ALS-G Auto Sampler Picarro

ALSG_API.dll and Picarro_Training.exe release notes

For Product:	ALS-G	**For Hardware Version:**	1.00 and 1.10
Document Number:	aibv15037	**Document Revision:**	1
Author:	Piet de Jager	**Date:**	04 Feb 2015
Release Version:	1.40	**Previous Release:**	1.30

#	**Bug Fixes 1.40**
1	Teach Wash Station moves now to the last calibrated x and y position of the selected wash station (previously this x:0 and y:0)
2	Teach Tray moves now to the last calibrated x and y position of the selected tray (previously this x:0 and y:0)
3	In Teach mode the movement status is now continuously displayed in the move status dialog for every move.
4	In Teach mode the Tray-z value is stored correctly and is a result of the distance between a Tray Vial detection and RunReferenceRun on the z-axis.
5	In Tray Teach mode a vial detection is performed in order to measure the z value of the selected tray.
6	In Teach mode the x and y coordinates are validated, should be within x-axis-min and x-axis-max (as determined by the auto x-width procedure).
7	On Teach screen the 'Z' button is replaced by Z (=absolute) or dZ (relative) and on clicking this button, the z- position is (re)calculated
8	On Teach screen, the 'Run xyz Reference' button text is replaced by 'Home' .
9	In Teach mode, cancelling a calibration procedure does not result in storing parameters in parameter.td file with wrong (aborted) x, y and z positions.
10	Show error message when parameter.td file version is not compatible with ALSG_API.dll version, and show that file contents will be updated to ALSG_API.dll version.
11	Set slower speed on x, y and z axis in teaching mode! This works more convenient in continuous teaching mode and on particular the z-axis there is less mechanical noise.
12	When x-axis home position is on the right side, this works now correctly (Bugzilla #17)
13	Tray 'II' z coordinate is now correct after calibration the position. Previously it was overwriting tray 'I' (Bugzilla #169)
14	Injection x-axis position cannot be calibrated outside the x-axis range (Bugzilla #166)

15	If x-Left, x-right or y-sensor active, do a reference run on x and y as well before release motor power to allow the user to move manually in manual calibration mode.
16	On selecting cancel in teaching mode do a x, y and z reference run and don't display messages that positions are determined, this is cancel!
17	Syringe suddenly fell down in home position when closing the application (Bugzilla #43). This happened when the z-axis was not in home position (hold by the magnet) and the motor current was turned off when exiting the application. Now the z-axis will first be homed and then the current turned off at application exit
18	Increased maximum limit to 300 for method parameter 'Insertion Speed'.
19	A direct move from injector x to Injector y.
20	Running a reference run on only the y-axis followed by a move to vial position x. Previously the syringe dropped down on its current location without moving to the vial position.
21	Installable and runnable on Vista, Win7, Win8, etc by separating the td files from the application when the user selects Windows Program Files as folder to install. From Vista and later user programs are not allowed to write in the Program Files folder.
22	Keep rest steps on u-axis when converting ul to mm to motor steps and handle these in the next u-axis move.
23	Liquid drawn from Waste during a "cold GoToWait" API call for ALSg hardware rev 1.00 and 1.10. Previously this happened and it shouldn't.
24	The syringe will always be emptied in the waste during a "cold GoToWait" call for ALSg rev 1.00 and 1.10
25	If the waste isn't calibrated, MicroStep commands could still be used. This is now blocked.
26	ALS-G firmware revisions b and c are now also handled as a REV 1.00 device. Previously only rev. d.
27	The syringe could sometimes not be released from the magnet. Created a 'z-axis departure' mode (choices: Normal and Intensified). When selecting 'Intensified', make sure to also measure the magnet space by clicking the "Auto" button, the syringe goes first up when in upper position before going down. Select this mode only if you are sure that the syringe can't be released from the magnet.

#	**New Features 1.40**
1	Supports a second wash station.
2	Supports four trays (previously 2).
3	Display Firmware version in Picarro Training application
4	Display both Picarro-Training application version and ALS-G_API.dll version in application
5	Increased X_AXIS_MAX from 470 mm to 1000 mm (=1 meter), the Auto-Width button contains a procedure to determine the length of the x-axis automatically. To allow the client application automatically measure the x-axis length, a new API call is implemented.
6	Beep two times when Z-axis moves less steps than expected. (= error beep)
7	Show warning message at application start when sampler positions aren't calibrated.
8	Supports two new tray types: "Well Plate 96" and "Well Plate 384".
9	Implemented an API to automatic measure space of z-axis magnet.
10	Implemented method parameter 'Retraction Speed' which is the z-axis speed when leaving the injector. Previously the syringe leaved the injector by a reference run on z-axis.
11	Redesign of Training and Configuration screen layout.
12	Implemented configuration parameter 'Moment_ReadyOut Start_Out'' , choices: 'After_Plunger_Down', 'At_Start_Of_Plunger_Down'. The start out signal at injection timing is determined by this parameter.
13	Faster injection. Previously the syringe up move leaving the injector, which is part of the injection procedure, was delayed with 0.5 seconds in order to generate a start out pulse of 0.5 s width. The pulse is now generated in parallel and doesn't extend the needle time in the injector.

Additional parameters in Parameter.td file in version 1.40

```
[Configuration]
WS2_TYP=0                    // 2nd WashStation type


[Installation]

TRAY_x[2]=-1.000             // Tray 3
TRAY_y[2]=-1.000
TRAY_z[2]=-1.000
TRAY_zOffset[2]=5.000

TRAY_x[3]=-1.000             // Tray 4
TRAY_y[3]=-1.000
TRAY_z[3]=-1.000
TRAY_zOffset[3]=5.000

SAMPLE_z[2]=-1.000           // Tray 3-z-Axis
SAMPLE_z[3]=-1.000           // Tray 4-z-Axis

WASH_STAT2_x=-1.000          // 2nd WashStation
WASH_STAT2_y=-1.000
SOLVENT2_z=-1.000
WASTE2_z=-1.000
ISTD2_z=-1.000
```

API changes in ALSG_API.dll

```
enum AlsgParam : int
{
   Parm_AutoWidth = 0,
   Parm_StoredMagnetSpace = 1,
   Parm_MeasuredMagnetSpace = 2,
   Parm_SpecialMotionZ = 3,
   Parm_RefRunZErrorCounts = 4,
}
```

```
enum AlsgParam : int
{
   Parm_AutoWidth = 0,
   Parm_StoredMagnetSpace = 1,
   Parm_MeasuredMagnetSpace = 2,
   Parm_SpecialMotionZ = 3,
   Parm_RefRunZErrorCounts = 4,
}
```

```
enum Axis : short
{
   x = 0,
   y,
   z,
   u
}
```

```
enum SysCtrlCmd : short
{
   MoveAxisPos = 0,
   MovePlunger = 1,
   SetMotorCurrent = 2,
   GetAxisPos = 3,
   GetHardwareVersion = 106,
   GetHardwareSubVersion = 107,
   ReadSensor = 108,
}
```

```
/*
Use the following two API calls to determine the width of the X-axis automatically
*/
```

```
// Auto determine Width x-axis only (parameter aAxis: X=0, see enum Axis)
// The determined width is stored in the parameter.td file as item "X_AXIS_MAX="
extern "C" __declspec (dllexport) BOOL _stdcall alsgDetermineAutoWidth(INT16 aAxis)

// Read auto determined width x-axis (parameter aParam: AutoWidth=0, see enum AlsgParam)
extern "C" __declspec (dllexport) double _stdcall alsgGetParam(int aParam)

/*
Use the following API calls to determine the space above the magnet of Z-axis
*/
// Determine space above magnet of Z-axis. (parameter aReserved: false)
extern "C" __declspec (dllexport) BOOL _stdcall alsgDetermineMagnetSpace(bool aReserved)

// Read determined space above magnet of Z-axis (parameter aParam:
// Parm_MeasuredMagnetSpace, see enum AlsgParam)
extern "C" __declspec (dllexport) double _stdcall alsgGetParam(int aParam)

// To store the magnet space above the magnet in the parameter.td file as item:
"MAGNET_SPACE_Z="
// Parameters: aParam=Parm_StoredMagnetSpace, aSet=measured space
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetParam(int aParam, double aSet)

/*
The Z-axis position is monitored after each move
If for instance a Run-Reference failed which means that Z is not in home position,
this will be noticed by the ALS-G_API.dll. It will retry to move to its home position
and increase a diagnsotics failure counter. This failure counter is reset at boot time.
*/

// Read Reference Run failures on Z-axis (parameter: aParam=Parm_RefRunZErrorCounts, see
enum AlsgParam)
extern "C" __declspec (dllexport) double _stdcall alsgGetParam(int aParam)

/*
When samplers encounter problems with getting released from the magnet of the Z-axis on move
down,
a special Z-motion can be set in order to release from the magnet.
*/
```

```
// Set special motion Z-axis in order to release from strong magnets
// parameter: aParam=Parm_SpecialMotionZ, aSet=1=true/0=false, see enum AlsgParam)
// Value will be stored in parameter.td file as item "SPECIAL_MOTION_Z="
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetParam(int aParam, double aSet)
```

```
/*
For diagnostic purposes the x, y, z, u and vial sensor sensor state can be read
*/

// Read Sensor state (parameters: nCmd=108 (see enum SysCtrlCmd), dArg1=sensor (x=0, y=1, z=2, u=3, vialsensor=4, x-left-sensor=5, x-right-sensor=6)
extern "C" _declspec (dllexport) double _stdcall alsgSysControl(INT16 nCmd, double dArg1, double dArg2, double dArg3, double dArg4, double dArg5, double dArg6, double dArg7, double dArg8)
```

```
/*
For diagnostic purposes the absolute x, y, z and u position can be read
*/
// Read actual position of x, y, z and u axis. (parameters: ncmd=3(see enum SysCtrlCmd), dArg1=0=current position/1=after_ref_run, dArg2=1)
extern "C" _declspec (dllexport) double _stdcall alsgSysControl(INT16 nCmd, double dArg1, double dArg2, double dArg3, double dArg4, double dArg5, double dArg6, double dArg7, double dArg8)
```

```
/*
Added a 2nd Wash Station.
The following already existing API calls require an extra parameter, the Wash Station number. This is 0 (Wash Station 0/default) or 1 (Wash Station 1)
*/
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetInstallWashStationPos(INT16 aWashStation, double dxPos, double dyPos)
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetInstallWashStationSolventDepth(short int aWashStation, double dzDepth)
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetInstallWashStationWasteDepth(short int aWashStation, double dzDepth)
extern "C" __declspec (dllexport) BOOL _stdcall alsgSetInstallWashStationISTDDepth(short int aWashStation, double dzDepth)

extern "C" __declspec (dllexport) BOOL _stdcall alsgSetConfigWashStation(INT16 aWashStation, INT16 nType)
extern "C" __declspec (dllexport) BOOL _stdcall alsgStepGoToSolvent(INT16 aWashStation, INT16 nPos, double dRelDepth)
```

```
extern "C" __declspec (dllexport) BOOL _stdcall alsgStepGoToSolvent(Int16 aWashStation, Int16 nPos, double dRelDepth)

extern "C" __declspec (dllexport) BOOL _stdcall alsgStepGoToWaste(Int16 aWashStation, Int16 nPos, double dRelDepth)

extern "C" __declspec (dllexport) BOOL _stdcall alsgStepGoToWait(INT16 aWashStation)

extern "C" __declspec (dllexport) BOOL _stdcall alsgRunInit(INT16 aWashStation)


// Set Active Wash Station for high level API calls like RunMethod(...)

extern "C" __declspec (dllexport) BOOL _stdcall alsgSetConfigActiveWashStation(Int16 aWashStation);


/*

Added 2 extra Trays, now 4 in total. These are numbered 0,1,2,3

There are no new API calls required to control 2 extra trays.

*/


/*

Teach new Wash Station and 2 new extra trays

Use the existing the Teach API's in combination with the new Calibration enum values below

*/

public enum nCalibration : short

{

  K_INJ_POINT_1 = 0,

  K_INJ_POINT_2,

  K_INJ_POINT_3,

  K_INJ_POINT_4,

  K_SYR_POS,

  K_TRAY_1,

  K_TRAY_2,

  K_SAMPLE_1,

  K_SAMPLE_2,

  K_WASH_STAT,

  K_SOLVENT,

  K_WASTE,

  K_INT,

  K_WASH_STAT2,     // Wash station 2

  K_SOLVENT2,       // Solvent Wash Station 2

  K_WASTE2,         // Waste Wash Station 2

  K_INT2,           // ISTD Wash Station 2

  K_TRAY_3,         // Tray 3
```

```
  K_TRAY_4,        // Tray 4
  K_SAMPLE_3,      // Tray Sample Depth 3
  K_SAMPLE_4,      // Tray Sample Depth 4
}
```

```
/*
Error definitions supporting 2nd Wash Station and 2 new Trays
*/
Rename of existing errors
#define ERROR_TRAY_0_POS_RANGE          34
#define ERROR_TRAY_1_POS_RANGE          35
#define ERROR_SAMPLE_TRAY_0_DEPTH_RANGE  36
#define ERROR_SAMPLE_TRAY_1_DEPTH_RANGE  37

New Errors:
#define ERROR_TRAY_2_POS_RANGE          42
#define ERROR_TRAY_3_POS_RANGE          43
#define ERROR_SAMPLE_TRAY_2_DEPTH_RANGE  44
#define ERROR_SAMPLE_TRAY_3_DEPTH_RANGE  45
```