

Skalar Analytical B.V.

Verkooijen, Corné

August 29, 2017

Revision 2.3 (under
construction)

Bluvision II

communication

protocol between PC

and BluVision

1.1Contents

2.1 Explanation Actual situation (03-05-2017): Document version 1.14.....	4
2.2 Way of working of (new concept) scheduler Agaram.....	5
2.3 Proposal to improve command communication between DA and BluVision	5
2.3.1 Explanation ActionStep element:	8
2.3.2 Explanation Measure element within ActionStep element:.....	9
2.3.3 Explanation Unload element within ActionStep element:	10
2.3.4 Explanation Load element within ActionStep element:	10
2.3.5 Response from BluVision to Discrete Access:.....	10
2.4 Get actual time stamp of running scheduler	12
3 Standard commands without schedule queue dependency	12
3.1 Status Cuvette Array	12
3.1.1 Get Status of Cuvette array from PC to BV	12
3.1.2 Response status Cuvette array from BluVision to PC:	12
3.1.3 Get Status of All Cuvette arrays from PC to BV.....	14
3.2 Get Waste Bin Status	15
4 Level detection of liquids before starting a run:.....	16
5. FilterWheel.....	17
5.1 loading filter wheel	17
5.2 filter check by user.....	17
6. Temperature control BluVision.....	18
6.1 Set temperature parameters	18
6.2 Get temperature parameters (setpoints)	18
6.3 Get temperature values (actual values).....	19
8 Notification to LCD display of BluVision.....	20
9 Set system time BluVision mainboard	21
10 Pause and Resume Bluvision in running state including cover check.....	22
11 Status check covers	23
12 Update report window (append / remove errors to/ from list)	24
13 Communication DLL Skalar (draft only, not released!!).....	26
13.1 Device discovery over Ethernet with SSDP protocol (uPNP) Draft Only not released!!!.....	28
14 BluVision states	29
16.0 Firmware implementation: Parallel measurement, load and unload while sample pickup.....	31
16.1 New functionality: parallel movement / measurement during run.	32
16.2 Scheduler for parallel action's like measurement, loading and unload of cuvettes Firmware side.	33
16.3 Scheduler actions From Discrete Access (Software side).	33
20 Obsolete commands (not in use anymore since communication version 2.0):.....	34

20.0 AddToQueue Commands	34
20.1 Commands type="WashToSample"	34
20.2 Commands type="TakeSample"	35
20.3 Commands type="SampleToCuv">	35
20.4 Commands type="DispenseSample"	36
20.5 Commands type="MoveToWash"	37
20.6 Commands type="Rinse"	37
20.7 Commands type="CuvToSample"	39
2.8 Commands type="MeasureCel"	40
20.9 Commands type="LoadCuvette"	40
20.10 Commands type="UnloadCuvette"	41
20.11 Commands type="CheckCuvBlock"	41
20.12 SampleToSample (for dilutions)	42
20.13 SampleMixing (for dilutions)	42
Obsolete part 15.0 Duration times for generating faster schedule queues	43
15.1 Get duration information from BV	43
15.2 Response duration information from BV to DA	43
21 Obsolete part: Recalculation of Schedule queue	45
Revision History	47
Update to paragraph 2.3.5 Response from BluVision to Discrete Access:	49
Update to paragraph 2.3.* Response from BluVision to Discrete Access:	49
Update to paragraph 2.3.* add functionality to do a pre rinse before take sample in main scheduler	49

2.0 Scheduler improvement / accuracy.

Discrete Access has problems with the prediction of a right queue in real time. This causes delay times while the machine could run. This is unwanted behaviour. Also the firmware has some weak points with this open loop control mechanism. To get more control in this scheduling / execution part, an update in the way of working is required!

2.1 Explanation Actual situation (03-05-2017): Document version 1.14

The execution of the scheduler in the firmware will lose critical time because it is not possible to define the exact time of execution by the DA software. The time variation is caused by longer travel distance of ZArm / Arm Rotate, large cuvette movements between measurements and dispense.

	Predicted time of execution by DA	Actual execution time by BluVision	Total duration time (timestamp)	Lost time out of sync	Parallel activities
WashToSample	100	80	100	20	Measure Cel
TakeSample	100	120	120	0	
MoveToCuv	100	140	140	0	
Dispense	100	70	100	30	
MoveToWash	100	90	100	10	
Rinse	100	100	100	0	
WashToSample	100	80	100	20	Measure Cel
TakeSample	100	120	120	0	
MoveToCuv	100	140	140	0	
Dispense	100	70	100	30	
MoveToWash	100	90	100	10	
Rinse	100	100	100	0	
Total time	1200	1200	1320	120	

- When the predicted time is shorter than the actual time, the bluvision is introducing a latency on the predicted measurement / dispense sample
- When the predicted time is larger than the actual time, the bluvision is waiting in expensive time.
- There is no information in the bluvision available to cross check if a measurement took place on time. This is the same for dispense.
- The execution is in **relative** time perspective. If the PC is too late with sending the data to the Bluvision, is will introduce a shift of execution which is later than expected.

2.2 Way of working of (new concept) scheduler Agaram.

Fase 1: collect samples from workload. These are the (approx.. 15 samples) which will be handled by the machine within a certain time slot. Input depended on user: method wise or rack wise.

This fase is executed at start and recalled after 50% execution of the previous Fase 1 action.

Fase 2a: Samples from fase 1 are divided in steps (conditional (incubation time related) and unconditual (free time , pickup a sample. No reaction take place.)).

Calculation take place at start for 2 Steps, after this 7 seconds before the last action is executed.

This action is waiting for the received unique ID of the bluvision. This calculation is only using predicted times, so no correction is made if machine uses more time to execute a step.

Fase 2b: Fitting of steps in the (basic) scheduler according to S+

Fase 2C: generate XML queue from selected steps in fase 2.

Communication with Bluvision is asynchrony. The received commands / information will be parsed on a separate thread. Data from this translation is stored in the database.

Scheduler will take status of bluvision from the database.

Fase 3: send commands to BV over Ethernet.

Fase 4: receive executed commands back from BV over Ethernet.

Possible issues:

- no feedback is used to correct missing time. Possible introduction of cumulative faults.

Once it out of sync, no correction / notification take place! This is an open loop control system.

2.3 Proposal to improve command communication between DA and BluVision

Proposal combine the commands to the bluvision and set **absolute** timings to do the execution of the time critical part of dispense and measure cuvette cell.

Improvements:

Combine commands like WashToSample, TakeSample, MoveToCuvette and dispense and MoveToWash, rinse in one recipe / step. This will give some benefits in reduction of commands and add more intelligent to BluVision by adding step / method name.

Benefits: DiscreteAccess can handle the duration time of this step in the calculation of the next similar method step.

Adding **absolute** time counter (counting from unique id 1) for dispense and measurement command. In this case the bluvision can calculate the difference between predicted and real executed time. For dispense it will wait until the time expired if the BluVision machine is on time.

Benefits: keep dispense / measure action on time as predicted in scheduler.

Tasks for Discrete access: Create an action block for each sample step

Benefits: keep measure action on time as set in method in scheduler. But needs to be aligned with DA because if collision in dispense action.

Discrete Access is still in control of making decision on what point dispense and measurement take place!!!

So, one total combined movement command with: WashToSample, TakeSample, MovetoCuv, Dispense, MoveToWash and Rinse. Let's call this an **ActionStep**

Theoretically, there are two actions which must met exactly on time: Measurement after adding reagent and dispense of a second reagent.

For dispense there will be an **absolute** dispense time given by the Discrete Access scheduler. The scheduler execution in the Bluvision will start dispense at this time point. DA will not schedule 2 measure / dispense events within #n Seconds of time. If there is no absolute time for dispense available (parameter ExeTime = 0), the Bluvision will dispense immediately without waiting.

If it is not possible to meet the required time: **The latency will be notified by the Bluvision to Discrete Access.**

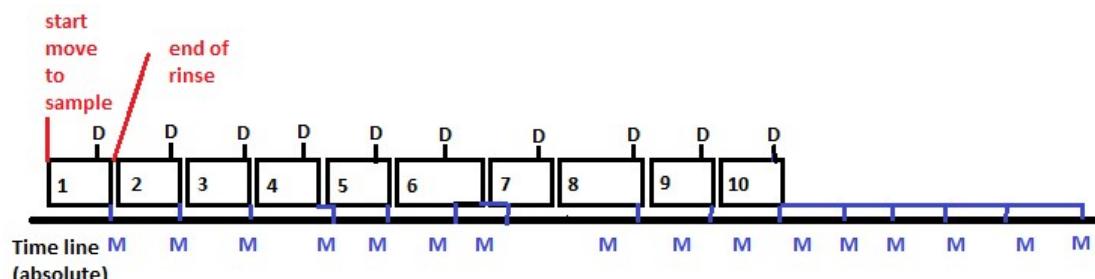
Each Action Block will send the duration time afterwards to the DA software. **Each duration time of step execution can be used by the database to calculate the rest of the schedule. This will make the schedule adaptive / self learning. Off course some extra time should be taken in account to give the bluvision space to operate.**

Task for Bluvision:

If a measurement and dispense or multiple measurements are scheduled in this action time slot, bluvision will execute the item with the first time point that expires.

After the execution of dispense / measurement, the Bluvision notifies DA directly with dispense / measurement time. After the execution of the last step (rinse), Bluvision notifies DA directly with the duration time of this entire step.

The Bluvision firmware can store dispense / measurement time points for each cuvette cell. The differences in timestamps can be used as cross check to validate the incubation time.



In the picture above there are 10 pickup and dispenses of samples. The D stand for dispense at a specific time, scheduled by DA. The M stand for Measure. The measure command execution can take place even after the rinse took place. It will just wait for the next expire time. If a new **Actionstep** is started after the rinse. It is possible that the sample is ready to dispense but the measure action in the buffer still must be handled. In this case the bluvision wait until the measurement took place. After that it will move in the cuvette position and dispense

Send from DA to BluVision XML it looks like:

Initial opening tags:

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
.... ActionSteps....
</AddToQueue >
```

Example 1: standard sample / reagent pickup

Take 5 µL Air (speed 10) + Take 400 µL sample (speed 8) from A1 without pre dilution with diluter speed , no air in between rotation, and dispense 350 µL at 13000 mSec after start scheduler it in cuvette without stir movement of needle. After this rinse needle with 1200 µL demiwasser (speed 6) on inside needle and 1300µL demiwasser on outside of the needle. After 15000 mSec from start scheduler Cuv02Cel03M will be measured with Filter 1 and gain 0. Sample reagent disk will move after take action to position A2

```
<ActionStep ID="101" Text="UnknownA1" SPos="A1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D" AirPre="5" AirAfter="0" AirDilSpd="10" STDilSpd="8" CDilSpd="5" STVol="400"
CDVol ="350" Stir="0" RinIn="1200" RinOut="1300" RinDilSpd="10" ExeTS ="3500" Dur="12000">
<Measure ID ="2" Text ="Stock20" CPos="Cuv02Cel03M"Gain="0" Fil="Fil1" NFil="Fil2" ExeTS
="15000"/>
<ActionStep />
```

Example 2: multiple measurements in one action step

Take 5 µL Air (speed 10) + Take 400 µL sample (speed 8) from A1 without pre dilution with diluter speed , no air in between rotation, and dispense 350 µL at 13000 mSec after start scheduler it in cuvette without stir movement of needle. After this rinse needle with 1200 µL demiwasser (speed 6) on inside needle and 1300µL demiwasser on outside of the needle. After 15000 mSec from start scheduler Cuv02Cel03M will be measured with Filter 1 and gain 0. After 18000 mSec from start scheduler Cuv02Cel03M will be measured with Filter 1 and gain 0. After 20000 mSec from start scheduler Cuv02Cel03M will be measured with Filter 1 and gain 0.Sample reagent disk will move after take action to position A2

```
<ActionStep ID="101" Text="UnknownA1" SPos="A1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D" AirPre="5" AirAfter="0" AirDilSpd="10" STDilSpd="8" CDilSpd="5" STVol="400"
CDVol ="350" Stir="0" RinIn="1200" RinOut="1300" RinDilSpd="10" ExeTS ="3500" Dur="12000">
<Measure ID ="2" Text ="Stock20" CPos="Cuv02Cel03M"Gain="0" Fil="Fil1" NFil="Fil2" ExeTS
="15000"/>
<Measure ID ="3" Text ="Stock21" CPos="Cuv02Cel03M"Gain="0" Fil="Fil1" NFil="Fil2" ExeTS
="18000"/>
<Measure ID ="4" Text ="Stock22" CPos="Cuv02Cel03M"Gain="0" Fil="Fil1" NFil="Fil2" ExeTS
="20000"/>
<ActionStep />
```

Example 3: creation working standard in cuvette from stock standard pickup

Take 5 µL Air (speed 10) + Take 300 µL Demiwasser (diluent) (speed 10) Take 200 µL sample (speed 8) from A1 without pre dilution with diluter speed , no air in between rotation, and dispense 350 µL at 13000 mSec after start scheduler it in cuvette without stir movement of needle. After this rinse needle with 1200 µL demiwasser (speed 6) on inside needle and 1300µL demiwasser on outside of the

needle. After 15000 mSec from start scheduler Cuv02Cel03M will be measured with Filter 1 and gain 0. Sample reagent disk will move after take action to position A2

```
<ActionStep ID="101" Text="WS1" SPos="A1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D" DemiDilSpd="10" DemiDilVol="300" AirPre="5" AirAfter="0" AirDilSpd="10" STDilSpd="8" CDilSpd="5" STVol ="200" CDVol ="350" Stir="0" RinIn="1200" RinOut="1300" RinDilSpd="10" ExeTS ="3500" Dur="12000">
<Measure ID ="2" Text ="Stock20" CPos="Cuv02Cel03M"Gain="0" Fil="Fil1" NFil="Fil2" ExeTS ="15000"/>
<ActionStep />
```

Example 4: Level detection Reagent R1 + unload cuvette 2 and load cuvette 2 at same time.

Take 0 µL sample (speed 8) from A1 without pre dilution with diluter speed , no air in between rotation, and dispense 0 µL at freetime if STVol = 0 and CDVol = 0 this indicates a levldection Rinse needle with 1200 µL demiwatet (speed 6) on inside needle and 1300µL demiwatet on outside of the needle. In freetime Cuv02 is unloaded. If level detection is done in R1 the cuvettedisk will turn to A2. If first unloading and then loading is finished, the cuvette disk is turning to Cuv01Cel03D.

```
<ActionStep ID="103" SPos="R1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D" AirPre="5" AirAfter="5" STVol ="0" CDVol ="0" Stir="0" RinIn="1200" RinOut="3000" ExeTS ="0" Dur="12000">
<Unload ID ="2" CPos ="Cuv02U" ExeTS="0" WasteBin="1" />
<Load ID ="2" CPos ="Cuv02L" ExeTS="0"/>
<ActionStep />
```

Example 5: creation dilution sample in samplecup from unknown overrange sample

Take 5 µL Air (speed 10) + Take 6500 µL Demiwatet (diluent) (speed 10) Take 200 µL sample (speed 8) from A1 , no air in between rotation, and dispense 6650 µL in freetime sample cup it in sample cup B2 without stir movement of needle. But it wil mix (Mix="15") for 15 times a full small diluter movement to mix the volume by moving to top op sample, take full 500 uL volume and dispense it with (CDilSpd) dilution speed 5 . After this rinse needle with 1200 µL demiwatet (speed 6) on inside needle and 1300µL demiwatet on outside of the needle.. Sample reagent disk will move after take action to position A2 No measurement take place in this example.

```
<ActionStep ID="101" Text="Dilution Sample Unknown A1" SPos="A1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D" SDilPos="B2" DemiDilSpd="10" DemiDilVol="6500" AirPre="5" AirAfter="0" AirDilSpd="10" STDilSpd="8" CDilSpd="5" STVol ="200" CDVol ="350" Stir="0" Mix="15" RinIn="1200" RinOut="1300" RinDilSpd="10" ExeTS ="0" Dur="12000">
<ActionStep />
```

2.3.1 Explanation ActionStep element:

An **ActionStep** element is one movement from rinse position to take a sample (possible with dilution) and dispense it in a cuvette cup.

Atributes:

ID="103" Identification number of actual **ActionStep**

Text="Text to show on tft screen for pending action" [max 32 ASCII characters]

SPos="A1" Take sample position

SNPos="A2" S/R disk will turn to this position after take sample

SDilPos="B2" S/R dispense dilution in sample cup (*optional attribute, if this is available no cuvette dispense will take place!*)

CPos="Cuv01Cel02D" Dispense sample position

CNPos="Cuv01Cel03D" Cuvettdisk next position after all executions in this block

AirPre="5" Air volume to take before entering sample cup in μL

AirAft="5" Air volume to take after leaving sample cup in μL

AirDilSpd="10" Speed [1-10]

STVol="400" Liquid volume to take from sample cup

In case of sample cup to sample cup dilution **STVol** is equal to the dispense volume in the next sample cup. So no extra volume can be taken!

DemiDilSpd="10" Dilution speed take dilution (*optional attribute, only required in combination with DemiDilVol!*)

DemiDilVol="200" Dilution DemiWater take volume inside needle in μL volume demiwatert (*optional attribute, if this is not available no dilution will take place!*)

CDVol="350" Liquid volume to dispense in cuvette in μL

Stir="0" Stir time in milli seconds in cuvette cup after / while dispensing liquid

RinIn="1200" DemiWater rinse volume inside needle in μL volume demiwatert

RinOut="3000" DemiWater rinse volume outside needle in μL volume demiwatert

RinDilSpd="10" Dilution speed rinse inside neelde

ExeTS="3500" Absolute execute timestamp to dispense liquid in Cuvette cup. If this time is set to 0 than it will execute directly without waiting.

Dur="12000" Expected duration time of Action step predicted by DA from start Actionstep until finish rinsing.

SDilSpd="10" Dilution speed dispense in sample cup during dilution in sample cup

STDilSpd="10" Dilution speed sample take in sample cup during take sample

CDilSpd="10" Dilution speed dispense in cuvette during normal dispense in cuvette

New since v2.3:

PreRinIn="1200" First rinse neelde before take sample with DemiWater rinse volume inside needle in μL volume demiwatert

PreRinOut="3000" First rinse neelde before take sample with DemiWater rinse volume outside needle in μL volume demiwatert

2.3.2 Explanation Measure element within ActionStep element:

<Measure ID ="2" CPos ="Cuv02Cel03M" Gain="0" Fil="Fil1" NFil="Fil2"

ExeTS="10000" Text="Measure Unknown1"/>

Atributes:

ID="2" Identification number of actual Measure action

CPos="Cuv01Cel02M" Measure cuvette position

Gain="0" Gain settings:

0 = 000 = 1x
1 = 001 = 2x
2 = 010 = 4x
3 = 011 = 8x
4 = 100 = 16x
5 = 101 = 32x
6 = 110 = 64x
7 = 111 = 64x

Fil="Fil1" filter position in filterwheel
NFil="Fil2" next filter position in filterwheel

ExeTS ="123339" is absolute execute timestamp of execution of this action by Bluvision. if parameter = 0 than it will be executed directly
Text="Text to show on tft screen for pending action" [max 32 ASCII characters]

2.3.3 Explanation Unload element within ActionStep element:

<Unload ID ="2" CPos ="Cuv02U" ExeTime="10000" Text="Unload Cuv1 in Waste bin 1"/>

Atributes:

ID="2" Identification number of actual Measure action
CPos="Cuv01U" Cuvette to unload position
ExeTS ="123339" is absolute execute timestamp of execution of this action by Bluvision
Text="Text to show on tft screen for pending action" [max 32 ASCII characters]
WasteBin="1" select the waste bin to eject to [1-2]

2.3.4 Explanation Load element within ActionStep element:

<Load ID ="2" CPos ="Cuv02L" ExeTime="10000" Text="Loading Cuvette pos 1"/>

Atributes:

ID="2" Identification number of actual Measure action
CPos="Cuv01L" Load cuvette position
ExeTS="123339" is absolute execute timestamp of execution of this action by Bluvision if parameter = 0 than it will be executed directly
Text="Text to show on tft screen for pending action" [max 32 ASCII characters]

2.3.5 Response from BluVision to Discrete Access:

For measurement:

<**Executed** ID="103" Type="M" Ready="11900" Start ="12000" End ="14000" LightSensor="153255" CorrSensor="153151" TempSensor="15185" **Delay**="100"/>

Type : M = measurement result
Ready = absolute time that measurement was ready to start.
Start = absolute time that measurement started.

End = absolute time that measurement ended.

Delay = executed with 100# milliseconds of delay.

For dispense:

<Executed ID="104" Type="D" Ready="11900" Start="12000" End="14000" Delay="100"/>

Type : D = dispense

Ready = absolute time that dispense was ready to start dispense

Start = absolute time that dispense started.

End = absolute time that dispense ended.

Delay = executed with 100# milliseconds of delay.

For load cuvette:

<Executed ID="105" Type="L" Ready="11900" Start ="12000" End ="18000" Delay="100"/>

Type : L = Load cuvette block

Ready = absolute time that Load was ready to start

Start = absolute time that Load started.

End = absolute time that Load ended.

Delay = executed with 100# milliseconds of delay.

For unload cuvette:

<Executed ID="106" Type="U" Ready="11900" Start ="12000" End ="18000" Delay="100"/>

Type : U = unload cuvette block

Ready = absolute time that Load was ready to start

Start = absolute time that unload started.

End = absolute time that unload ended.

Delay = executed with 100# milliseconds of delay.

<Executed ID="106" Type="R" Ready="11900" Start ="12000" End ="18000" Delay="100"/>

Type : R =Rinse

Ready = absolute time that Load was ready to start

Start = absolute time that Rinse started.

End = absolute time that Rinse ended.

Delay = executed with 100# milliseconds of delay.

<Executed ID="106" Type="A" Ready="12000" Start ="12000" End ="18000" Delay="100"/>

Type : A =ActionStep time

Ready = absolute time that Action step was ready to start

Start = absolute time that Action step started.

End = absolute time that Action step ended.

Delay = executed with 100# milliseconds of delay.

2.4 Get actual time stamp of running scheduler

Retrieve the actual time stamp of the running to synchronize the scheduler in the PC this command can be used by Discrete Access

```
<?xml version="1.0" encoding="utf-8"?>
<GetTimeStamp/>
```

Response from BluVision

```
<?xml version="1.0"?>
<Status Timestamp="1502155">
```

3 Standard commands without schedule queue dependency

3.1 Status Cuvette Array

3.1.1 Get Status of Cuvette array from PC to BV

Retrieve **status of Cuvettes** from PC to BluVision: (can be useful at initialisation state) (is accessible directly after connection is made with pc)

```
<?xml version="1.0" encoding="utf-8"?>
<GetStateCuv>Cuv01</GetStateCuv>
```

3.1.2 Response status Cuvette array from BluVision to PC:

If a new Cuvette in position 1 is **loaded** the BV will send **automatically** this update :

```
<?xml version="1.0"?>
<Status Name="CuvetteDisk">
  <Cuv01 Cel01="E" Cel02="E" Cel03="E" Cel04="E" Cel05="E" Cel06="E" Cel07="E" Cel08="E"
  Cel09="E" Cel10="E" />
</Status>
```

If an used Cuvette in position 1 is **unloaded** the BV will send **automatically** this update :

```
<?xml version="1.0"?>
<Status Name="CuvetteDisk">
  <Cuv01 />
</Status>
```

This will return as response after **CheckCuvBlock** from BV to PC : (**Legenda** Empty="E" Full="F"

Checked="C")

```
<?xml version="1.0"?>
```

```
<Status Name="CuvetteDisk">
  <Cuv01 Cel01="C" Cel02="C" Cel03="C" Cel04="C" Cel05="C" Cel06="C" Cel07="C" Cel08="C"
Cel09="C" Cel10="C" />
</Status>
```

If you start a run and you use only the first 5 cels of the Array. (You still can use the last 5 cels) this will be the response after a request **GetStateCuv**

```
<?xml version="1.0"?>
<Status Name="CuvetteDisk">
  <Cuv01 Cel01="F" Cel02="F" Cel03="F" Cel04="F" Cel05="F" Cel06="C" Cel07="C" Cel08="C"
Cel09="C" Cel10="C" />
</Status>
```

3.1.3 Get Status of All Cuvette arrays from PC to BV

To retrieve all cuvette status information at once, use this command: response after a request

```
<?xml version="1.0" encoding="utf-8"?>
<GetStateAllCuv/>
```

Feedback from BV:

-New feedback for status cuvettedisk

```
<?xml version='1.0'?>
<Status Name="CuvetteDisk">
  <Cuv01 Cel="EEEEEEEEEE"/>
  <Cuv02 Cel="EEEEEEEEEE"/>
  <Cuv03 Cel="EEEEEEEEEE"/>
  <Cuv04 Cel="EEEEEEEEEE"/>
  <Cuv05 Cel="EEEEEEEEEE"/>
  <Cuv06 Cel="EEEEEEEEEE"/>
  <Cuv07 Cel="EEEEEEEEEE"/>
  <Cuv08 Cel="EEEEEEEEEE"/>
  <Cuv09 Cel="EEEEEEEEEE"/>
  <Cuv10 Cel="EEEEEEEEEE"/>
  <Cuv11 Cel="EEEEEEEEEE"/>
  <Cuv12 Cel="EEEEEEEEEE"/>
  <Cuv13 Cel="EEEEEEEEEE"/>
  <Cuv14 Cel="EEEEEEEEEE"/>
  <Cuv15 Cel="EEEEEEEEEE"/>
  <Cuv16 Cel="EEEEEEEEEE"/>
</Status>
```

```
<?xml version='1.0'?>
<Status Name="CuvetteDisk">
  <Cuv01 Cel="FFFFFEEEEE"/> // F means Cuvette is FULL
  <Cuv02 Cel="EEEEEEEEE"/> // E means Cuvette is EMPTY
  <Cuv03 Cel=" "/> // MEANS no Cuvette in Cuvette wheel
  <Cuv04 Cel="EEEEEEEEEE"/>
  <Cuv05 Cel="EEEEEEEEEE"/>
  <Cuv06 Cel="EEEEEEEEEE"/>
  <Cuv07 Cel="EEEEEEEEEE"/>
  <Cuv08 Cel="EEEEEEEEEE"/>
  <Cuv09 Cel="EEEEEEEEEE"/>
  <Cuv10 Cel="EEEEEEEEEE"/>
  <Cuv11 Cel="EEEEEEEEEE"/>
  <Cuv12 Cel="EEEEEEEEEE"/>
  <Cuv13 Cel="EEEEEEEEEE"/>
  <Cuv14 Cel="EEEEEEEEEE"/>
  <Cuv15 Cel="EEEEEEEEEE"/>
  <Cuv16 Cel="EEEEEEEEEE"/>
</Status>
```

3.2 Get Waste Bin Status

Internally the BV will record the amount of available cuvettes in the waste bin. After opening the waste bin, the user is asked if they empty the waste bin. Both Bin's can handle 64 cuvettes in total 128 pcs.

Command to retrieve wastebin status from DA to BV:

```
<?xml version='1.0'?>
<GetWasteBinStatus/>
```

Command to update waste bin status from BV to DA :

The value gives the available spaces left to eject the cuvettes

```
<?xml version='1.0'?>
<WasteBinStatus>
    <WasteBin1>64</WasteBin1>
    <WasteBin2>64</WasteBin2>
    <EmptyTime>###Timestamp###</ EmptyTime >
</WasteBinStatus>
```

4 Level detection of liquids before starting a run:

Sequence:

- Move from wash position Zarm Up
- [parallel movement]
- Rotate arm to Sample Reagent ring position
- Rotate Sample/reagent disk to position
- Wait until SR disk + Arm rotate in position
- [end parallel movement]
- Zarm down until level detection reaches point of liquid
- Zarm up
- [parallel movement]
- Rotate arm to Wash position
- Rotate Sample/reagent disk to next position
- Wait until Arm rotate in position
- [end parallel movement]
- ZDown and start Rinsing sequence pumps and valves with defined Time

Obsolete command:

```
<Commands type="GetLevelCup">
<UniqueId>1</UniqueId>
<Pos>R1</Pos>
<NextPos>B2</NextPos>
<Time>500</Time>
<DurationTime>2000</DurationTime>
</Commands>
```

If VolTake == 0 and VolDis == 0 than the arm movement will act as a level detection in SR disk.

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
<ActionStep ID="103" SPos="R1" SNPos="A2" CPos="Cuv01Cel02D" CNPos="Cuv01Cel03D"
AirPre="5" AirAfter="5" STVol ="400" CDVol ="350" Stir="0" RinIn="1200" RinOut="3000" ExeTS
="0" Dur="12000">
<ActionStep />
</ AddToQueue >
```

Pos = position of Sample / reagent disk. This can be a sample or reagent cup

NextPos = next position of sample reagent disk to improve speeds

Time = rinsing time after a level detection.

Response:

```
<?xml version="1.0"?>
<Level Name="R1" Volume="100"/>
```

It returns the volume back in uL this value is caluclated from the internal table in the BluVision.

5. FilterWheel

5.1 loading filter wheel

It is possible to go manually to a filter wheel position in idle state by pressing the settings tab -> filter wheel.

If the DA software want to go to a certain position (physical movement of filter wheel disk) and to update the wavelength parameter value, DA software has to send this command:

It is only possible to move the filter wheel by DA software if the system is in IDLE state:

```
<?xml version="1.0" encoding="utf-8"?>
<LoadFilter Pos="Fil1" Wave="540" Enable="1"/>
```

If no Filter in position is needed: (Wave will be ignored)

```
<?xml version="1.0" encoding="utf-8"?>
<LoadFilter Pos="Fil2" Wave="660" Enable="0"/>
```

5.2 filter check by user

It is necessary to check the wavelength of the filter by user because there is no sensor / input on the filterwheel if a sensor is available or the right filter is used!

Conformation is needed by user to check if the right filter is at the right spot

To notify the user, the software should send a notification to the BV display also.

First send the load filter position, after that send the question to the BV (see chapter 8 Notification to LCD display of BluVision)

Example of communication flow DA to BV:

```
<?xml version="1.0" encoding="utf-8"?>
<LoadFilter Pos="Fil1" Wave="540" Enable="1"/>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Popup Msg="Please insert 540NM filter in position 1 of Filterwheel. Succeed?" Btn1="Yes" Btn2="" Btn3="No" Show="1"/>
```

Ask same question in DA software to user.

If user press on "Yes" on LCD display. BV will send:

```
<?xml version="1.0" encoding="utf-8"?>
<PopupPressed>Btn1</PopupPressed>

<?xml version='1.0'?>
<PopupPressed>
  <Button>Btn1</Button>
</PopupPressed>
```

If user press on "Yes" on PC DA display. DA will send to BV:

```
<?xml version="1.0" encoding="utf-8"?>
<Popup Show="0"/>
```

6. Temperature control BluVision

6.1 Set temperature parameters

Temperature Reagent wheel:

```
<?xml version="1.0" encoding="utf-8"?>
<SetTemperature>
  <SRDisk Temp="12.0" Enable="1"/>
</SetTemperature>
```

Temperature Cuvette disk:

```
<?xml version="1.0" encoding="utf-8"?>
<SetTemperature>
  <CuvetteDisk Temp="45.2" Enable="1"/>
</SetTemperature>
```

Temperature Needle heating:

```
<?xml version="1.0" encoding="utf-8"?>
<SetTemperature>
  <Needle Temp="40.5" Enable="1"/>
</SetTemperature>
```

Also possible to send it as once:

```
<?xml version="1.0" encoding="utf-8"?>
<SetTemperature>
  <SRDisk Temp="12.0" Enable="1"/>
  <Needle Temp="40.5" Enable="1"/>
  <CuvetteDisk Temp="45.2" Enable="1"/>
</SetTemperature>
```

6.2 Get temperature parameters (setpoints)

To retrieve the setpoints from the BluVision use this command:

```
<?xml version="1.0" encoding="utf-8"?>
<GetTemperature>
  <SRDisk/>
  <CuvetteDisk/>
  <Needle/>
</GetTemperature>
```

Or per Item is also possible:

```
<?xml version="1.0" encoding="utf-8"?>
<GetTemperature>
  <SRDisk/>
</GetTemperature>
```

Response of BluVision to DA software:

```
<?xml version="1.0" encoding="utf-8"?>
<GetTemperature>
  <SRDisk Temp="20.4" Enable="1"/>
  <CuvetteDisk Temp="20.4" Enable="1"/>
  <Needle Temp="20.4" Enable="1"/>
</GetTemperature>
```

6.3 Get temperature values (actual values)

Every 10 seconds, the BluVision will update the temperature status and send it to the PC.

These messages will be send automatically!

The DA software will **receive** these commands:

```
<?xml version="1.0" encoding="utf-8"?>
<Temperature>
  <SRDisk Temp="20.4"/>
  <CuvetteDisk Temp="40.4"/>
  <Needle Temp="30.4"/>
</Temperature>
```

8 Notification to LCD display of BluVision

To ask a question to the user to verify something like:

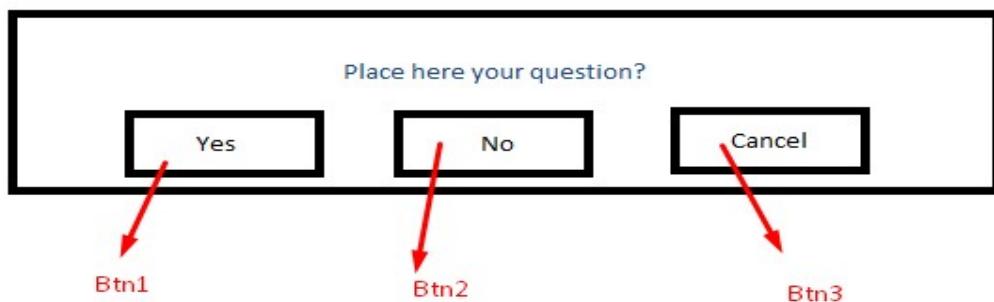
- Is the right filter in the filter wheel?
- Are there enough empty cuvettes available?
- Is there enough demineralized water available?
- Is the waste bin empty?

A popup can be used on the BluVision LCD.

To enable this use this command:

```
<?xml version="1.0" encoding="utf-8"?>
<Popup Msg="Place here your question?" MsgSub="Are you sure" Btn1="Yes" Btn2="No"
Btn3="Cancel" Show="1" Timeout="60"/>
```

In revision 5 of this document: **Timeout** is added: the value represents the time that the popup is showed in seconds. If the timeout expires, the popupbox hides by itself without any notification!



If the content of the button name is empty like `Btn1=""` the button is hided!

To hide entire popup from LCD screen send this command from DA software:

```
<?xml version="1.0" encoding="utf-8"?>
<Popup Show="0"/>
```

Response on button click from Bluvision to DA software:

```
<?xml version="1.0" encoding="utf-8"?>
<Popup_Pressed="Btn1"/>
<?xml version='1.0'?>
<PopupPressed>
  <Button>Btn1</Button>
</PopupPressed>
```

After a button click on the LCD, the popup will hide automatically!

2 row text on display is possible just add “\n” to text string and rest of the line will be showed on the next row!

9 Set system time BluVision mainboard

The BluVision Mainboard contains an internal Time / date clock. The only disadvantage of this clock is when the power is from the device, the timestamp will start at 2010 ,1st of January.

It is not possible to add a separate backup battery to control the date / time stamp. Therefor we need a date time stamp from the PC every time the pc has connected to the Bluvision.

```
<?xml version="1.0" encoding="utf-8"?>
<SetTimeStamp Time ="11:12:18" Date="2015/09/11"/>
```

For Date notation: YYYY/MM/DD

For Time notation: HH:MM:SS

10 Pause and Resume Bluvision in running state including cover check

To Pause and Resume the BluVision during a run needs some special handling:

Senario:

- 1) System is running an analysis.
- 2) User presses pause button because of adding some new (priority) samples. / change rack
- 3) Discrete access will recalculate the scheduler according to what the user want to change.
- 4) System scheduler is still running to finish pending samples (if required) Filling actions
- 5) System scheduler is still running to finish pending samples (if required) measurement actions
- 6) User will be notified if Sample Reagent disk can be access safely

During running mode the user can click on a button on the screen to fill / change Sample / Reagent disk. If this is pressed Discrete access will receive this message

Discrete access can now show a popup with a question what does the user want to do :

- These actions need to be declared !!

11 Status check covers

The BluVision machine has several covers on it which can be opened. These covers will have a sensor to see if it is closed or not:

- 1) Sample / reagent disk cover (to keep the coolbox cool)
- 2) Cuvette disk cover (to keep the cuvettes on temperature and keep the environment light from the light unit)
- 3) Main dust cover (to secure the arm movement / needle protection)
- 4) Waste bin cover (to see if the waste bin is closed)
- 5) Filterwheel cover (to change filters in the filterwheel)

The Main dust cover has a magnetic lock on it to secure the cover while running the machine.

Command to retrieve cover status from DA to BV:

```
<?xml version='1.0'?>
<GetCoverStatus/>
```

Command to update cover status from BV to DA :

In this case the schedule **can** start / resume :

```
<?xml version='1.0'?>
<CoverStatus>
  <ReadyToRun>true</ReadyToRun>
  <SRCover>true</ SRCover>
  <CuvCover>true</CuvCover>
  <MainCover>true</MainCover>
  <WasteCover>true</WasteCover>
  <FilterCover>true</FilterCover>
</CoverStatus>
```

In this case the schedule **cannot** start / resume :

```
<?xml version='1.0'?>
<CoverStatus>
  <ReadyToRun>false</ReadyToRun>
  <SRCover>true</ SRCover>
  <CuvCover>true</CuvCover>
  <MainCover>false</MainCover>
  <WasteCover>true</WasteCover>
  <FilterCover>true</FilterCover>
</CoverStatus >
```

12 Update report window (append / remove errors to/ from list)

The BluVision machine has several types of internal errors / warnings. It is also possible to append / remove errors / warnings from the DA software into the BluVision. After an update of the DA software the report window on the TFT is shown.

Report system in Bluvision works with different kind of levels. In order of impact:

Error levels	Number	description
None	0	No error. Other error info is ignored at error level None
Status	1	No error. Status info only.
Warning	2	Machine can operate normal, but there is a possible problem
Busy	3	Machine is busy and will reach normal operation soon
Fail	4	Failure that any operator can fix
Maintenance	5	Machine needs maintenance and cannot operate any longer
Severe	6	Severe error that needs machine repair
Emergency	254	Error that needs manufacturer intervention

Command to retrieve error status from DA to BV:

```
<?xml version='1.0'?>
<GetErrors/>
```

Command to update error / warning status from BV to DA :

```
<?xml version='1.0'?>
<ErrorList>
  <Emergency Index="1">
    <Description>Emergency ! 703 Rebooted.</Description>
    <Explain>Please contact supplier!</Explain>
  </Emergency>
  <Emergency Index="2">>
    <Description>Emergency ! Motor not in position.</Description>
    <Explain>Please check: CuvetteDisk motor!</Explain>
  </Emergency>
  <Warning Index="3">>
    <Description>Temperature sensor </Description>
    <Explain>Problem with sensor: PT Heater In</Explain>
  </Warning>
</ErrorList>
```

If more than 5 items are located in the errorlist , a new message will be send to stay below the 1400 bytes of ethernet chunk size

Command to append an error / warning to error list (report screen) from DA to BV :

Note : If the ID already exists, the error message will be overwritten !

```
<?xml version='1.0'?>
<ErrorList>
  <Append ID="10" Level="2" Msg="Reagent A1 low level." MsgSub="Please fill cup"/>
</ErrorList>
```

Removal of one item :

```
<?xml version='1.0'?>
<ErrorList>
    <Remove ID="10">
</ErrorList>
```

Removal of all DA software items :

```
<?xml version='1.0'?>
<ErrorList>
    </RemoveAll>
</ErrorList>
```

13 Communication DLL Skalar (draft only, not released!!)

To connect the PC application to the Bluvision, you need the communication module of Skalar.

New version with number 1.0.0.1 (oldversion 1.0.0.0)

Size of read buffer inside communication module: 4096 bytes. (was 1024 previously).

This are the interfaces of the Ethernet client:

```
namespace SkalarEthernet
{
    public sealed class EthernetClient{
        public SocketError Connect(string ipAddress, int port);
        // (socket time out of max 80 seconds. Windows will standard use 75 seconds on socket timeout)
        public void Disconnect();
        public void Receive();
        public SocketError Send(string message);
    }
}
```

The interfaces of Ethernet server:

```
namespace SkalarEthernet
{
    public delegate void ReceivedEventHandler(object sender, ChangeArgs e);
    public delegate void ErrorEventHandler(object sender, ErrorArgs e);

    public class ChangeArgs : EventArgs
    {
        public string Message { get; set; }

        public ChangeArgs(string message)
        {
            Message = message;
        }
    }

    public class ErrorArgs : EventArgs
    {
        public SocketError ErrorCode { get; set; }

        public ErrorArgs(SocketError errorCode)
        {
            ErrorCode = errorCode;
        }
    }

    public class EthernetServer
    {
        private volatile bool bRunning = false;
        private TcpListener ServerSocket;
        private TcpClient ClientSocket;

        public event ReceivedEventHandler Received;
        public event ErrorEventHandler Error;

        public void Start(int port)
        {
            String strHostName = string.Empty;
            strHostName = Dns.GetHostName();
            string IP = "";
            IPAddress[] addresses = Dns.GetHostAddresses(strHostName);
            IPAddress address;
            foreach (var ipaddress in addresses)
            {
                if (System.Net.IPAddress.TryParse(ipaddress.ToString(), out address))
                {
                    if (ipaddress.AddressFamily == System.Net.Sockets.AddressFamily.InterNetwork)
                    {
                        ServerSocket = new TcpListener(address, port);
                        IP = address.ToString();
                    }
                }
            }
        }

        ClientSocket = default(TcpClient);
    }
}
```

```
ServerSocket.Start();
Thread workerThread = new Thread(this.Run);
workerThread.Start();

ServerSettings.Instance.IpAddress = IP;
ServerSettings.Instance.Port = port;
}

public void stop()
{
    if (ServerSocket != null)
        ServerSocket.Stop();

    bRunning = false;
}

public void OnReceive(ChangeArgs e)
{
    if (Received != null)
        Received(this, e);
}

public void OnError(ErrorArgs e)
{
    if (Error != null)
    {
        try
        {
            Error(this, e);
        }
        catch { /* Ignore, most likely the application is in shutdown */ }
    }
}

private void Run()
{
    byte[] readBuffer = new byte[4096];
    StringBuilder message = new StringBuilder();
    bRunning = true;
    int numberofBytesRead = 0;

    try
    {
        ClientSocket = ServerSocket.AcceptTcpClient();

        while (bRunning)
        {
            NetworkStream networkStream = ClientSocket.GetStream();

            numberofBytesRead = networkStream.Read(readBuffer, 0, readBuffer.Length);
            message.AppendFormat("{0}", Encoding.ASCII.GetString(readBuffer, 0, numberofBytesRead));

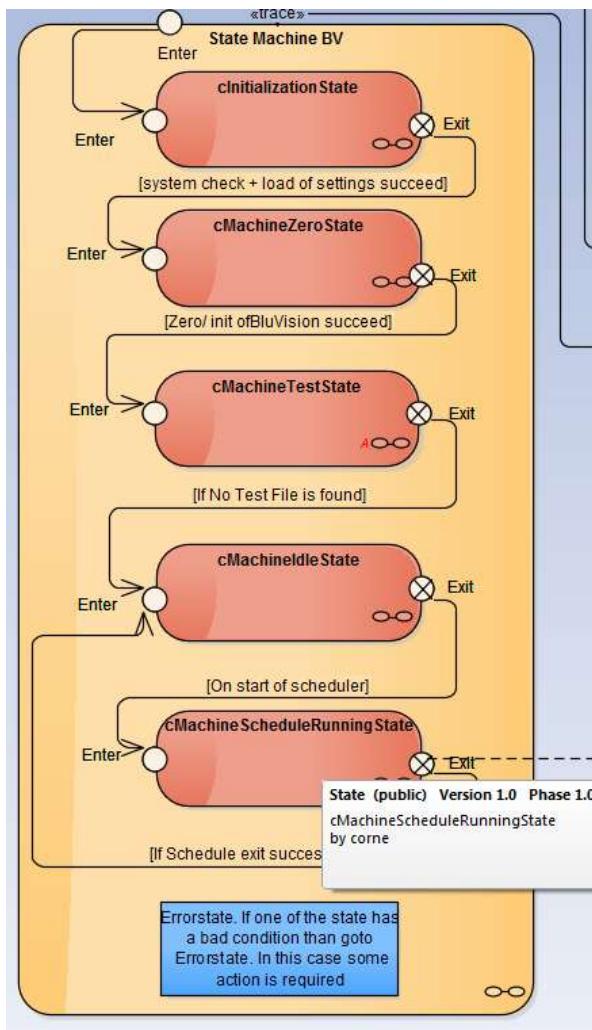
            var eventArgs = new ChangeArgs(message.ToString());
            OnReceive(eventArgs);
            message.Clear();
            numberofBytesRead = 0;
        }
    }
    catch (SocketException se)
    {
        var eventArgs = new ErrorArgs(se.SocketErrorCode);
        OnError(eventArgs);
        bRunning = false;
    }
    catch (Exception e)
    {
        var eventArgs = new ErrorArgs(SocketError.SocketError);
        OnError(eventArgs);
        bRunning = false;
    }
}
```

```
    }  
}  
}
```

13.1 Device discovery over Ethernet with SSDP protocol (uPNP) Draft Only not released!!!

14 BluVision states

The Bluvision can be in several states:



To retrieve the state of the BV this command must be used:

```
<?xml version='1.0' encoding="utf-8"?>
<GetState/>
```

This is the response of the BV:

```
<?xml version='1.0' encoding="utf-8"?>
<SystemState>Idle</SystemState>
```

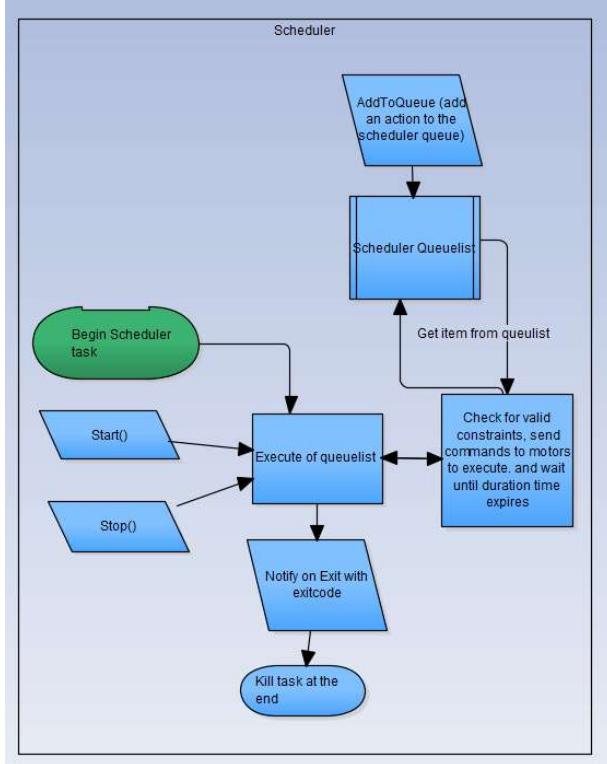
This are the available states:

Initialization, Idle, Running, Standby, ChangeFilter, Bootloading, Zero, Test, Paused, Error, Unknown

The DA software is only able to start a new sequence, or continue a sequence in the state Idle, Running, Standby, Paused. In all other states it isn't possible to do an action execution.

16.0 Firmware implementation: Parallel measurement, load and unload while sample pickup.

Actual situation until version 1.4 of BV firmware:

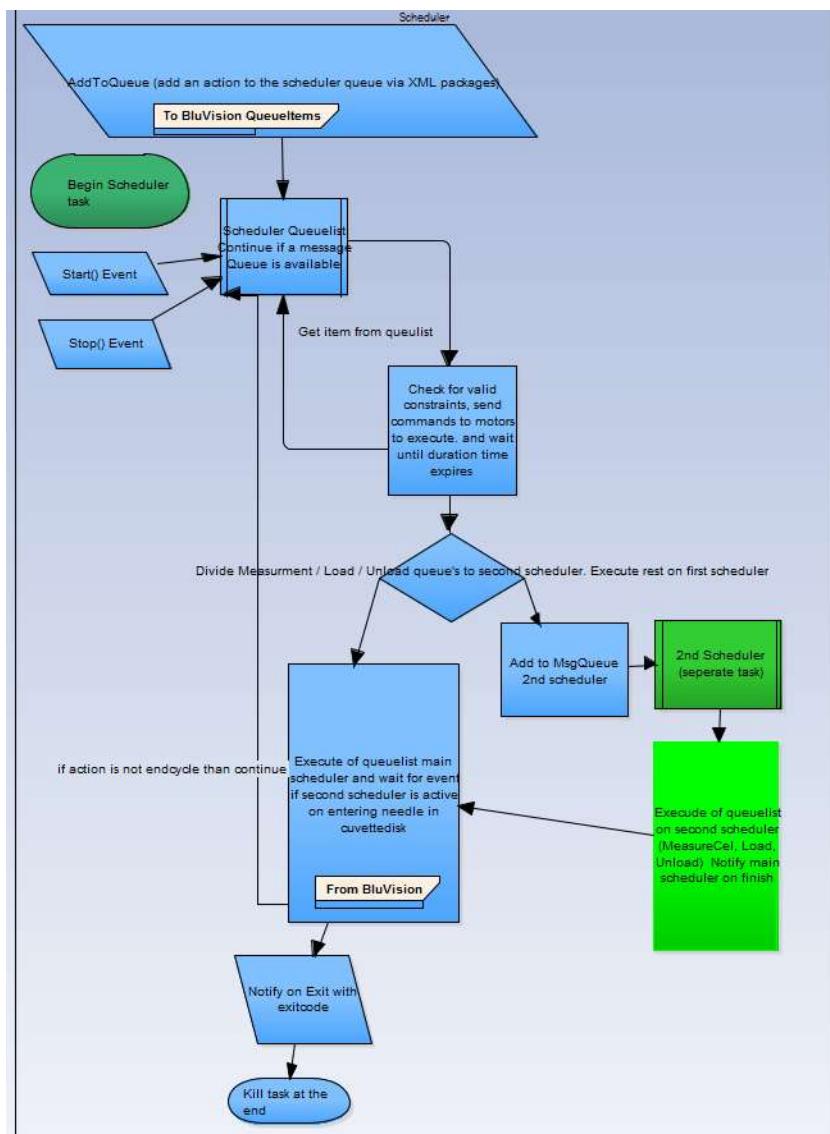


16.1 New functionality: parallel movement / measurement during run.

According to the specs, the system needs to have parallel actions during rinsing, take sample **and** sample to sample actions (all actions which are not directly related to the cuvette disk). During the arm movements, actions like: measure cel, load cuvette and unload cuvette should be possible. There must me a lock mechanism that prevents parallel actions which could lead to a broken needle.

All queue message items are received by the main scheduler and will be executed by the Queue list.

(New situation, since FW version 1.5)



16.2 Scheduler for parallel action's like measurement, loading and unload of cuvettes Firmware side.

The start/ basis is the actual main sequential scheduler. The only extension is, that there will be a 2^e scheduler in a separate task. The main scheduler gets an additional event with it. This will ensure that the MoveToCuvette is blocked at the moment the needle wants to go in the cuvette and the second scheduler is not ready. The 2nd scheduler is powered by a message Queue with a copy of the scheduler queue item actions from the first scheduler.

For the 2nd scheduler task execution to proceed, it will first check whether the needle is in the cuvette. This is the only condition blockade which is in the 2nd task scheduler.

There are 3 different sequence actions possible in the 2nd scheduler: measure cell, unload and loading cuvettes.

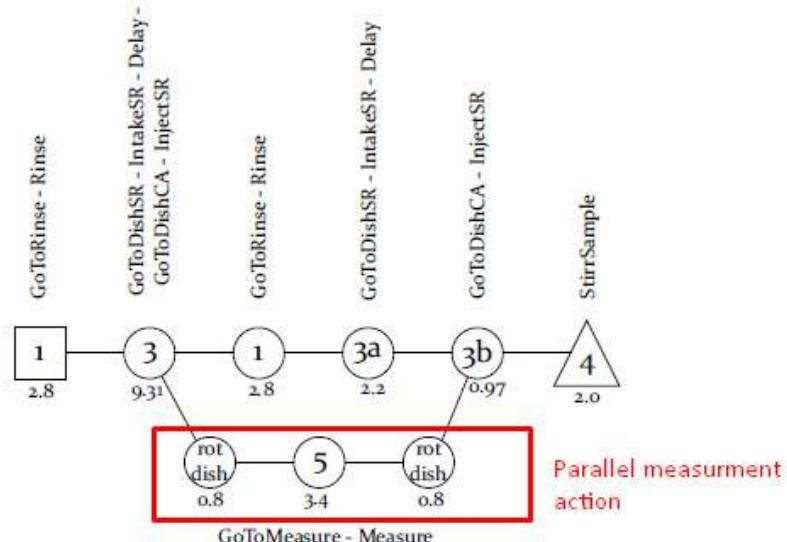
As long as there are actions in the 2nd scheduler that have yet to be executed/finished, the first scheduler process will be blocked to move the cuvettes. The remaining sequence of the first scheduler can continue.

By keeping up with this principle, it is possible to measure simultaneously with sample / reagent picked up. Mix and wait for 2nd scheduler to finish.

16.3 Scheduler actions From Discrete Access (Software side).

In addition to the regular / previous queue, the **Measure**, **Load** and **Unload** will act **differently**. These will be performed immediately after the wait parameter is passed and the cuvette disc is available. If the Queue is correct, the parallel actions will be executed **in** during the time that the main scheduler is moving. The parallel actions will be executed until the point where the main scheduler wants to dispense sample/reagent into the cuvette cell. As for the Measure, Load, and Unload commands parameters are extended, please refer to the description in paragraph 2.

Figure 136: Method protocol steps for Nitrite 1 - 100 ugL.



20 Obsolete commands (not in use anymore since communication version 2.0):

20.0 AddToQueue Commands

20.1 Commands type="WashToSample"

As discussed I send you the new commands to improve speed of movement / reduction of QueueItems / Use of abstract positions defined in BluVision machine:

These commands will be added to the communication documentation of the BluVision!

Is it possible to have these commands in the software by Tuesday after lunch? (30 th of June).
Thanks in advance!

From Wash (No Rinse) To Sample / Reagent movement: (WashToSample)

- Fast up to Top level Wash
- Stirr very short (eliminate drops) (variable input)
- Fast up to Top ZArm
- [parallel movement]
- Turn Sample Reagent disk to (variable in Abstract position)
- Diluter pump Air Volume (input variable in ul)
- Rotate Arm(variable in Abstract position)
- [end parallel movement]
- Down Z Arm into cup (stop on level detection)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="WashToSample">
    <UniqueId>1</UniqueId>
    <Pos>A1</Pos>
    <NextPos>Cuv2Cel1M</NextPos>
    <Stirr>100</Stirr>
    <Volume>100</Volume>
    <DilSpd>10</DilSpd>
    <DurationTime>2000</DurationTime>
    <DemiVol>200</DemiVol>
    <DemiSpd>10</DemiSpd>
  </Commands>
</AddToQueue>
```

Pos = Sample Reagent Position with rack positon A1.

Stirr = Stirr time in mSec for drop removal

Volume = Air in Diluter in ul volume

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

DemiVol = is amount of demiwatert pickup bythe small diluter to create a dilution in a cuvettecel. **Not allowed to use in SAMPLE CUP!!!**

DemiSpd = is speed of large diluter during dispense of sample. The small diluter will use the DilSpd which already exists

If **DemiVol >= 0** ; During take of Demiwater in the needle, the needle will stay in the rinse position. Also the waste water pump is activated in this situation.

20.2 Commands type="TakeSample"

- TakeSample:** (TakeSample)
 - Take sample with Diluter (var volume)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="TakeSample">
    <UniqueId>1</UniqueId>
    <Volume>500</Volume>
    <DilSpd>10</DilSpd>
    <DurationTime>2000</DurationTime>
  </Commands>
</AddToQueue>
```

Volume = Sample volume in Diluter in ul

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

20.3 Commands type="SampleToCuv">

- 3) **Sample Rack to Cuvette:** (SampleToCuv)
 - Fast up to Top level Cup (sample / reagent)
 - Stir very short (eliminate drops) (variable input)
 - Fast up to Top ZArm
 - [parallel movement]
 - Turn Cuvette disk to (variable in Abstract position)
 - Diluter pump Air Volume (input variable in ul)
 - Rotate Arm(variable in Abstract position)
 - [end parallel movement]
 - Down Z Arm into cuvette (stop on lowest cuvette position)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="SampleToCuv">
    <UniqueId>1</UniqueId>
    <Pos>Cuv1Cel3D</Pos>
    <NextPos>B1</NextPos>
    <Stirr>100</Stirr>
    <Volume>100</Volume>
    <DilSpd>10</DilSpd>
    <DurationTime>2000</DurationTime>
  </Commands>
</AddToQueue>
```

Volume = Air in Diluter in ul volume

Pos = Cuvette Position with cuvette rack positon 1 cel 3.

Stirr = Stirr time in mSec for drop removal

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

20.4 Commands type="DispenseSample"

4) Sample Dispense:

- Dispense sample with Diluter (var volume)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
<Commands type="DispenseSample">
  <UniqueId>1</UniqueId>
  <Volume>450</Volume>
  <DilSpd>10</DilSpd>
  <Stirr>100</Stirr>
  <DurationTime>2000</DurationTime>
  <DemiVol>5000</DemiVol>
  <DemiSpd>10</DemiSpd>
</Commands>
</AddToQueue>
```

Volume = Sample volume in Diluter in ul ! Attention, you have to fill in the number off ul including the Air that is inside your system

Stirr = Stirr time in mSec to stir liquids if stir time is 0 , no stirring is done

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

DemiVol = is amount of demiwatert dispensed from the large diluter to create a dilution in a sample cup. **Not allowed to use in CUVETTE CEL!!!**

DemiSpd = is speed of large diluter during dispense of sample.

If DemiVol >= 0 During dispense of Demiwatert in the sample cup, the needle will move up in cup. For safety a calculation will be done by the BV to prevent overflood of demiwatert in sample cup.

-

20.5 Commands type="MoveToWash"

- 5) Move to Wash (No Rinse) from Cuvettecel: (WashToSample)
 - Start pump Rinsing system (to be ready on flush / rinse)
 - Fast up to Top level Cuvette
 - Stirr very short (eliminate drops) (variable input)
 - Fast up to Top ZArm

[parallel movement]

 - Rotate Arm(variable in Abstract position)

- Rotate Cuvettedisk to next position. (Z Arm Will not wait for completeness, wait only for Rotate arm on position)

[end parallel movement]

 - Down Z Arm into wash position

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="MoveToWash">
    <UniqueId>1</UniqueId>
    <Stirr>100</Stirr>
    <DurationTime>2000</DurationTime >
    <NextPos>Cuv2Cel1M</NextPos>
  </Commands>
</AddToQueue>
```

Stirr = Stirr time in mSec for drop removal

20.6 Commands type="Rinse"

Rinse:

- Move diluter to 0 position
- Open valves to rinse inner and outer needle
- Rinse for x time
- SlowMove of needle from low to high point to get a dry needle
- Close valves rinsing system
- Stop Rinse pump
- Diltuter to 0 position

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="Rinse">
    <UniqueId>1</UniqueId>
    <Time>500</Time>
    <DilSpd>10</DilSpd>
    <DurationTime>2000</DurationTime >
    <Volume>100</Volume>
  </Commands>
</AddToQueue>
```

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

New in 1.13: Volume: Take # amount of uL volume in small diluter to use in a predilution action in cuvette cel DilSpd is still in use as dilutior speed.

20.7 Commands type="CuvToSample"

- 7) Move to Cuvette to sample no Wash: (CuvToSample)
 - Fast up to Top level Wash
 - Stirr very short (eliminate drops) (variable input)
 - Fast up to Top ZArm
 - [parallel movement]
 - Turn Sample Reagent disk to (variable in Abstract position)
 - Diluter pump Air Volume (input variable in ul)
 - Rotate Arm(variable in Abstract position)
 - [end parallel movement]
 - Down Z Arm into cup (stop on level detection)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="CuvToSample">
    <UniqueId>1</UniqueId>
    <Pos>A1</Pos>
    <NextPos>Cuv01Cel02M</NextPos>
    <Stirr>100</Stirr>
    <Volume>100</Volume>
    <DilSpd>10</DilSpd>
    <DurationTime>2000</DurationTime>
  </Commands>
</AddToQueue>
```

Stirr = Stirr time in mSec for drop removal

Pos = Cuvette Position with cuvette rack positon 1 cel 3.

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

So I want to change the Cuv3Cel1 name into :

- Cuv03Cel01D for dispense position Cuvette 3 Cel 1
- Cuv03Cel01M for measure position Cuvette 3 Cel 1
- Cuv03L for loading position Cuvette 3
- Cuv03U for Unloading position Cuvette 3

For each number, I want to use 2 digits!

How to use the diluter pump:

Diluter pump volume: <Volume>100</Volume>

- In case of pickup air this means that the diluter pickup 100 ul of air

next step is pickup sample: <Volume>100</Volume>

- In this case the diluter will pickup 100 ul of liquid. The position of the diluter is now on position of 200 ul (100ul air + 100 ul sample)

Next step is dispense sample: <Volume>80</Volume>

- In this case the diluter will dispense 80 ul of liquid. The position of the diluter is now on position of 120 ul (100ul air + 100 ul sample – 80 ul dispense)

ADC gain: probably for each method / color filter we need a dedicated ADC Gain setting to adjust the measurement values. We need this parameter to prevent that the result will be out of bounds. The parameter value of this setting can be : 0,1,2,3,4,5,6,7

2.8 Commands type="MeasureCel"

Measure cuvette cel: (MeasureCel)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="MeasureCel">
    <UniqueId>1</UniqueId>
    <Pos>Cuv02Cel03M</Pos>
    <NextPos>Cuv03Cel01D</NextPos>
    <ADCGain>0</ADCGain>
    <Filter>Fil1</Filter>
    <NextFilter>Fil2</NextFilter>
    <DurationTime>2000</DurationTime>
    <WaitTime>2000</WaitTime>
  </Commands>
</AddToQueue>
```

Gain settings:

0 = 000 = 1x
1 = 001 = 2x
2 = 010 = 4x
3 = 011 = 8x
4 = 100 = 16x
5 = 101 = 32x
6 = 110 = 64x
7 = 111 = 64x

WaitTime: this parameter will set the amount of mSec to wait before doing the real measurement. This is needed in case of parallel movement (the arm can continue). It is possible that sample is not ready for the incubation time yet, in this case the WaitTime will wait for this amount of mSec. In the meantime the position of the filterwheel and the cuvettedisk are already set to the right position (if they weren't already)...

20.9 Commands type="LoadCuvette"

Load / Unload Cuvettes:

Load cuvette array into cuvette disk, this will place a cuvette into the desired location.

Attention: The BluVision will keep an internal registration of usage of cuvettes. You should first unload the cuvette of that position. If a cuvette is in this position according the internal registration, NO new cuvette will be inserted into the system!!! If loading of new cuvette fails for what reason, the bluvision will send the PC an error message.

Load cuvette in a queue:

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="LoadCuvette">
```

```
<UniqueId>1</UniqueId>
<Pos>Cuv01L</Pos>
<NextPos>Cuv01Cel01M</NextPos>
<DurationTime>2000</DurationTime>
</Commands>
</AddToQueue>
```

20.10 Commands type="UnloadCuvette"

Unload cuvette in a queue:

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
<Commands type="UnloadCuvette">
<UniqueId>1</UniqueId>
<Pos >Cuv01U</Pos>
<NextPos >Cuv01Cel01M</NextPos>
<WasteBin>1</WasteBin>
<DurationTime>2000</DurationTime>
</Commands>
</AddToQueue>
```

WasteBin 1 = front waste bin

WasteBin 2 = rear waste bin

Each waste bin can contain 64 cuvettes. (128 in total)

See chapter 3.2 for getting status of wastebin

20.11 Commands type="CheckCuvBlock"

Command to check an entire cuvette Block: starting from Cuv01Cel01

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
<Commands type="CheckCuvBlock">
<UniqueId>1</UniqueId>
<Pos >Cuv01Cel01M</Pos>
<NextPos >Cuv02Cel01M</NextPos>
<ADCGain>0</ADCGain>
<Filter>Fil1</Filter>
<NextFilter>Fil2</NextFilter>
<DurationTime>2000</DurationTime>
</Commands>
</AddToQueue>
```

This will return the same response as you measure a CuvetteCel. The Bluvision is not capable to see if a CuvetteCel is good or bad. The software should add some constraints over the test results!

20.12 SampleToSample (for dilutions)

- 12) **Sample Rack to Sample Rack: (SampleToSample)**
- Fast up to Top level Cup (sample / reagent)
 - Stir very short (eliminate drops) (variable input)
 - Fast up to Top ZArm
 - [parallel movement]
 - Turn sample disk to (variable in Abstract position)
 - Diluter pump Air Volume (input variable in ul)
 - Rotate Arm(variable in Abstract position)
 - [end parallel movement]
 - Down Z Arm into sample disk (stop on interrupt in sample position)

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
  <Commands type="SampleToSample">
    <UniqueId>1</UniqueId>
    <Pos>A1</Pos>
    <Stirr>100</Stirr>
    <Volume>100</Volume>
    <DilSpd>10</DilSpd>
    <DemiVol>5000 </DemiVol>
    <DemiSpd>10</DemiSpd>
    <DurationTime>2000</DurationTime >
  </Commands>
</AddToQueue>
```

Volume = Air in Diluter in ul volume

Pos = Sample rack position.

Stirr = Stirr time in mSec for drop removal

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

DemiVol = is amount of demiwasser dispensed from the large diluter to create a dilution in a sample cup. Not allowed to use in CUVETTE CEL!!!

DemiSpd = is speed of large diluter during dispense of sample.

If DemiVol > = 0 : During dispense of Demiwasser in the sample cup, the needle will move up in cup.
For safety a calculation will be done by the BV to prevent overflow of demiwasser in sample cup.

20.13 SampleMixing (for dilutions)

Before sample mixing the Diluter must be empty. To do this first use command SampleToSample and then command DispenseSample. The **Volume** parameter of SampleMixing is the volume that is used during sample pickup and dispense

After this Sample Mixing sequence, you should do a **MoveToWash** command! Because the needle is polluted !!

13) Sample Rack Mixing in Cup: (SampleMixing)

- [begin repeat x times]

- down to bottom of sample / reagent.
- Diluter sample pickup with defined volume
- Fast up to Top level Cup (sample / reagent)
- Zarm down on interrupt stop
- dispense diluter with defined volume
- [END repeat x times]

```
<?xml version="1.0" encoding="utf-8"?>
<AddToQueue>
<Commands type=" SampleMixing ">
<UniqueId>1</UniqueId>
<Pos>A1</ Pos >
<Volume>100</Volume>
<DilSpd>10</DilSpd>
<DurationTime>2000</DurationTime>
<Number>10</Number>
</Commands>
</AddToQueue>
```

Volume = during sample pickup and dispense

Pos = Sample rack position.

Stirr = Stirr time in mSec for drop removal

DilSpd = diluter speed constraints between 1 – 10 (1 = slow – 10 = fast);

Number = x times to repeat the mixing sequence

Obsolete part 15.0 Duration times for generating faster schedule queues

To calculate an accurate schedule, the BluVision provide the system statistics for each movement

15.1 Get duration information from BV

To get these parameters from the BV, DA need to send this comment:

```
<?xml version='1.0' encoding="utf-8"?>
<GetDurationTable/>
```

15.2 Response duration information from BV to DA

This is the response of the BV:

For each action an ID is given as name of the action ="TakeSample"

DurTime="350" is base duration time without any liquid handling.

DilFact = "10" add msec for each speed step, so for diluter

DilVol="7" add mSec for each 10 uL

Look up table diluter speed

Diluterspeed	Multiply DilFact	Sum	Add duration time mSec
10	0	0x10mSec	0
9	1	1x10mSec	10
8	2	2x10mSec	20
7	3	3x10mSec	30

6	4	4x10mSec	40
5	5	5x10mSec	50
4	6	6x10mSec	60
3	7	7x10mSec	70
2	8	8x10mSec	80
1	9	9x10mSec	90

Look up table diluter volume

Diluter volume	Multiply DilVol for each 10 uL volume	Sum	Add duration time mSec
10	7	1x7mSec	7
20	7	2x7mSec	14
30	7	3x7mSec	21
40	7	4x7mSec	28
50	7	5x7mSec	35
60	7	6x7mSec	42
500	7	50x7mSec	350

Example of calculation TakeSample with dilution speed 7 Volume pickup 300 uL
<Action ID="TakeSample" DurTime="520" DilFact="38" DilVol="7"/>

Total duration time: 520mSec + (10 - 7) *38 + ((300/10) * 7) = 520 + 114 + 210 = 844 mSec
See lookup dilspeed See lookup table volume

```
<?xml version='1.0' encoding="utf-8"?>
<DurationTable>
<Action ID="TakeSample" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="SampleToCuv" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="DispenseSample" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="MoveToWash" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="WashToSample" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="Rinse" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="CuvToSample" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="LoadCuvette" DurTime="520"/>
<Action ID="UnloadCuvette" DurTime="520"/>
<Action ID="MeasureCel" DurTime="220"/>
<Action ID="SampleToSample" DurTime="520" DilFact="38" DilVol="7"/>
<Action ID="SampleMixing" DurTime="520" DilFact="38" DilVol="7"/>
</DurationTable>
```

21 Obsolete part: Recalculation of Schedule queue

If it is necessary to change the actual send queue, the DA software should sent this command to notify it to the BluVision:

```
<?xml version="1.0" encoding="utf-8"?>
<DeleteQueue DeleteAfterID="500"/>
```

Reply from BluVision to DA software after receiving command:

```
<?xml version="1.0" encoding="utf-8"?>
<DeleteQueue status="received"/>
```

Reply from BluVision to DA software after correct deleted command:

```
<?xml version="1.0" encoding="utf-8"?>
<DeleteQueue status="executed"/>
```

Reply from BluVision to DA software after incorrect execution of delete command:

```
<?xml version="1.0" encoding="utf-8"?>
<DeleteQueue status="failed"/>
```

After sending this command, DA software can start sending the new schedule queue to the BluVision.

For example:

- The BluVision received queue items 1 – 800.
- And it is executing a schedule queue at ID 450
- User want to change the order of execution during this run.
- DA re calculates this request. It seems that the Queue id's after 500 (501 till 800) are invalid.
- DA send the Delete Queue message to the BluVision. <DeleteQueue DeleteAfterID="500"/>
- BluVision replies this request to the DA software. (<DeleteQueue status="received"/>)

- After this, the DA software sends a new queue list with UniqueID 501 Till 900 (or something else).
- BluVision will execute the previous unique ids including 500. UniqueID 501 until 800 will be ignored. After second UniqueID 501 is found this is the start of the new QUEUE, BV will execute this one. At this moment the BV will also send a status update of the Delete Queue action: .(<DeleteQueue status="executed"/>)
- The new updated command list will be executed.

Revision History

Version 1.0

Created: May 8, 2015
Author: Corné Verkooijen
Change: Initial release

Version 1.1

Created: 2015
Author: Corné Verkooijen
Change: Add commands for filter wheel (chapter 5) and temperature control (chapter 6), recalculation schedule queue (chaper 7), Notification to LCD display (chapter 8)

Version 1.2

Created: 2015
Author: Corné Verkooijen
Change: change send attribute settings “ into ‘ character in XML structure.

Version 1.3

Created: August 5, 2015
Author: Corné Verkooijen
Change: “true” and “false” XML attributes states into “1” and “0”
Added 2.13 SampleMixing (for dilutions)

Version 1.4

Created: August 5, 2015
Author: Corné Verkooijen
Change: add 9 Set system time BluVision mainboard
Changed chapter 8 button pressed response on request of Agaram

Version 1.5

Created: October 19, 2015
Author: Corné Verkooijen
Change: Changed chapter 7, removed ExecuteUntilID and ResumelD. Add one new command : <DeleteQueue DeleteAfterID="500"/> this was done to have a more clear command to communicate between DA and BV
Add chapter 10 Pause and Resume Bluvision in running state including cover check
Add chapter 11 Status check covers
Change: chapter 8: add timeout value to popup window

Version 1.6

Created: November 4, 2015
Author: Corné Verkooijen
Change: Add chapter 12 Update report window (append / remove errors to/ from list)

Version 1.7

Created: December 21, 2015
Author: Corné Verkooijen
Change: Add one command to get all cuvette status information at once: 3.1.1 Get Status of All Cuvette arrays from PC to BV

Version 1.8

Created: December 23, 2015
Author: Corné Verkooijen

Change: Changed reply from BV for one command to get all cuvette status information at once: 3.1.1 Get Status of All Cuvette arrays from PC to BV

Version 1.9

Created: December 27, 2015

Author: Corné Verkooijen

Change: added to paragraph 12 extention "Index" and limit the maximum errorlist size to 5 items per Ethernet frame.

Version 1.10 not released

Created: Januari 28, 2016

Author: Corné Verkooijen

Change: added to paragraph 13 Communication DLL Skalar

Version 1.11 not released

Created: February 23, 2016

Author: Corné Verkooijen

Change: added paragraph 13.1 Communication DLL Skalar with automatic search Skalar devices by SSDP protocol (UPNP)

Added paragraph 14 Bluvision states

Version 1.12

Created: Januari 18, 2017

Author: Corné Verkooijen

Change: Added paragraph 14 Bluvision machine states,
Added paragraph 15 new command actions for parallel movement in scheduler.

Added parameter to Unload Cuvette to select eject location.

Added paragraph 3.2 Get Waste bin status

Added parameter to MeasureCel action paragraph 2.8

Version 1.13

Created: februari 28, 2017

Author: Corné Verkooijen

Change:

- update paragraph 15 washtosample was missing.
- Add in Rinse command -> Volume to take for dilution in cuvettedisk.
- Update paragraph 2.4 dispense sample with <DemiVol>5000 </DemiVol>
<DemiSpd>10</DemiSpd>for dilutions.

Version 1.14

Created: march 16, 2017

Author: Corné Verkooijen

Change:

-

- Remove in Rinse command -> Volume to take for dilution in cuvettedisk.

Add take sample for diluent in WashToSample position.

Version 2.0

Created: May 5, 2017
Author: Corné Verkooijen
Change:

- Remove BeginGroupCommand (not used anymore) (obsolete)
- Remove EndGroupCommand (obsolete)
- move the dilution volume position for dilution in sample cup from DispenseSample to 2.12 SampleToSample (for dilutions)
- Paragraph 2. Add new combined scheduler action to make the scheduler more accurate
- Paragraph 4 leveldetection changed (update with new commands)
- Obsolete part 15.0 Duration times for generating faster schedule queues
- 21 Obsolete part: Recalculation of Schedule queue

Version 2.1

Created: June 12, 2017
Author: Corné Verkooijen
Change:

Update to paragraph 2.3.5 Response from BluVision to Discrete Access:
Added response command for actionstep.

Version 2.2

Created: August 7, 2017
Author: Corné Verkooijen
Change:

Update to paragraph 2.3.* Response from BluVision to Discrete Access:
Added response attribute Delay to the Execution command.

Version 2.3

Created: August 29, 2017
Author: Corné Verkooijen
Change:

Update to paragraph 2.3.1 add functionality to do a pre rinse before take sample in main scheduler.

PreRinIn="1200" First rinse needle before take sample with DemiWater rinse volume inside needle in µL volume demiwater

PreRinOut="3000" First rinse needle before take sample with DemiWater rinse volume outside needle in µL volume demiwater