be initialized. This is done using the Stirrer initialize command (see Figure 17).

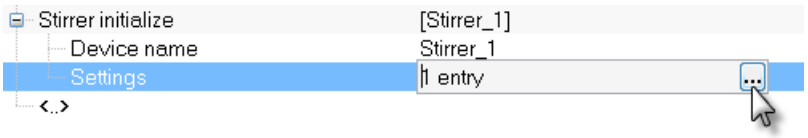

Figure 17 – The Stirrer initialize command

Using the Stirrer initialize command, it is possible to specify a name for the stirrer unit (default: Stirrer_1), under device name. It is also possible to define the Settings of the stirrer by clicking the … button (see Figure 17).

A new window will be displayed, in which the serial number of the Stirrer must be specified (see Figure 18). This serial number consists of the five last digits printed on the MSB cable[7].

[5] Refer to the Metrohm Stirrer user manual for more information.
[6] Please refer to the Command list document available from the Help menu of Nova for more information.
[7] In case the first of the five digits is a zero, then use the last four digits instead.

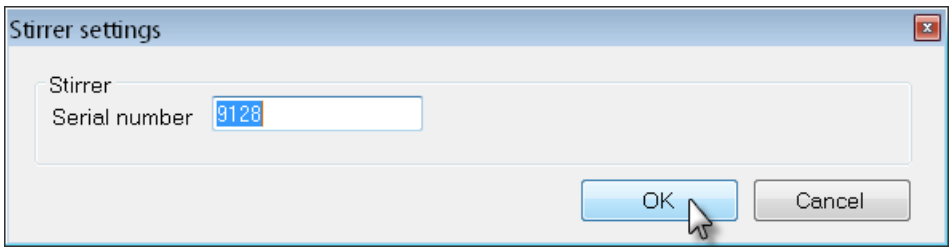

Figure 18 – The Stirrer settings

Note: it is possible to control more than one stirrer through NOVA. However, for each stirrer, a separate Stirrer initialize command must be added to the procedure. The device name must be unique.

Once a stirrer has been initialized, the remaining commands available in the Control Metrohm device group can be used.

2.4.2 – Stirrer speed

This command defines the rotation rate and direction of the stirrer. The speed is defined by entering an integer between -15 and 15. The actual rotation rate is the entered value multiplied by a *speed change per step coefficient,* which is the case of the 801 Stirrer is 180 RPM/step[8].

If the speed value is positive, the stirrer will rotate in the clockwise direction. A negative value will rotate the stirrer anti-clockwise. Figure 19 shows an overview of the Stirrer speed command used in a procedure.

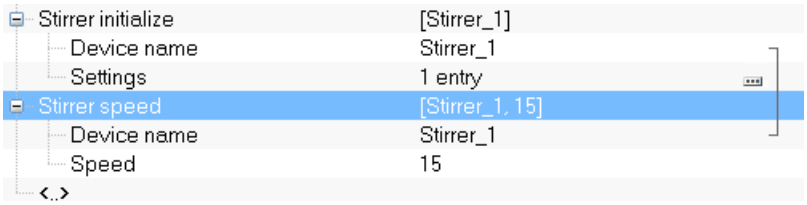

Figure 19 – Using the Stirrer speed command

Note: the Device name parameter must be the same as one of the initialized dosing units.

2.4.3 – Stirrer close

This command must **always** be used to release the connection to the Stirrer. To close a connection, add the Stirrer close command to the procedure and define the Device name to terminate the remote control of that device (see Figure 20).

[8] Factory default value.

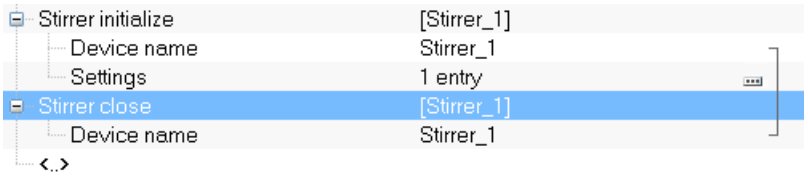

Figure 20 – The Stirrer close command must always be used to release the control of the Metrohm stirrer

2.5 – Sampler processor 814/815/858 control

The current version of NOVA supports the Metrohm 814 USB Sample processor, the 815 Robotic USB Sample processor and the 858 Professional Sample processor through the USB interface[9]. Dedicated commands are available in the Control Metrohm devices group of the command browser (see Figure 21).

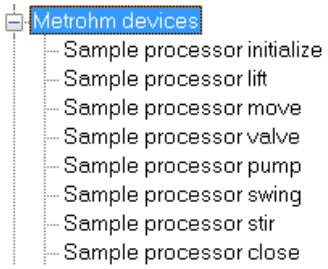

Figure 21 – The Sample processor related commands are located in the Control Metrohm devices group

The following Sample processor control commands are available[10]:

- **Sample processor initialize** – used to setup the Sample processor and initialize the control.
- **Sample processor lift** – defines the position of the lift of the Sample processor.
- **Sample processor move** – rotates the sample rack to the defined position.
- **Sample processor valve** – controls the valves of the Sample processor.
- **Sample processor pump** – controls the pumps of the Sample processor.
- **Sample processor swing** – controls the position of the Swing head.
- **Sample processor stir** – controls the stirrer of the Sample processor.
- **Sample processor close** – terminates the Sample processor control.

Note: all the Sample proceessor control commands are tagged as *Intermediate* commands, except the Sample processor valve, Sample processor pump and Sample processor swing commands that are tagged as *Advanced* commands[11].

[9] In the rest of this document, the supported sample processors will be referred to as Sample processor.
[10] Refer to the Metrohm Sample processor user manual for more information.
[11] Please refer to the Command list document available from the Help menu of Nova for more information.

2.5.1 – Sample processor initialize

To control a Metrohm Sample processor with NOVA, the device must be initialized. This is done using the Sample processor initialize command (see Figure 22).

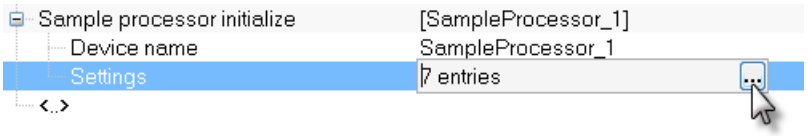

Figure 22 – The Sample processor initialize command

Using the Sample processor initialize command, it is possible to specify a name for the device (default: SampleProcessor_1), under device name. It is also possible to define the Settings of the device by clicking the ... button (see Figure 22).

A new window will be displayed, allowing the specification of the Sample processor serial number, lift, swing and shift rates, as well as the location of the Sampler processor configuration files (*Config.xml*) and Rack configuration file (*6204xxxx.mtrf*[12]) (see Figure 23).

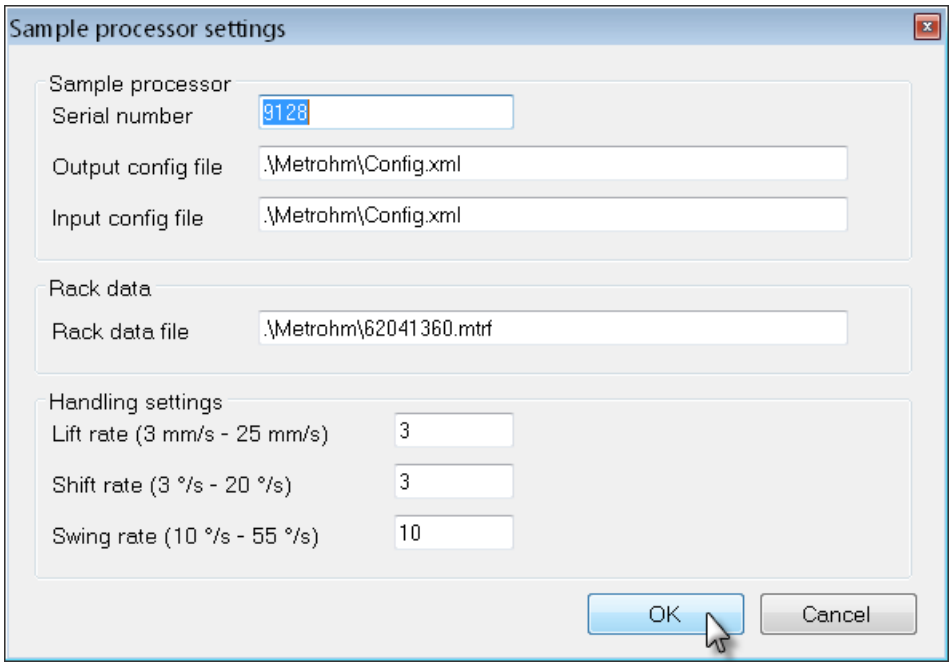

Figure 23 – Overview of Sample processor settings

The lift rate is defined in mm/s, the shift rate and the swing rate are defined in degrees/s. The serial number consists of the five last digits of the Sample processor serial number[13].

[12] The four last digits depend on the rack used with the sample processor.
[13] In case the first of the five digits is a zero, then use the last four digits instead.

Note: the *Config.xml* and *6204xxxx.mtrf* files must be located in the specified folders. Make sure that the rack code file corresponds to the rack code used on the Sample processor.

Warning: when the 786 Swing head is used, a special modification of the Config.xml file is required, depending on the type of Robotic arm mounted on the Swing head. Please refer to the Appendix for more information.

Note: it is possible to control more than one Sample processor through NOVA. However, for each device, a separate Sample processor initialize command must be added to the procedure. The device name must be unique.

Once a Sample processor has been initialized, the remaining commands available in the Control Metrohm device group can be used. When the initialize command is used in a procedure, the Sample processor automatically resets the lift(s) and the sample rack to their respective initial positions.

2.5.2 – Sample processor lift

This command changes the position of the lift on the specified Sample processor tower. The position of the lift can be defined between 0 mm (top of the tower) and 235 mm (bottom of the tower). The rate of ascent and descent is defined in the Sample processor initialize command, in mm per second.

Figure 24 shows an overview of the Sample processor lift command used in a procedure.

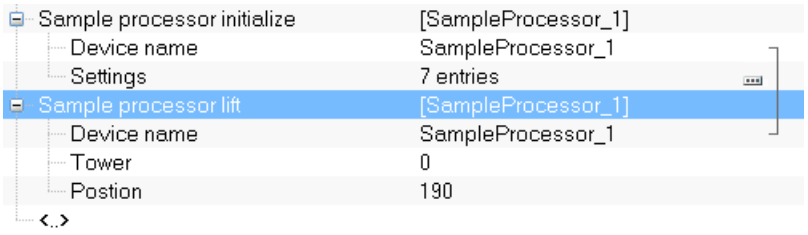

Figure 24 – Using the Sample processor lift command

Note: tower 0 is the first tower and tower 1 is the optional second tower. It is only possible to control one lift at a time.

2.5.3 – Sample processor move

This command changes the position of the sample rack, relative to the specified Sample processor tower, to the required sample position. The sample position defined between 1 (initial position) and *x*, where *x* is the number of position on the rack. The rate of rotation of the rack is defined in the Sample processor initialize command, in degrees per second.

Figure 25 shows an overview of the Sample processor move command used in a procedure.

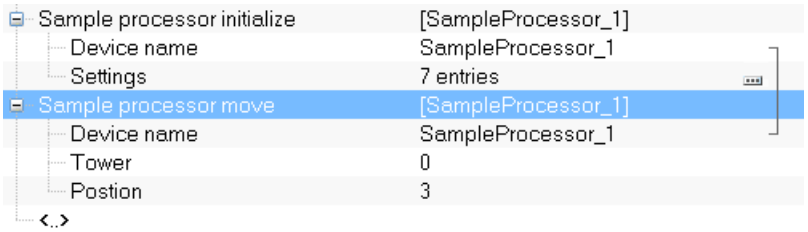

Figure 25 – Using the Sample processor move command

Note: tower 0 is the first tower and tower 1 is the optional second tower.

2.5.4 – Sample processor valve

This command controls the valves of the specified tower. Setting the value to 0 switches the valve off, while setting the value to 1 switches the valve on. Four combinations are allowed:

- **00** – valve 1 off, valve 2 off
- **01** – valve 1 off, valve 2 on
- **10** – valve 1 on, valve 2 off
- **11** – valve 1 on, valve 2 on

Figure 26 shows an overview of the Sample processor valve command used in a procedure.

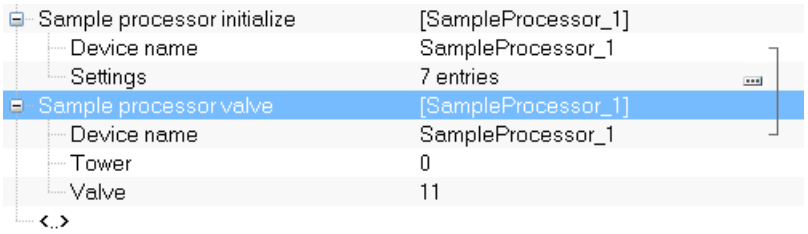

Figure 26 – Using the Sample processor valve command

Note: tower 0 is the first tower and tower 1 is the optional second tower.

2.5.5 – Sample processor pump

This command controls the pumps of the specified tower[14]. Setting the value to 0 switches the pump off, while setting the value to 1 switches the pump on. Four combinations are allowed:

- **00** – pump 1 off, pump 2 off
- **01** – pump 1 off, pump 2 on
- **10** – pump 1 on, pump 2 off
- **11** – pump 1 on, pump 2 on

[14] The pumps can be used for rinsing the titration head and aspirating off samples that have been processed. The necessary tubing is supplied with the corresponding Metrohm instruments (refer to the Metrohm Installation Instructions for more information).

Figure 27 shows an overview of the Sample processor pump command used in a procedure.

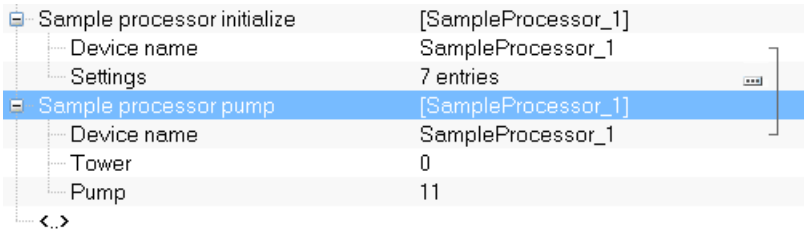

Figure 27 – Using the Sample processor pump command

Note: tower 0 is the first tower and tower 1 is the optional second tower.

2.5.6 – Sample processor swing[15]

This command changes the position of the swing head on the specified Sample processor tower. The position of the swing can be defined between 0 degrees (initial position) and 90 degrees. The swing rate is defined in the Sample processor initialize command, in degrees per second.

Figure 28 shows an overview of the Sample processor lift command used in a procedure.

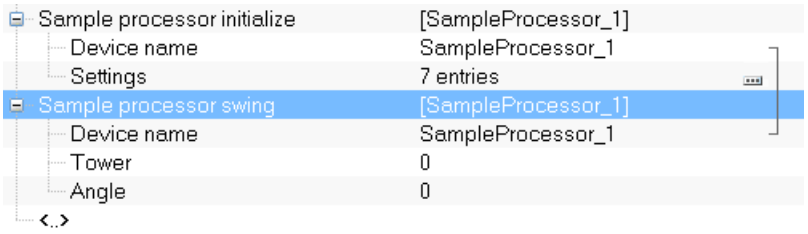

Figure 28 – Using the Sample processor swing command

Note: tower 0 is the first tower and tower 1 is the optional second tower. It is only possible to control one lift at a time.

[15] This command requires the 786 Swing head.

2.5.7 – Sample processor stir[16]

This command defines the rotation rate and direction of the stirrer on the specified tower. The speed is defined by entering an integer between -15 and 15. The actual rotation rate is the entered value multiplied by a *speed change per step coefficient,* which is the case of the 802 Stirrer is 140 RPM/step[17].

If the speed value is positive, the stirrer will rotate in the clockwise direction. A negative value will rotate the stirrer anti-clockwise. Figure 29 shows an overview of the Stirrer speed command used in a procedure.

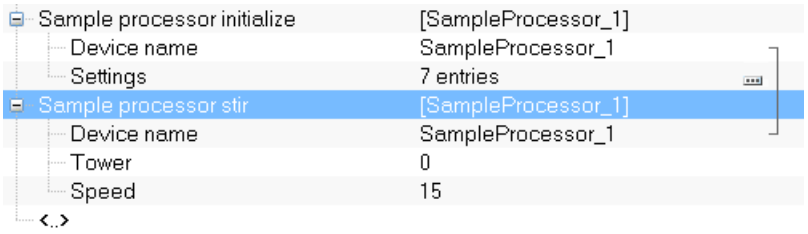

Figure 29 – Using the Sample processor stir command

Note: tower 0 is the first tower and tower 1 is the optional second tower.

2.5.8 – Sample processor close

This command must **always** be used to release the connection to the Sample processor. To close a connection, add the Sample processor close command to the procedure and define the Device name to terminate the remote control of that device. (see Figure 30)

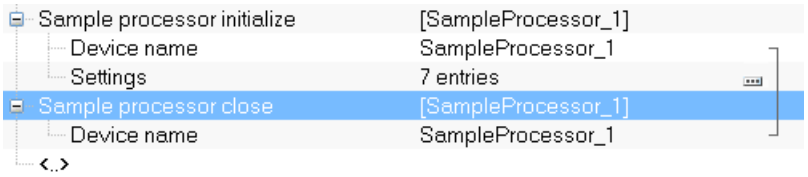

Figure 30 – The Sample processor close command must always be used to release the control of the Metrohm sample processor

[16] This command requires the 802 Stirrer.
[17] Factory default value.

3 – TTL triggers (DIO)

The Digital Input/Output (DIO) of the Autolab offers the possibility of synchronizing measurements with external devices that can be controlled by TTL signals.

Every Autolab instrument is equipped with one or two digital input/output connectors (DIO) that can be used to receive or send a digital TTL trigger. Depending on the instrument type, two different connector layouts are available.

3.1 – Autolab PGSTAT series 7 and series 8

The Autolab PGSTAT12, 128N, 30, 302, 302N and 100 from the 7 and 8 series are fitted with two DIO connectors, located on the back plane of the instrument. Each DIO connector has 25 pins and is divided into three ports (see Figure 31):

- Port A, pins 1 to 8
- Port B, pins 17 to 24
- Port C, pins 9 to 12 (C upper) and pins 13 to 16 (C lower)
- Pin 25 is the digital ground (DGND)

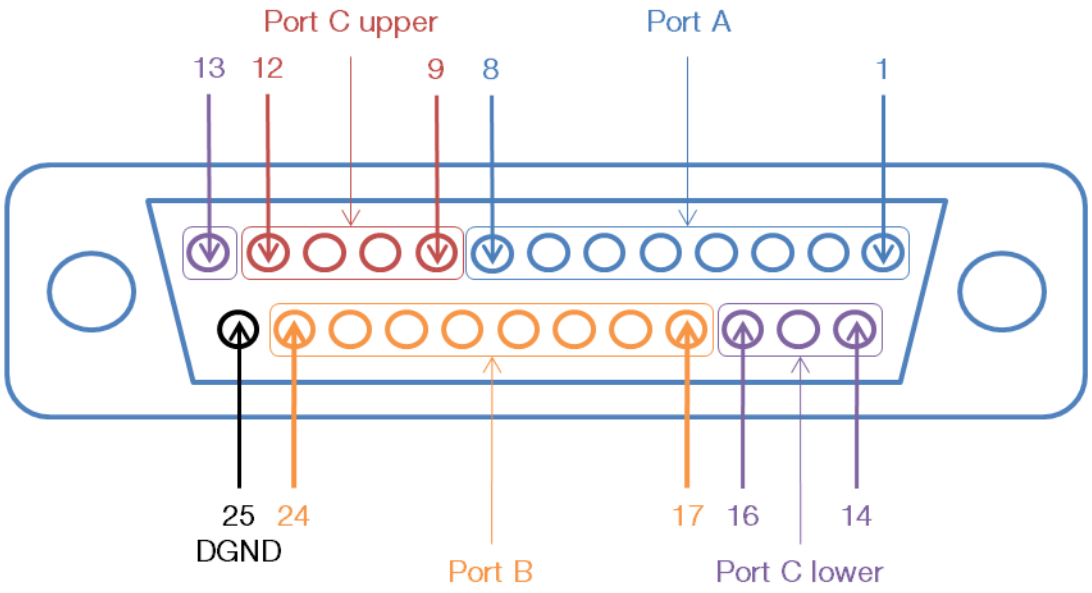

Figure 31 – Overview of the DIO connector

In the software is it possible to address each of the 24 available pins, by port. Each port can be set to write or read. This provides a convenient way of sending or receiving a predefined digital TTL trigger to or from another device in order to synchronize measurements or to control events during a measurement. The TTL triggers must be set according to the communication protocols defined in user manual of the ancillary device connected to the DIO connector on the Autolab.

Note: there is a risk of introducing a ground loop when connecting an external device to the Autolab. This can result in a higher than expect noise level during the measurements.

3.2 – Autolab PGSTAT101

The PGSTAT101 is fitted with a single, non configurable DIO connector located on the back plane of the instrument. The DIO connector has a total of eight write lines and four read lines (see Figure 32).

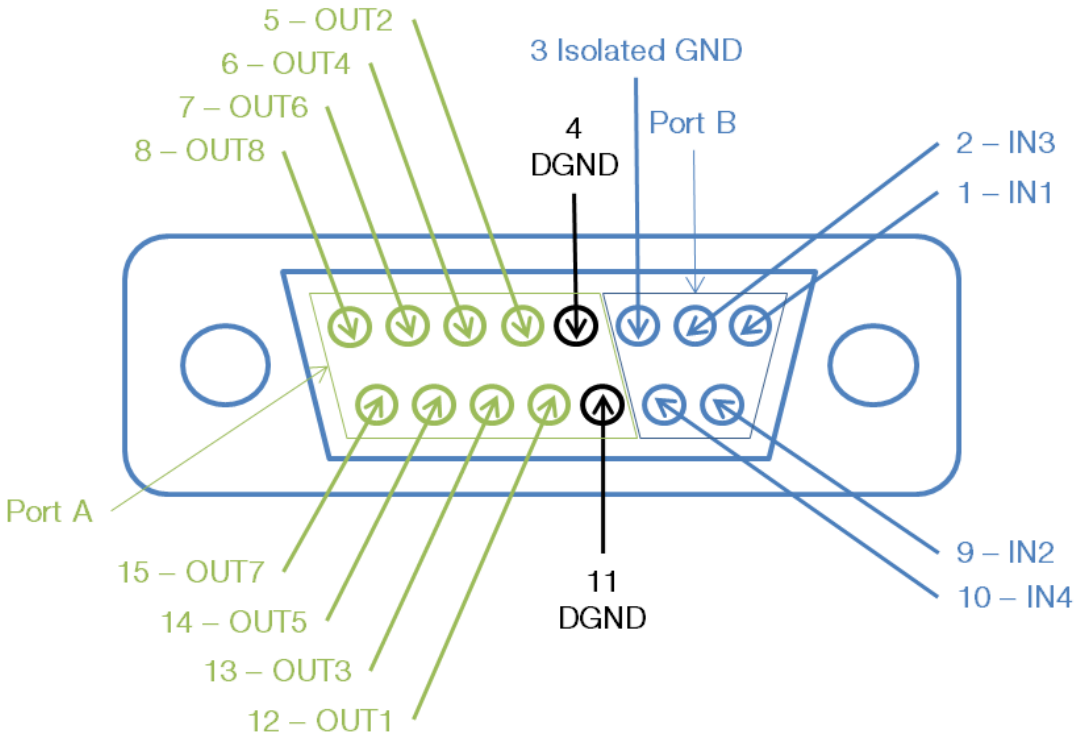

Figure 32 – Mapping of the PGSTAT101 DIO connector

Port A includes 8 write lines and two digital ground pins (pin 4 and pin 11). Port B includes 4 read lines and an isolated ground pin (pin 3). All pins on Port B are galvanically isolated. Pin 4 or pin 11 should not be used as ground pin for the read lines.

Important: the write lines of the PGSTAT101 DIO connector are capable of supplying a maximum current of 200 mA. Suitable pull-down resistors should be placed in the write lines of the DIO cable connected to the PGSTAT101. A typical value for the pull-down resistance is about 1 kΩ. Please refer to the user manual of the external device connected to the PGSTAT101 for more information.

3.3 – DIO trigerring

Each pin on the DIO connector(s) of the Autolab can be set to two different levels:

- **Low (0 V):** this is the default status of each pins when the Autolab is initialized. This status corresponds to a digital "0".
- **High (5 V):** this status corresponds to a digital "1".

Note: the maximum current load on each pin is 2.5 mA for all the PGSTAT instruments except for the PGSTAT101 for which the maximum load current is 200 mA.

Depending on the communication protocols, an external device connected to the Autolab can be triggered by a down to up transition or an up to down transition.

3.3.1 – Port initialization

Before a DIO connector can be used to send or receive TTL triggers, the ports involved must be initialized. Two different modes are available:

- **Direction – Input:** the port is initialized to 'listen' mode and will be used to receive TTL triggers.
- **Direction – Output:** the port is initialized to 'talk' mode and will be used to send TTL triggers.

Initialization of a port is performed in the Autolab control window, on the DIO(1) tab (see Figure 33).

Note: this setting is not available for the PGSTAT101 since the pins of the DIO connector are pre-configured to input and output (see Section 3.2). Ports A and B are initialized at startup.

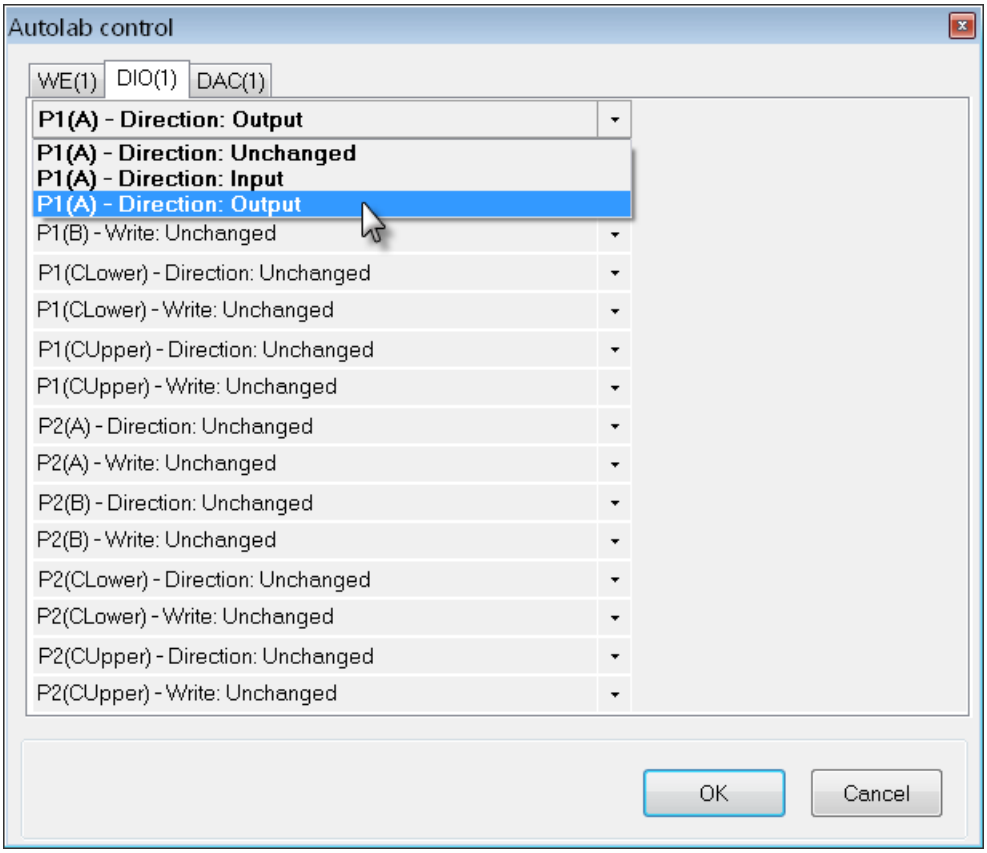

Figure 33 – Initialization of the port is done in the Autolab control window

Each individual port can be set to Input or Output independently. It is recommended to initialize the DIO ports at the beginning of the procedure.

3.3.2 – Sending triggers

To send TTL triggers, the *Autolab control* command must be used. Before a DIO port can be used to send a TTL trigger, it must be initialized into Output direction (see previous section).

The Autolab control window is used to define the DIO port status according to an 8-bit binary string or the converted decimal value corresponding to this string. The string is specified directly in the Write drop-down box in the Autolab control window (see Figure 34).

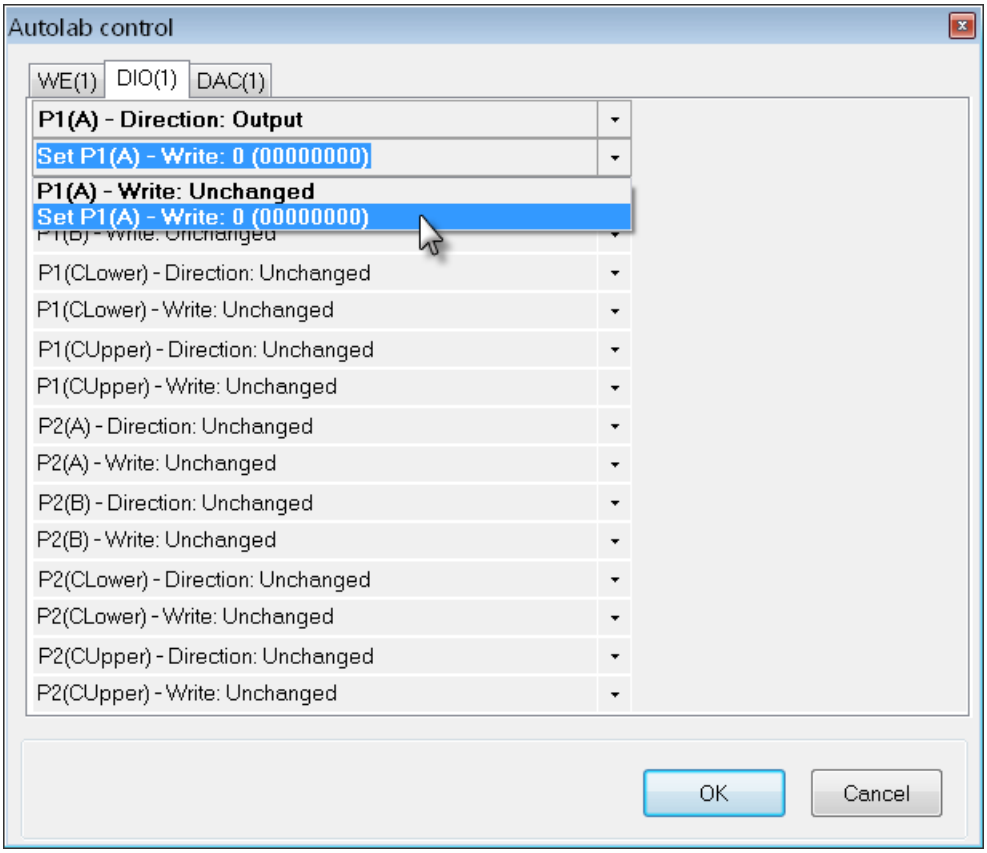

Figure 34 – The Autolab control window is used to set the DIO port

The following syntax is used to define a trigger bit sequence. Each pin of a port corresponds to one bit. Each pin can be set to Low (0) or High (1) status and the sequence of 8 pins defines the trigger bit sequence. The trigger sequence can be written in *decimal* or in *binary:*

- **Binary:** the sequence is written between brackets: (00000000)
- **Decimal:** the value is written as an integer: 0

Example

Suppose that the 8 pins of port A of DIO connector 1 must be set to the following sequence (see Figure 35):

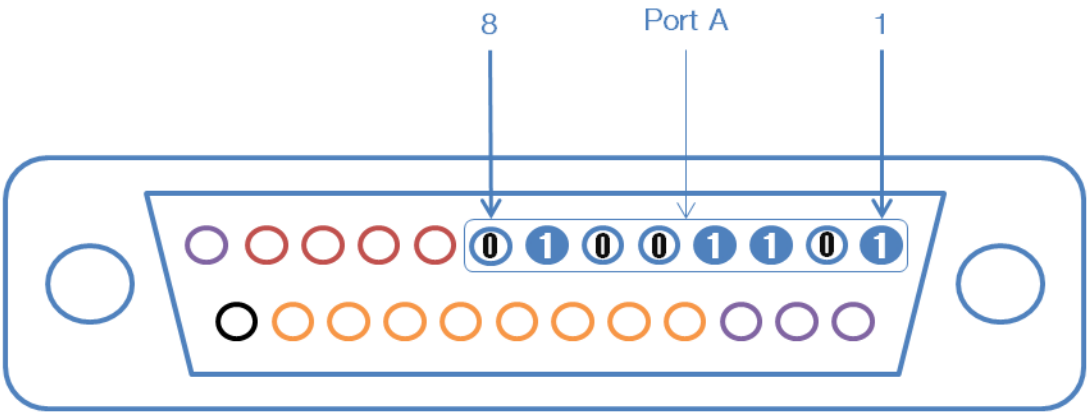

Figure 35 – Example of a DIO bit sequence

The bits for each pin would be written as:

Pin #	8	7	6	5	4	3	2	1
Bit	0	1	0	0	1	1	0	1

The corresponding byte would be **01001101** in binary which corresponds to **77** in decimal.

Tip: converting binary to decimal can be quickly done using the Windows calculator (in Programmer mode). Start the Calculator and switch to binary mode using the Bin button. Type the byte in the calculator (see Figure 36).

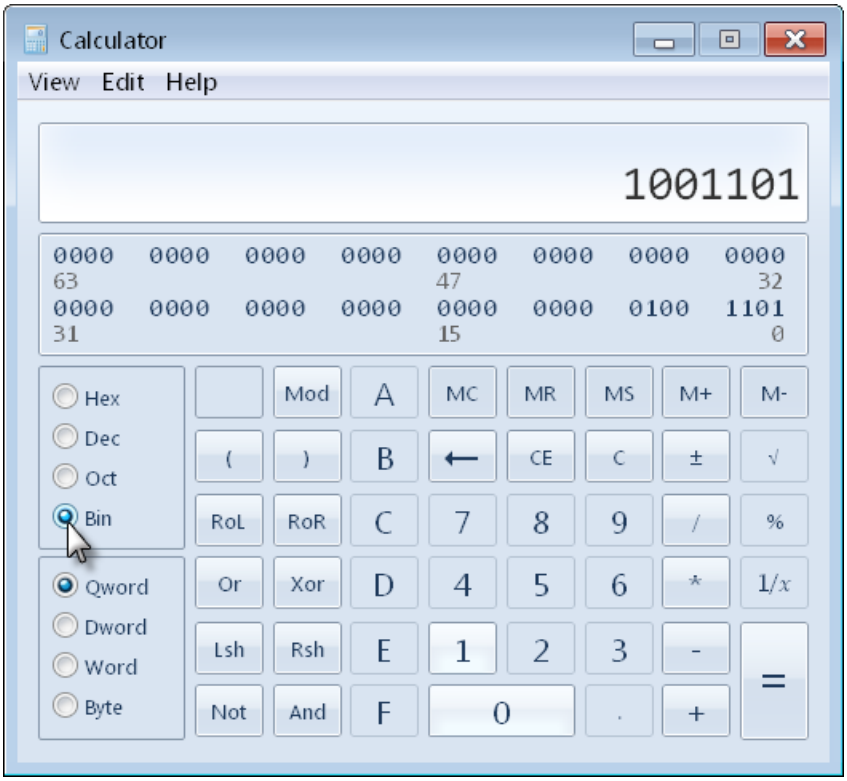

Figure 36 – The windows calculator can be used to convert binary to decimal

Click the Dec button in the top left corner of the calculator to convert the 8 bit sequence into decimal (see Figure 37).

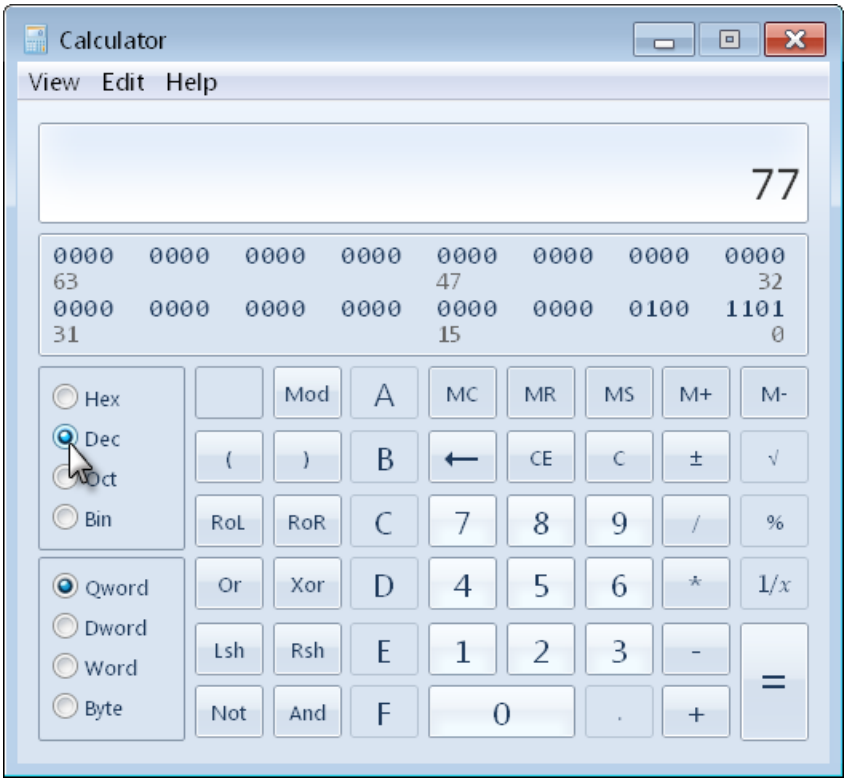

Figure 37 – The windows calculator can be used to convert binary to decimal

The trigger sequence shown in this example can be specified in the Autolab control window in *decimal* (see Figure 38) or in *binary* (see Figure 39).

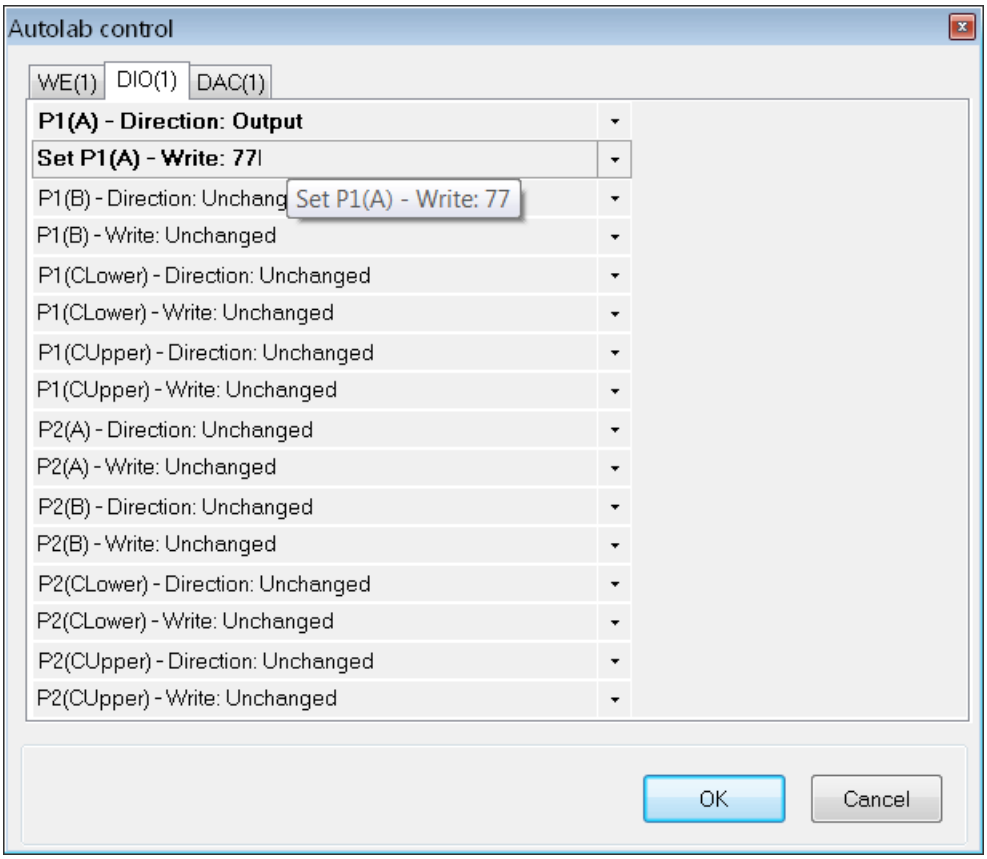

Figure 38 – Writing the trigger sequence in decimal

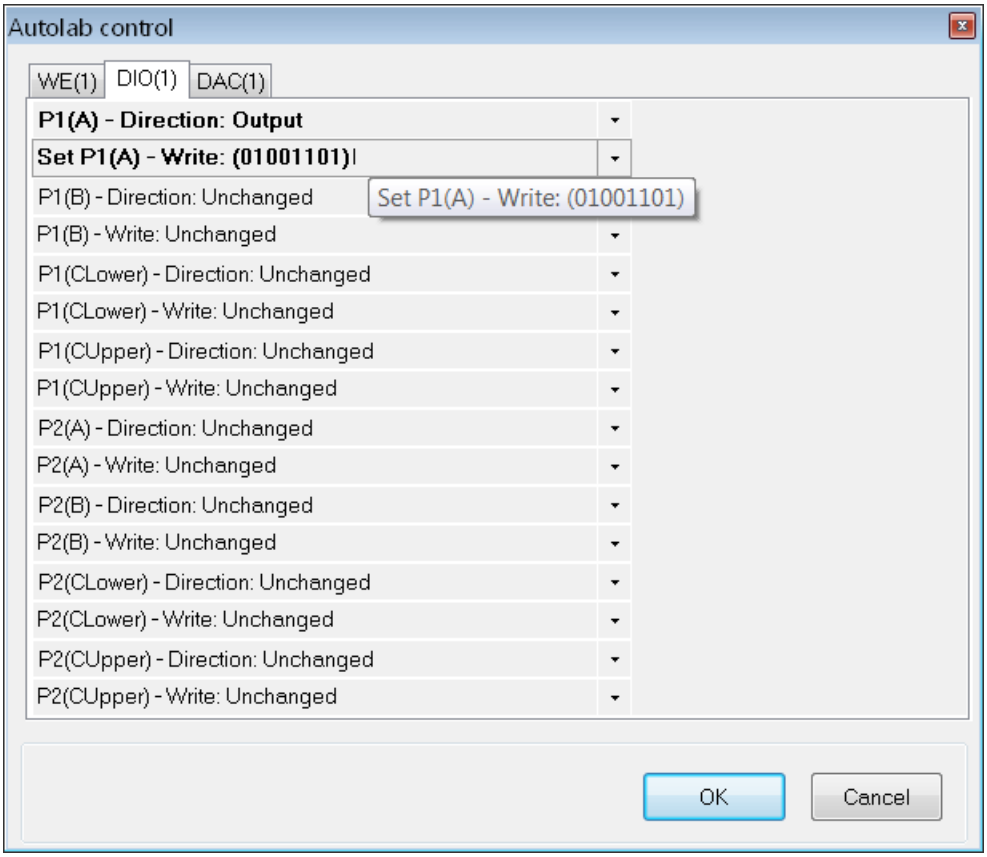

Figure 39 – Writing the trigger sequence in binary

Note: when the port output is set to the defined bit sequence it will remain unchanged until a new sequence is defined. The new sequence can be defined by adding another *Autolab control* command to the procedure.

Note: the PGSTAT101 has a single 8 pin output port on its DIO connector (Port A). The Autolab control window will therefore only show a single output sequence (see Figure 42).

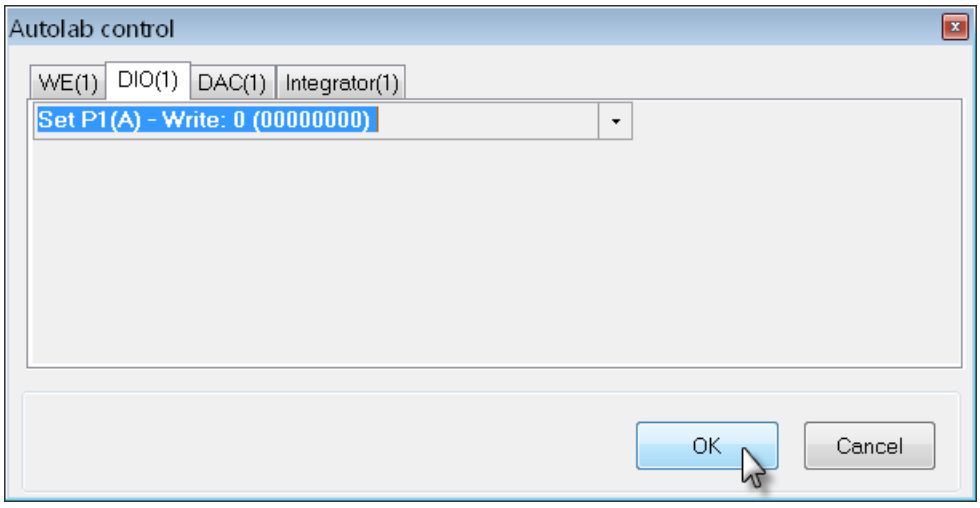

Figure 40 – The 8 pins of the write section of the single DIO port of the PSGTAT101 can be set directly from the Autolab control command

3.3.3 – Receiving triggers

Before a port can be used to receive a trigger, the port must be set to Input mode (see Figure 41 and previous section).

Note: this setting is not available for the PGSTAT101 since the pins of the DIO connector are pre-configured to input and output (see Section 3.2). Ports A and B are initialized at startup.

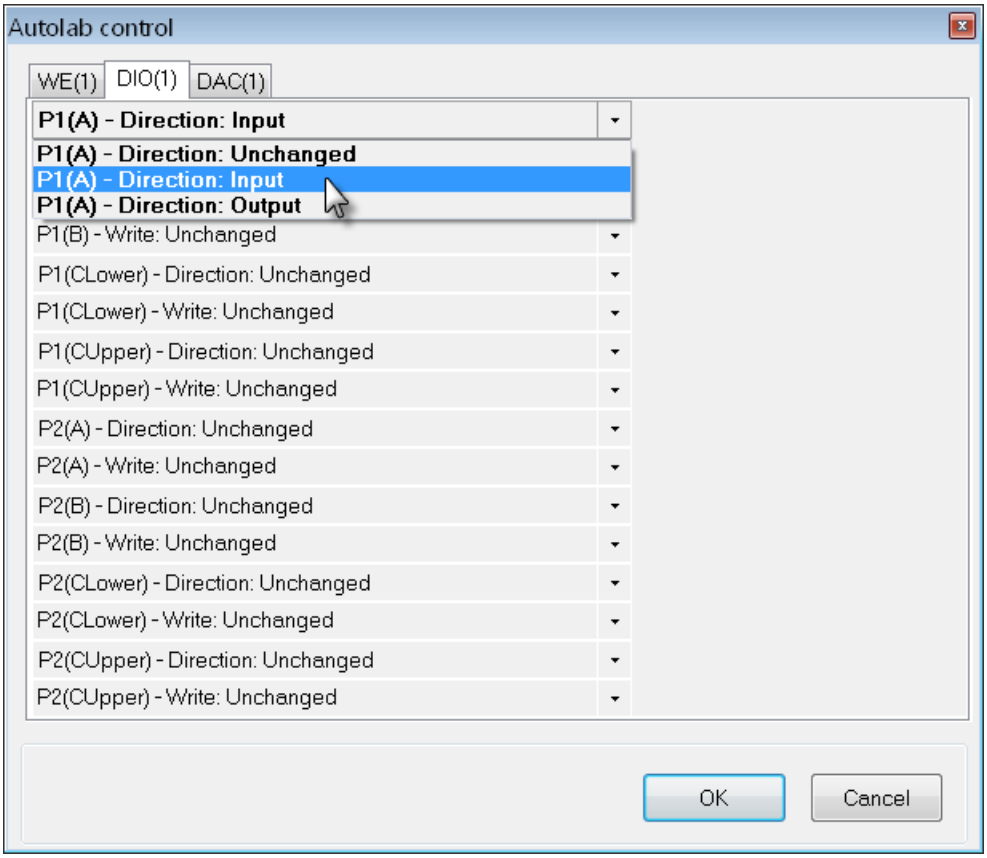

Figure 41 – Setting a port to Input mode

Receiving triggers is done using the dedicated *Wait for DIO trigger* command, available in the Measurement – General group (see Figure 42).

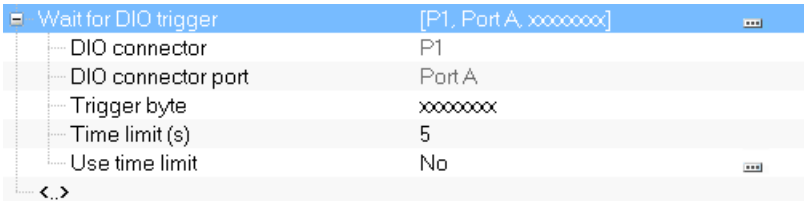

Figure 42 – The Wait for DIO trigger command can be used to receive a TTL trigger

The *Wait for DIO trigger* command has the following parameters:

- **DIO connector:** identifies the connector used to receive the trigger
- **DIO connector port:** defines the port used to receive the trigger (Port A, B, C, C upper, C lower)
- **Trigger byte:** this is the required byte that must be received in order to proceed. A 1 correspond to a "high" position, a 0 corresponds to a "low" position, and an "x" is a wildcard.
- **Time limit (s):** this defines the time limit for the previous parameter.
- **Use time limit: yes/no** – this defines whether an optional timeout limit is used. If a time limit is used, the Wait on trigger command will stop waiting when the time limit is exceeded and the pre-defined trigger byte has not been received.

Pressing the button in the procedure editor opens a dialog window which can be used to set the connector and port to be used to receive the trigger (see Figure 43).

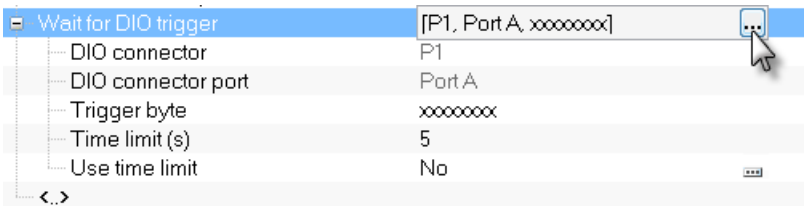

Figure 43 – Opening the Wait for DIO trigger dialog window

The Wait for DIO trigger dialog window displays the available DIO connectors and ports that can be used for receiving the TTL trigger. The trigger byte and the maximum wait time can also be defined (see Figure 44).

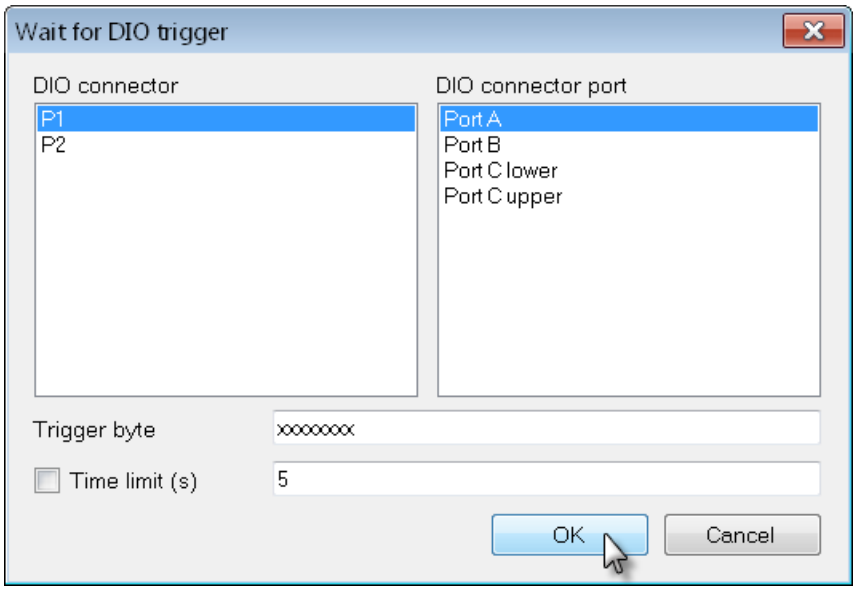

Figure 44 – The Wait for DIO trigger dialog can be used to define the command parameters

For the PGSTAT101, a single input port is available on the DIO connector (see Figure 45).

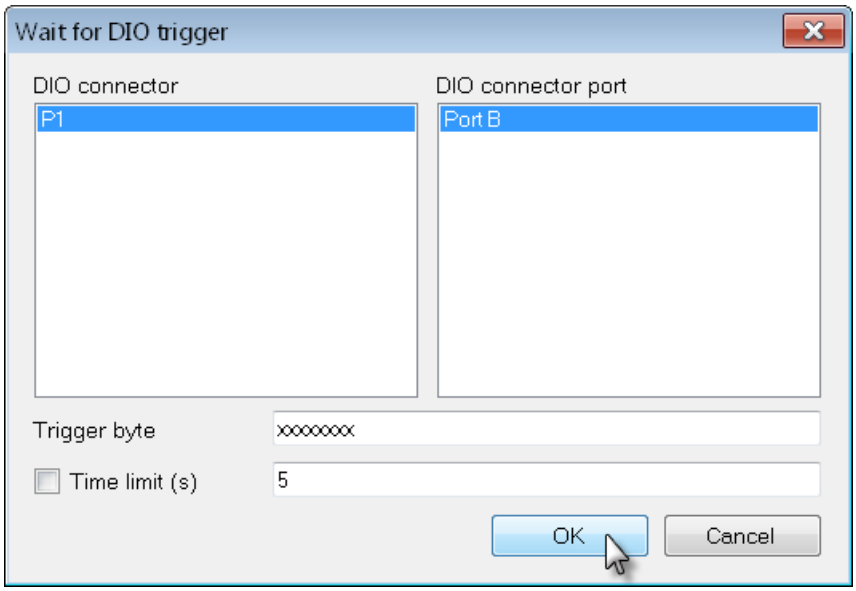

Figure 45 – A single DIO connector is available for the PGSTAT101

Note: the trigger byte consists of 4 bits for the PGSTAT101.

The trigger byte that should be received is defined in the *Wait on DIO trigger* command using an 8 character string containing 1 (high), 0 (low) and x (wild card)[18]. Each character in the string corresponds to a pin on the DIO port, from right to left.

For example, if the measurement should be allowed to proceed when the following trigger byte is received:

Pin #	8	7	6	5	4	3	2	1
Bit	1	0	1	1	0	0	1	0

The corresponding byte would be **10110010**, in binary.

If this Trigger byte is defined in the *Wait on DIO trigger* command, the measurement will be allowed to proceed only when pins #2, 5, 6 and 8 are set to high position **AND** pins #1, 3, 4 and 7 are set to low position.

If the status of pins 1, 2 and 3 are irrelevant, the wildcard symbol, "x", can be used:

Pin #	8	7	6	5	4	3	2	1
Bit	1	0	1	1	0	x	x	x

The corresponding byte would be **10110xxx**, in binary.

[18] Or 4 characters for the PGSTAT101.

If this Trigger byte is defined in the *Wait on DIO trigger* command, the measurement will be allowed to proceed only when pins #5, 6, and 8 are set to high position **AND** pins #4, and 7 are set to low position. The status of pins #1, 2 and 3 is irrelevant.

Note: during the execution of the *Wait for DIO trigger* command, the port status is checked every 100 ms.

When the *Wait for DIO trigger* command is encountered, the Autolab display shows the status of the port at the bottom of the window (see Figure 46). This value is updated every 100 ms.

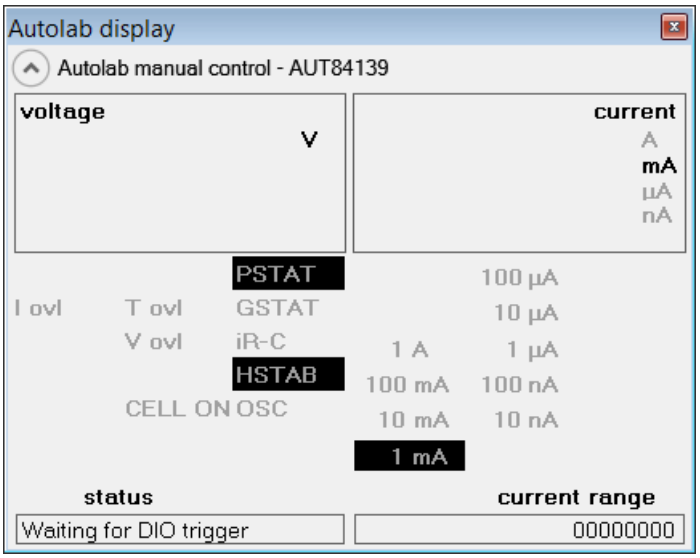

Figure 46 – The Autolab display shows the status of the DIO port

4 – Recommended Standard 232 (RS232)

The RS232 standard describes a communication method where information is sent bit by bit on a physical channel. The information must be broken up in data words. The length of a data word is variable (usually between 5 and 8 bits). For proper transfer additional bits are added for synchronisation and error checking purposes. It is important, that the transmitter and receiver use the same number of bits. Otherwise, the data word may be misinterpreted, or not recognized at all.

The current version of NOVA supports the control of external devices through RS232 communication. Four dedicated commands are available in the Measurement – External devices group of the command browser (see Figure 47).

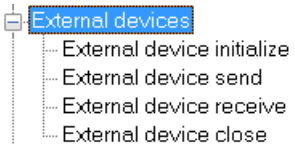

Figure 47 – The commands related to the control of external devices through RS232 communication are located in the Measurement – External devices group

Note: external devices can be controlled through RS232 with or without the Autolab potentiostat / galvanostat connected to the computer.

The following commands are available:

- **External device initialize** – used to setup the connection
- **External device send** – used to send a 'string' to the external device
- **External device receive** – used to read a 'string' from the external device
- **External device close** – used to close the connection

Note: setting up an RS232 communication protocol can be time consuming. We recommend reading the reference manual of the external device carefully and to use, if possible, tools like HyperTerminal to test the connection.

4.1 – External device initialize

Before the RS232 communication can be used to control an external device, the settings of the connection must be defined according to the requirements of the device. These settings can be found in the user manual of the external device.

To setup the communication protocol for the external device, add the External device initialize command to the procedure (see Figure 48).

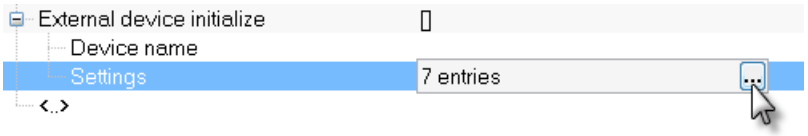

Figure 48 – Add the External device initialize to setup the RS232 communication protocol

Once the command has been added to the procedure, a name for the device must be provided and the Settings of the communication must be defined. These settings can be edited by clicking the ... button (see Figure 48).

A new window will be displayed, allowing the specification of the communication settings (see Figure 49).

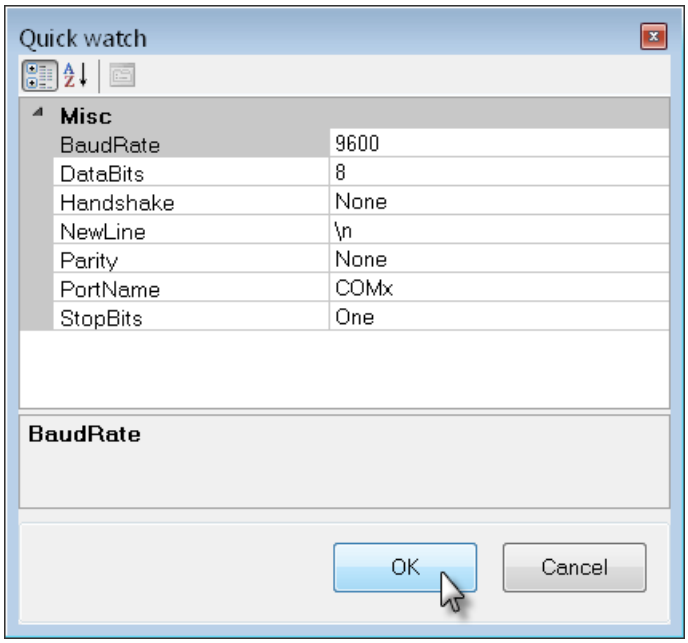

Figure 49 – The RS232 communication settings

The following values need to be defined according to the requirements of the external device:

- Baud rate (Bd)
- Data bits
- Handshake (None, XOnXOff, RequestToSend, RequestToSendXOnXOff)
- New line (\n, \r)
- Parity (None, Odd, Even, Mark, Space)
- Port name: COMx, x is a valid port number
- Stop bits

The example shown in Figure 49 corresponds to the settings for the Julabo HE/F34 water bath.

4.2 – External device send

Once the RS232 communication protocol has been defined and the external device initialized, it is possible to send data strings to the device for control purposes.

To send a string to the external device, add the External device send command to the procedure (see Figure 50).

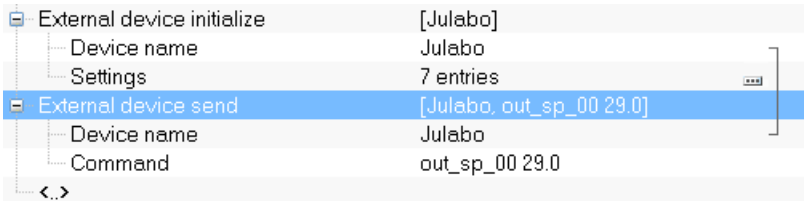

Figure 50 – The External device send command can be used to send a control string to the external device

The example shown in Figure 50 is an example of a control string used to set the temperature control of the Julabo HE/F34 water bath to 29 °C. The syntax of the control string is specific for this device. More information about the communication syntax can usually be found in the user manual of the external device.

Note: the Device name must be the same as the one defined in the External device initialize command. To avoid errors, it is possible to link the Device name parameter of an External device command to the same parameter in the External device send command (as shown in Figure 50).

Tip: it is possible to create a linkable parameter in the data string. This is very convenient because it allows combination between the External device send command and other NOVA commands. Figure 51 shows an example of a link between an Input box parameter and the temperature parameter in the control string.

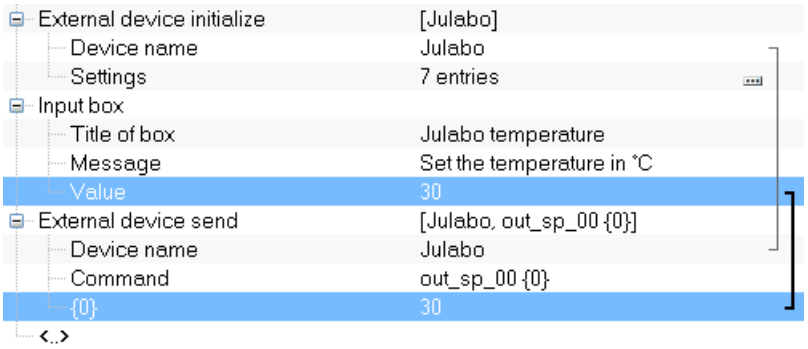

Figure 51 – Adding a linkable parameter to the data string

To create a linkable parameter in the data string, replace the value in the string by **{x}**, where **x** is an integer between 0 and 4. An additional parameter, **{x}** will be added to the command. This new parameter is linkable. The example shown in Figure 51 illustrates the use of this feature. The **{0}** parameter is linked to the value defined in the Input box.

4.3 – External device receive

This command can be used to receive a data string from the external device.

To receive a string from the external device, the External device send is first used to send a specific string to the external device. The External device receive command is then added to the procedure to read the reply string from the external device (see Figure 52).

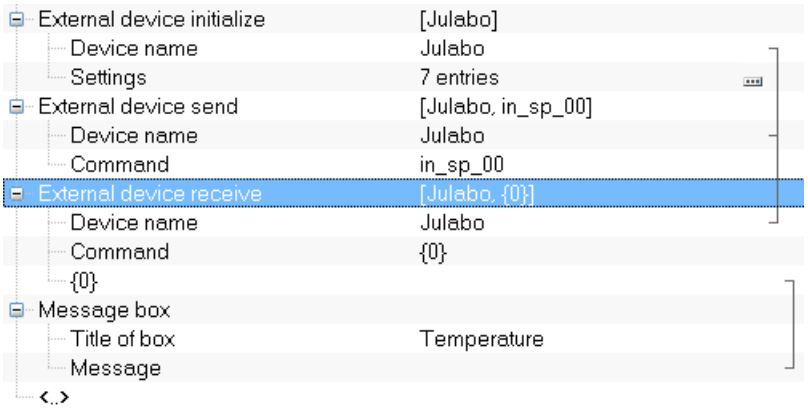

Figure 52 – Use the External device receive to wait for a data string from the external device

The example shown in Figure 52 can be used to request the temperature of the Julabo HE/F34 water bath. The answer will be stored in the {0} parameter, which can be displayed in a message box, as shown in Figure 52.

Note: the Device name must be the same as the one defined in the External device initialize command.

4.4 – External device close

This command must **always** be used to terminate the connection to the external device. To terminate a connection, add the External device close command to the procedure and define the Device name to terminate the remote control of that device.

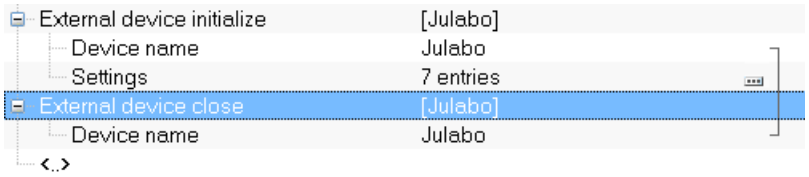

Figure 53 – The External device close command must always be used to release the control of the External device

Appendix – Modification of the Config.xml file for Swing arm control

Depending on the type of Robotic arm mounted on the 786 Swing arm, specific settings in the Config.xml configuration file must be adjusted in order to use the Swing arm correctly.

Presently, the following standard Robotic arms are supported (see Figure 54).

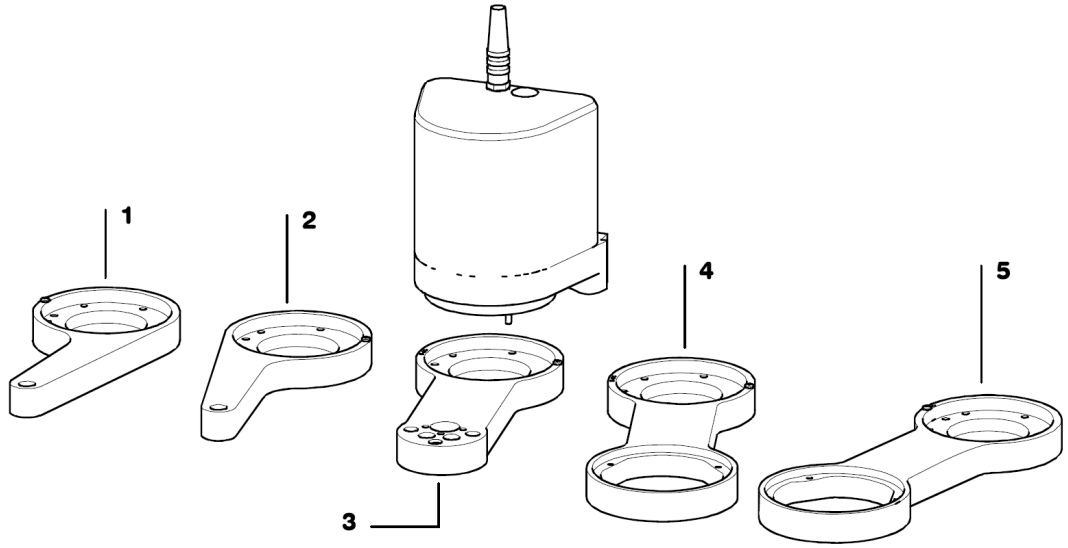

Figure 54 – Overview of the supported standard Robotic arms: 1 – 6.1462.030, 2 – 6.1462.040, 3 – 6.1462.050, 4 – 6.1462.060, 5 – 6.1462.070

Table 1 provides an overview of the specifications of the different standard Robotic arms.

Metrohm code	6.1462.050	6.1462.050	6.1462.050	6.1462.050	6.1462.050
Swing/Angle offset	8°	8°	0°	-8°	-8°
Max. swing angle/range	117°	117°	84°	73°	73°
Swing radius	112 mm	112 mm	110 mm	127 mm	127 mm
Swing direction	→	←	⇄	→	←

Table 1 – Overview of the standard Robotic arm settings

Table 2 provides an overview of the specifications of the different advanced Robotic arms.

Metrohm code	6.1462.080	6.1462.090	6.1462.150	6.1462.160	6.1462.170
Swing/Angle offset	8°	8°	0°	-8°	-8°
Max. swing angle/range	117°	117°	84°	73°	73°
Swing radius	112 mm	112 mm	110 mm	127 mm	127 mm
Swing direction	→	←	⇄	→	←

Table 2 – Overview of the advanced Robotic arm settings

To edit the *Config.xml* file used to control the Sample processor, the following steps must be followed.

1. Connect the Sample processor to the computer using the provided USB cable. Position the sample rack on the Sample processor and switch the device on.
2. Start NOVA.
3. Setup the Sample processor settings as described on page 15. Run the following procedure once (see Figure 55).

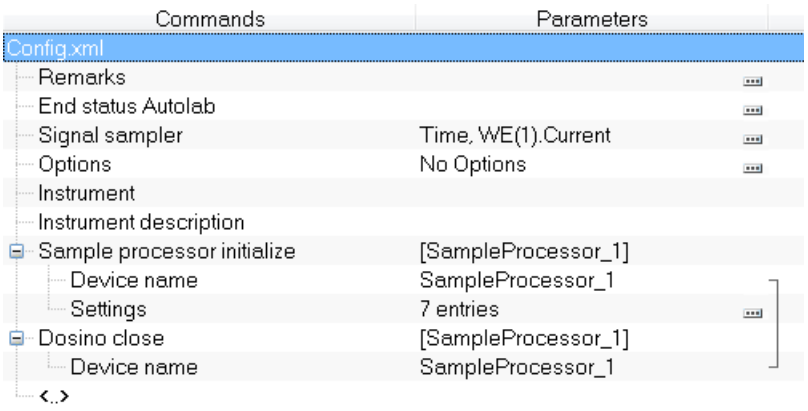

Figure 55 – The procedure required to read the Config.xml file from the Sample processor

The procedure shown in Figure 55 is used to initialize the Sample processor once and to read the *Config.xml* file from the device. This file is stored in the location provided in the Sample processor settings window (see Figure 56).

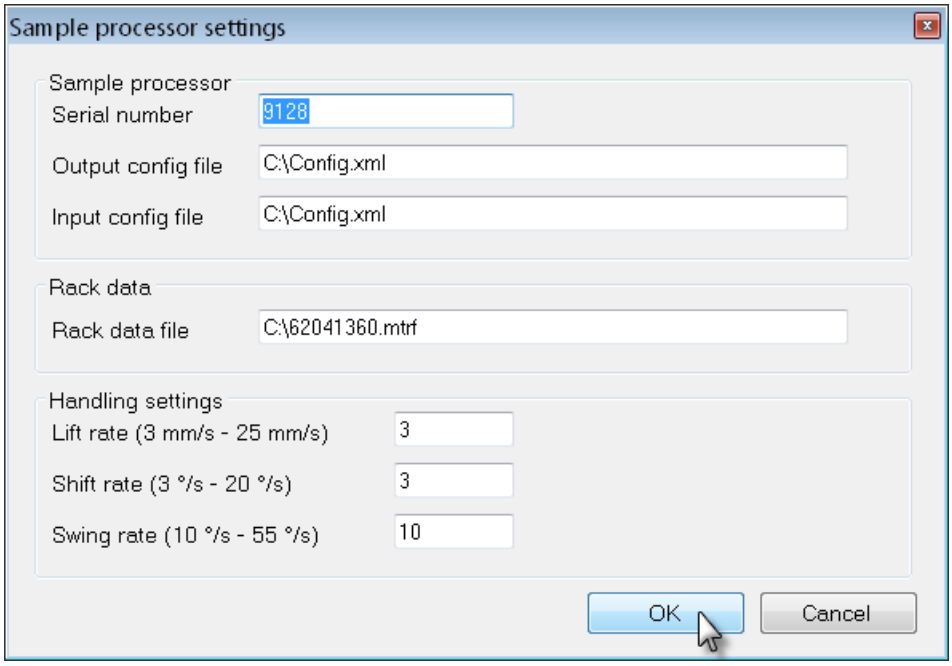

Figure 56 – The Sample processor settings window is used to define the location of the Config.xml file

4. Once this procedure is finished, use a text editor (for example Notepad), to edit the Config.xml file (see Figure 57).

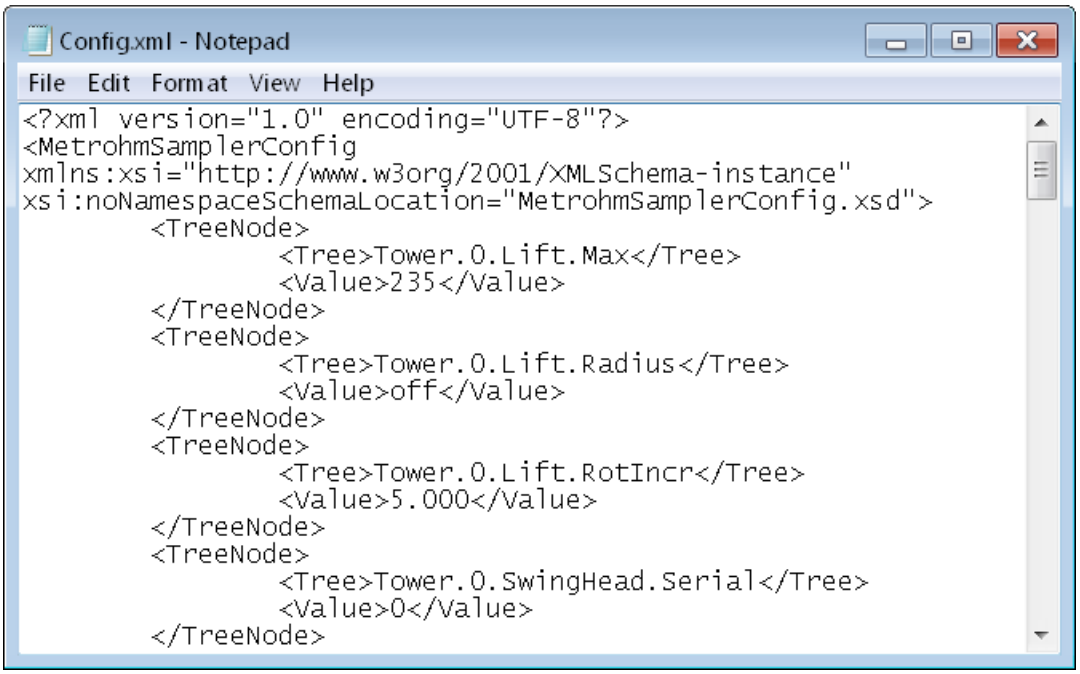

Figure 57 – Open the Config.xml file with a text editor

5. In the *Config.xml* file, locate the following items:

Swing/Angle offset

```
<TreeNode>
<Tree>Tower.0.SwingHead.Offset</Tree>
<Value>0.0000</Value>
</TreeNode>
```

Replace the value in bold lettering by the value listed in Table 1 or Table 2, for the Robotic arm used with the Swing head. If the value is negative, specify this value with the - sign.

Max. swing angle/range

```
<TreeNode>
<Tree>Tower.0.SwingHead.MaxRange</Tree>
<Value>60.0000</Value>
</TreeNode>
```

Replace the value in bold lettering by the value listed in Table 1 or Table 2, for the Robotic arm used with the Swing head.

Swing radius

```
<TreeNode>
<Tree>Tower.0.SwingHead.Radius</Tree>
<Value>110.0000</Value>
</TreeNode>
```

Replace the value in bold lettering by the value listed in Table 1 or Table 2, for the Robotic arm used with the Swing head.

Swing direction

```
<TreeNode>
<Tree>Tower.0.SwingHead.Dir</Tree>
<Value>-</Value>
</TreeNode>
```

Replace the sign in bold lettering by a + or - sign, depending on swing direction. The following convention is used to specify the swing direction:

- + is used when the swing direction is clockwise
- - is used when the swing direction is anti-clockwise

6. Save the changes as a new xml file, for example, *Config_modified.xml.*

7. The new xml file can now be used in the *Sample processor initialize* command. Define the **Input config file** in the Sample processor settings window as the new xml (in Figure 58, *Config_modified.xml).*

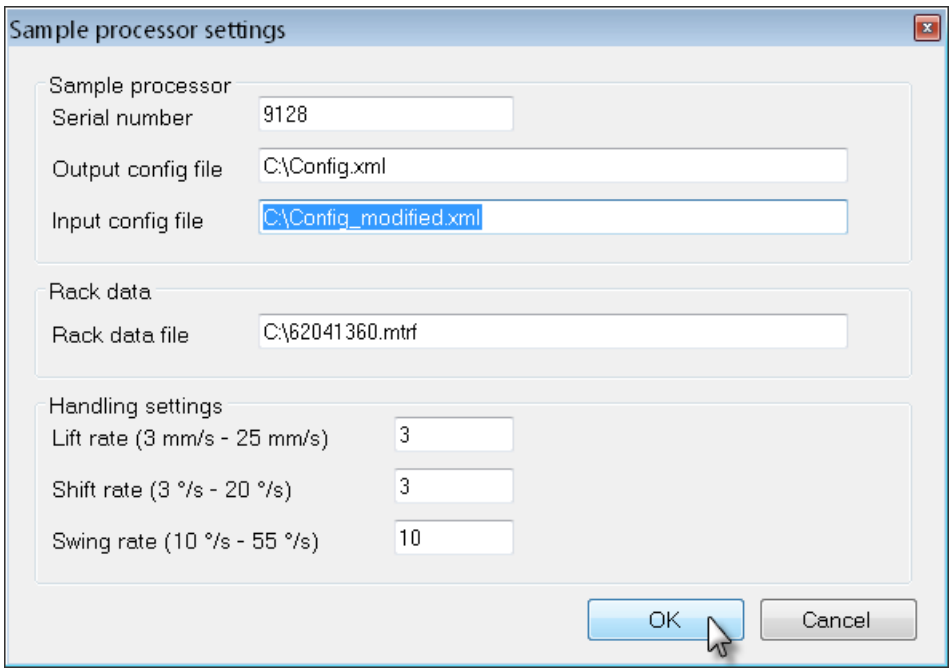

Figure 58 – The new *Config.xml* file can now be used as the Input config file

When the Sample processor is initialized again, the settings for the Swing head and the Robotic arm defined in the edited xml file will be used.