Asycube

Mezzo & Forte

Programming Guide

Document	Asyril_ACUBE-MEZ_FOR-02_Programming_Guide_E		
Version	v2.1	**Date**	01.06.2015
Articles	ACUBE-MEZ-02 ACUBE-FOR-02		

Table of Contents

1. Introduction

1.1. Generalities

The following document is the property of Asyril S.A. and may not be copied or circulated without permission. The information contained in this document is subject to change without notice for the purpose of product improvement. Before operating your product, please read this document in order to ensure a correct use of the product. Nevertheless, if you meet difficulties during the operation or the maintenance, please, feel free to contact Asyril customer service.

In this manual, the safety precautions that you must respect are classified as: "Danger", "Warning" and "Note"; the following symbols are used:

DANGER!

Failure to observe the instruction may result in death or serious injury.

DANGER!

Failure to observe the instruction may result in electrocution or serious injury due to electric shock

WARNING!

Failure to observe the instruction may result in injury or property damage.

NOTE :

The user should read carefully this information to ensure the correct use of the product, although failure to do so would not result in injury.

Refer to …

For more information on a specific subject, the reader should read other manual, or refer to other paragraph.

WARNING!

Asyril shall not be liable whatsoever for any loss or damage arising from a failure to observe the items specified in "Safety Precautions" of the OPERATING MANUAL. The customer is responsible to provide the necessary instruction to the persons concerned.

NOTE :

All dimensions in this document are expressed in millimeters

1.2. Related manuals

As described in Table 1-1, this manual is an integral part of the Asycube documentation set. This manual covers the information about how to use and integrate an Asycube Mezzo or Forte.

Manual Title	Manual reference	Description of the content
Asycube Unpacking Instructions	ACUBE-MEZ-FOR-02_Unpacking_Instructions	Describes how to unpack your asycube.
Asycube Operating manual	ACUBE-MEZ-FOR-02_Operating_Manual	Technical description, safety precautions, installation, maintenance and reparation information
Asycube HMI manual	ACUBE-MEZ-FOR-LA5_User_Interface_Manual	Describes how to use the HMI (simply move the parts, and configure the Asycube's vibrations …)
Asycube Programming guide	ACUBE-MEZ-FOR-02_Programming_Guide	THIS MANUAL
Asycube DLL	ACUBE-DLL_Integration_Guide	Describes the Asycube dll's that are High-level libraries which helps to integrate Asycubes using .Net 3.5 environment.

Table 1-1: related manuals

2. How to use the Asycube Mezzo and Forte

This chapter gives the main information about the use and tuning of the Asycube. It shows general information and behavior, presents the main procedure from setup to running in production with an Asycube Mezzo or Forte and describes then each step. The next chapters will detail the chosen working mode.

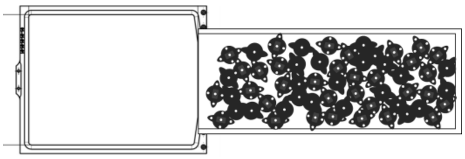
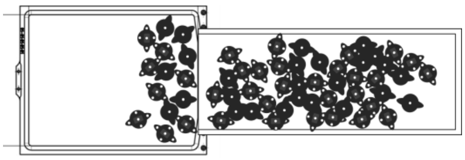
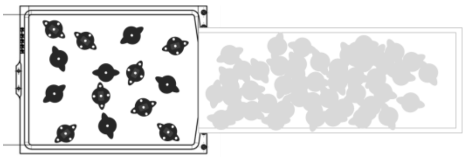
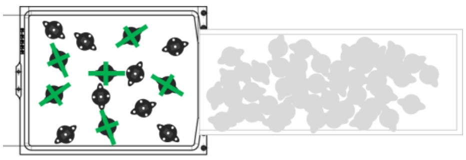
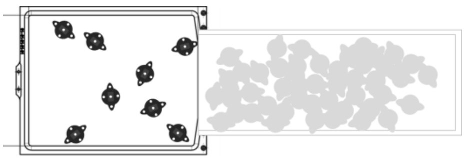
2.1. Introduction

The Asycube is a flexible feeding system, which can spread the parts on the picking surface by using smart vibration. The displacements in all directions are the result of the combination of the vibration of the different actuators.

Here below are some main definitions and descriptions of the basic behavior of the Asycube.

The table shows the main sequence of use of the Asycube starting from an initial empty state with interaction of a vision system and a robot. The role of the Asycube is to:

- Distribute and flip the components over the surface
- Manage the feeding from bulk to have enough components on the picking surface

Initial state		#AllParts < threshold ⇨ Feed
Asycube feeds from bulk		#AllParts < threshold ⇨ Feeding + spreading sequence
Asycube spreads components with intelligent vibration		
A **vision system** detects which parts are ready to be taken		Get the lists of AllParts & GoodParts Positions
A **robot** picks good parts and assembles them		

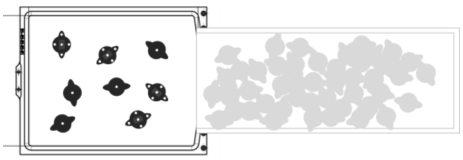
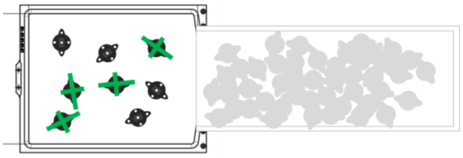
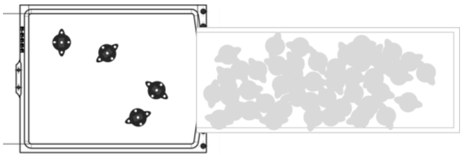
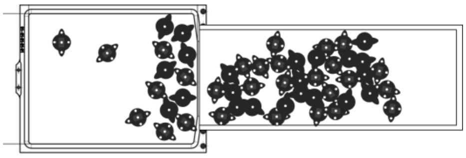
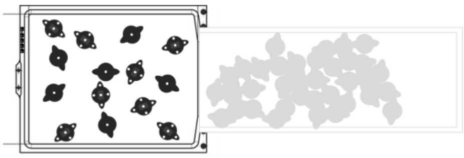
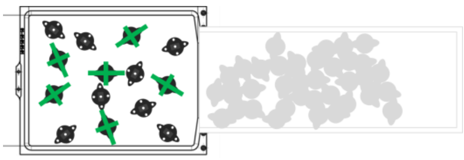
Asycube spreads components with intelligent vibration		#AllParts > threshold ⇨ Spreding sequence
A **vision system** detects which parts are ready to be taken		Get the lists of AllParts & GoodParts Positions
A **robot** picks good parts and assembles them		
Asycube feeds from bulk		#AllParts < threshold ⇨ Feeding + spreading sequence
Asycube spreads components with intelligent vibration		
A **vision system** detects which parts are ready to be taken and a robot picks them		…

Table 2-1: sequence of use of a flexible feeder

2.2. Procedure from connecting until production run

Figure 2-1 shows the different interfaces within the Asycube, while Table 2-2 describes the different steps of set up, configuration and tuning and refers to the corresponding section of this documentation.

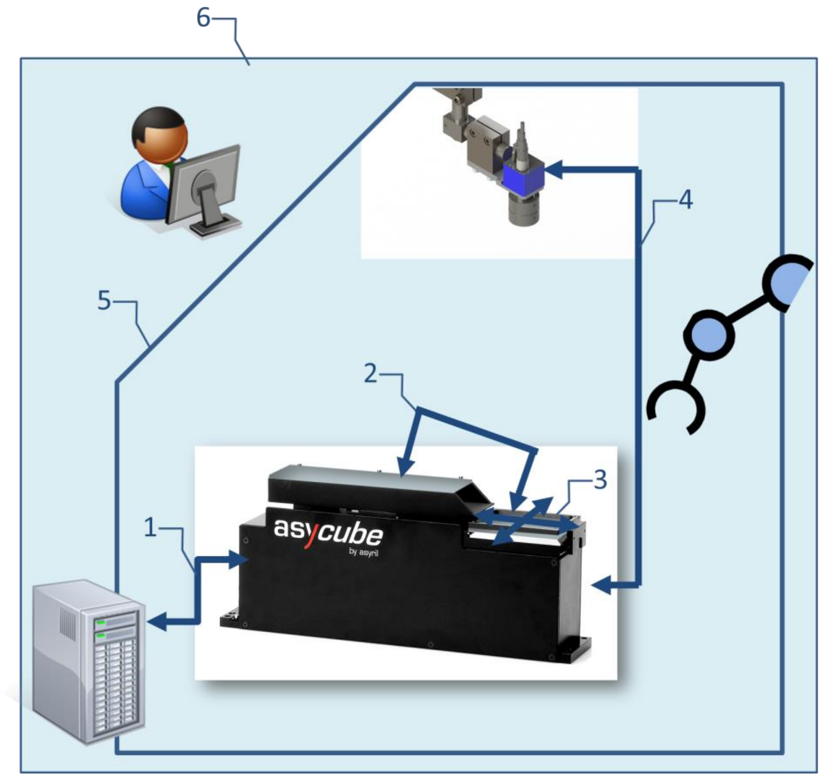


Figure 2-1: the Asycube in situation

	Step	Subject	Section
1	**Connection & Communication**	This section shows different ways to integrate and work with an Asycube.	§2.3
2	**Bulk feeding**	Connection and configuration of the outputs are detailed.	§2.4
3	**Vibrating platform**	This section describes how to get an optimal spreading and flipping behavior of the parts on the vibrating platform.	§2.5
4	**Parts detection**	It shows the backlight configuration to ensure the parts detection.	§2.6
5	**Process**	This section shows how to synchronize the Asycube within the feeding operation and how to parameterize this sequence in function of the parts distribution over the vibrating platform.	§2.7
6	**Saving**	Functionalities and contents of recipes are exposed here.	§2.8

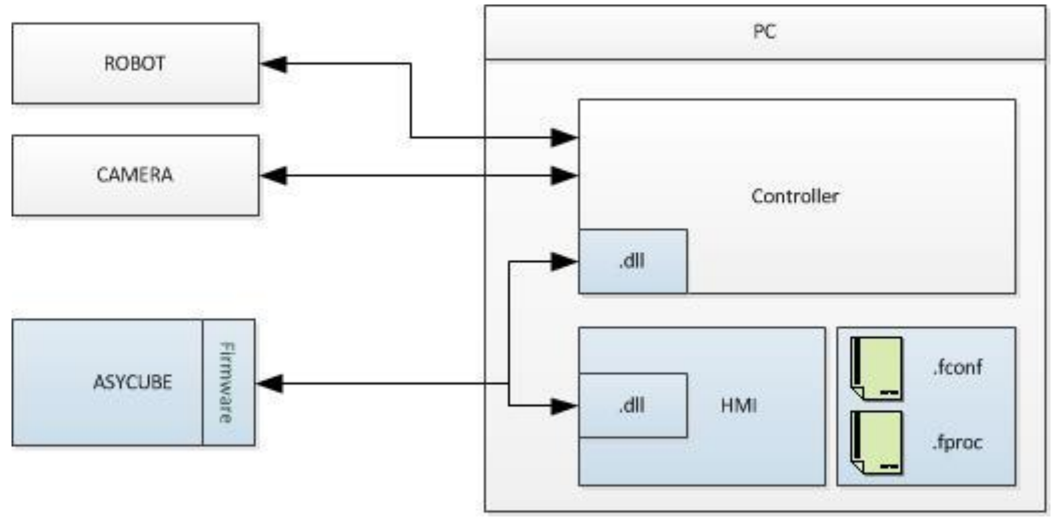
Table 2-2: the steps from setup to production with an Asycube

2.3. Integration modes

Different ways of integration are available. Here below is a brief description of the main ones. More information is then available in the corresponding section. The main tasks necessary to use, configure and integrate the Asycube are described depending the chosen integration mode. The next tables and figures describe the tasks in charge of the Asycube and the ones due to the integrator. The light blue color represents the levels offered by Asyril, the white one the levels in charge of the integrator.

2.3.1. HMI to configure and smart vibration process through Asycube Library

An HMI is available to test, configure and optimize the different movements of the components on the vibrating surface, to parameterize the behavior of the outputs and backlight and to adjust the process parameters. It allows the loading/saving of recipes. Chapter 3 gives the requirements for the installation and use of the HMI.



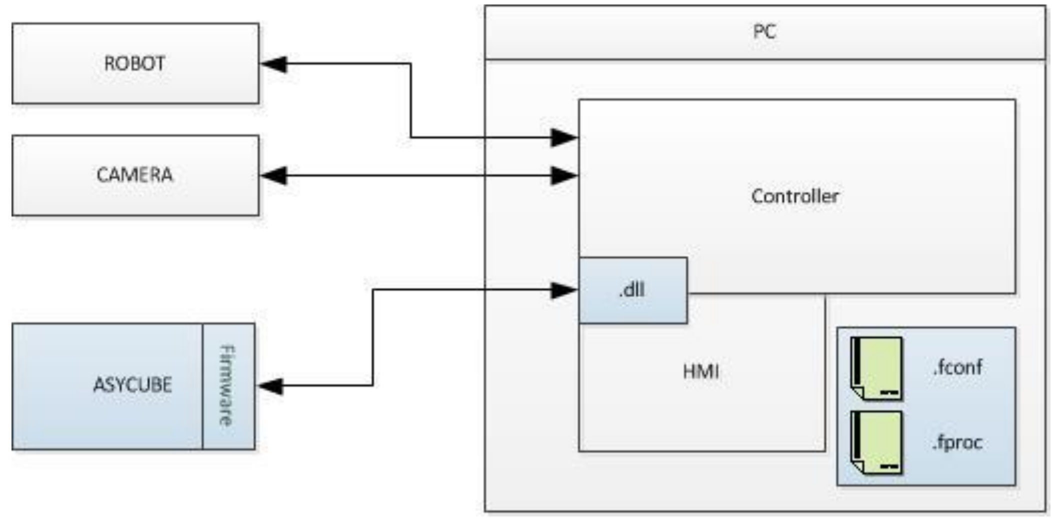
LEVEL	TASKS
Asycube	• Generation and synchronization of the desired vibration • Activation of the outputs and backlight
Asycube Library	• Communication with the Asycube • Treatment of the recipe (read/write parameters) • Choice/Management of the optimal vibration based on the vision feedback (run the process) • Easy-tuning function to help the configuration and optimization of the vibration parameters
Asycube HMI	• Configuration and optimization of the feeding (vibration) parameters with easy-tuning method and for advanced users • Configuration of the process (HMI) • Recipe management
Integrator	• Synchronization of the feeder and the machine

Table 2-3: Integration mode: Asycube with Library and HMI

2.3.2. Interface through the Asycube Library

The Asycube Library is available to manage the communication with the Asycube. It includes also the smartness allowing the management of the optimal vibration based on the feedback information from a vision system, of the recipes and a tool to simplify the tuning of the vibration. Chapter 4 details the use, requirements and functions dedicated to the Library.

Recipe files can be generated and loaded by the Library. A recipe comprises two xml files: one for the batch configuration (.Fconf), the second one for the process description (.Fproc). Section 2.8 details their contents, structure and creation. The HMI includes the Asycube Library. In consequence, any controller using the Library can either load the recipes files generated by the HMI.

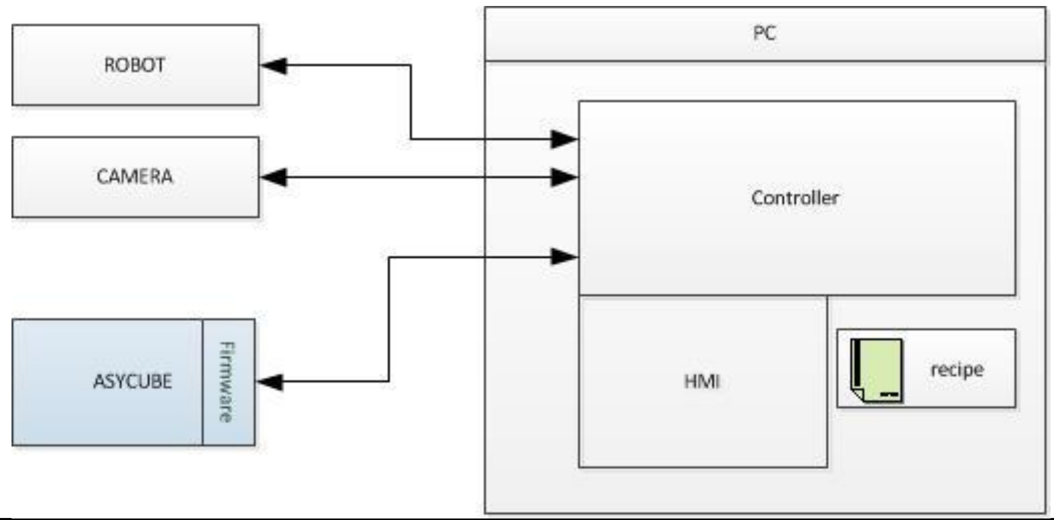


LEVEL	TASKS
Asycube	• Generation and synchronization of the desired vibration • Activation of the outputs and backlight
Asycube Library	• Communication with the Asycube • Treatment of the recipe (read/write parameters) • Choice/Management of the optimal vibration based on the vision feedback (run the process) • Easy-tuning function available to help the configuration and optimization of the vibration parameters
Integrator	• Configuration and optimization of the feeding (vibration) parameters (HMI) • Configuration of the process (HMI) • Recipe management • Synchronization of the feeder and the machine

Table 2-4: Integration mode: Asycube Library

2.3.3. Connecting the Asycube

The Asycube contains its own electronics able to execute and stop the vibration, load different set of values, commute the backlight and activate the outputs. It concerns all elements to generate and synchronize the desired vibration and to activate the outputs and backlight. Chapter 5 details the communication protocol. Chapter 5.4 lists the commands and chapter 6 the parameters.



LEVEL	TASKS
Asycube	• Generation and synchronization of the desired vibration • Activation of the outputs and backlight
Integrator	• Communication with the Asycube • Treatment of the recipe (read/write parameters) • Recipe management • Configuration and optimization of the feeding (vibration) parameters (HMI) • Configuration of the process (HMI) • Choice/Management of the optimal vibration based on the vision feedback (run the process) • Synchronization of the feeder and the machine

Table 2-5: Integration mode: Asycube only

2.4. Bulk feeding

The Asycubes Mezzo and Forte have an integrated reservoir to feed the vibrating surface with as many parts as necessary to get a stable number of available parts for the manipulator.

The way to synchronize the reservoir within the feeding operation is directly related to the Process definition and will be described in section 2.7.

Table 2-6 shows the available parameters to tune the behavior of the reservoir and Table 2-7 gives the main configuration to get the Forward and Backward movement.

Sets of data named "batches" are used to custom the reservoir behavior. Twenty-six batches are available and contain the three parameters of Table 2-6. Section 6.7 « Reservoir activation (B for Bulk) » details the corresponding activation commands.

Label	Explanations	Unit	Range
Amplitude	Corresponds to the power of the signal sent to the actuator and defined as the percentage of the maximum amplitude	[%]	0 to 100 %
Frequency	The frequency of the signal sent to the actuator	[Hz]	0 to 350 Hz
Waveform	Waveform is defined as the shape of the signal	[-]	0=no signal, 1=sinus, 2=ramp up, 3= ramp down

Table 2-6: reservoir parameters

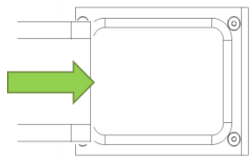
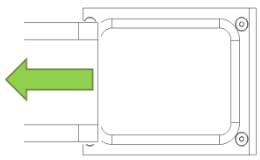
Direction	Batch		Frequency range	Waveform
Forward		A	60-80 Hz	Ramp up
Backward		B	60-80 Hz	Ramp down

Table 2-7 : pre-defined reservoir movements and corresponding batches with the optimal frequency range.

NOTE :
The Backward movement is not feasible with a bulk 45°. This configuration has to be specified at order and allow a more robust Forward feeding.

2.5. Vibrating platform

The movement of the parts on the vibrating surface depends on the combination and the excitation of the actuators of the Asycube. A good behavior is achieved when:

- the parts are sliding on the surface and are able to cross the plate in all direction;
- only the Flip order makes them jumping.

Each of the actuators can be adjusted with its own set of parameters:

Label	Explanations	Unit	Range
Amplitude	Corresponds to the power of the signal sent to each actuator and defined as the percentage of the maximum amplitude	[%]	0 to 100 %
Frequency	The frequency of the signal sent to each actuator	[Hz]	0 to 350 Hz
Phase	Indicates the Phase shift between the first and the reference actuator signal	[°]	0 to 360 degrees
Waveform	Waveform is defined as the shape of the signal	[-]	0=no signal, 1=sinus, 2=ramp up, 3= ramp down

Table 2-8: actuators parameters

A “batch of vibration” represents the collection of these data for all actuators of the platform and duration. It allows synchronizing their vibration to get a specific displacement/behavior of the parts.

Additionally to the vibration parameters, the duration of vibration corresponds to the time for the parts to cross the plate in the given direction. The process (see section 2.7) will use this value to compute the real vibration duration based on the positions of the parts on the plate and in order to center them on the picking surface.

2.5.1. Definitions

- **Batch**: combination of the vibrations of each actuator to get the desired movement of the parts on the picking surface; there are 26 batches named from A to Z, each of them contains 4 parameters for each of the 4 actuators and vibration duration, so a total of 17 parameters.
- **Process**: optimal sequence of vibration to feed, distribute, orient and flip the components on the picking surface. The sequence varies depending the number of parts and their location on the platform (§2.7)
- **Pre-defined movements** : displacement and flip of the parts on the picking surface; by convention these movements corresponds to the batches A to I
- **Custom movement/batch**: all other vibration combinations; by convention they are stored in batches J to Z.

2.5.2. Tuning of the movements

Rules to tune the movements of the parts on the vibrating platform:

- The best way to get smooth displacement is to use sinus waveform.
- The frequency is always identical for all actuators in a batch.
- All movements use a vertical vibration, smooth for lateral displacement and stronger to flip the parts.

The frequency is usually the same for all movements but the Flip. Once you have found a good behavior in forward and backward, you can usually use the same frequency for all other displacements.

The direction of the parts movement on the platform is the result of the combination of the two in-plane actuators. Switching the phase from 0° to 180° inverts the orientation of the movement (see example on Figure 2-2). If the parts are not moving perfectly on the target, adjusting the amplitude of both in-plane actuators will correct the direction.

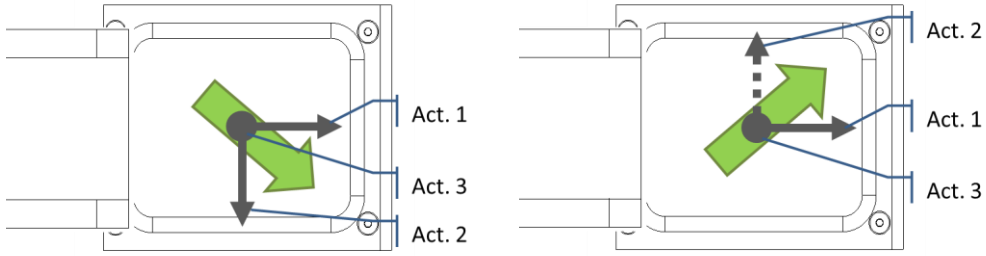


Figure 2-2 : actuators disposition and example of activation to get a diagonal movement of the parts (green arrow) (solid line: 0° phase, dashed line: 180° phase)

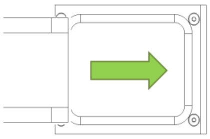
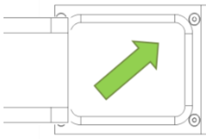
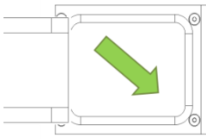
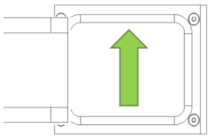
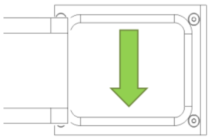
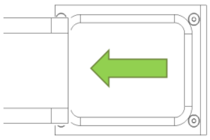
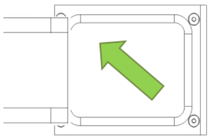
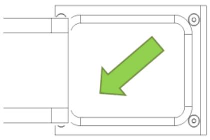
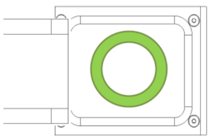
Direction	Batch		Frequency range
Forward		A	
Forward Left		B	
Forward Right		C	
Left		D	
Right		E	60 – 90 Hz
Backward		F	
Backward Left		G	
Backward Right		H	
Flip		I	90 – 110 Hz

Table 2-9: pre-defined movements and corresponding batches with the optimal frequency range.

2.6. Parts detection

2.6.1. Backlight activation

There are two ways to activate the backlight illumination and to synchronize it with the image acquisition:

1. By software commands
    a. Commands allow switching ON and OFF; the duration corresponds then to the time between both commands.
    b. A flashing working mode is also available with a parameterized duration (see 6.4 for the detail of the commands).
2. By hardware connection through the "Backlight synchronization" connector
    a. Using the backlight synchronization input, the illumination time corresponds to the length of the pulse signal.

When using both types of activation the backlight will be ON if the command OR the signal are activated.

There is a timeout implemented to prevent any heating damage on the system. Once achieved it is necessary to validate the alarm by setting OFF the backlight. The timeout is set to 30s at 100% intensity and is proportionally increased for lower intensity.

2.6.2. Intensity

The intensity of illumination can be varied from 0% to 100%. The defined illumination level is then applied for both activation types (see 7.1 for the corresponding parameters).

NOTE :
The backlight color has to be specified when ordered.

For more information on the electrical interface and connector reference of the backlight synchronization, the reader should refer to the Operating Manual.

2.7. Process

The process is a configurable sequence of vibrations that is then adjusted within the number of parts and their location on the surface. The input of the process is so the feedback of the vision system. The output is the sequence of the optimal batches to distribute and flip the parts as well as the activation of the outputs to feed the platform from bulk. A brief description is provided here below. For more information, please refer to sections 4.4 and 4.5.

2.7.1. Configuring the vibration sequences

Depending on the number of parts detected on the Asycube platform, it is possible to configure a specific vibration sequence. Between each configured sequence, the vibration time is interpolated linearly.
Example:
If, for 10 parts, the bulk must vibrate 1000 ms, and for 20 parts it must vibrate 500 ms, for 15 parts found on the platform, the bulk will be vibrated for 750 ms).

All the configured batches can be used in the sequence. To distribute the parts uniformly on the platform and take into account the position of the parts on the platform, the “calculated” option is included in the sequence. In this case, the algorithm will automatically define the vibration time and the optimal batch. A typical sequence may be:

- Calculated
- Flip
- Stabilization

In order to reduce the parts stabilization time or to orient specifically the parts, the Asycube platform may be machined (grooves, holes, etc.). In this case, the vibration sequence must be adapted to the type of platform (so that the parts are directly positioned in the grooves or the holes for example). In the case of a grooved platform, a typical vibration sequence may be:

- Flip
- Forward
- Backward

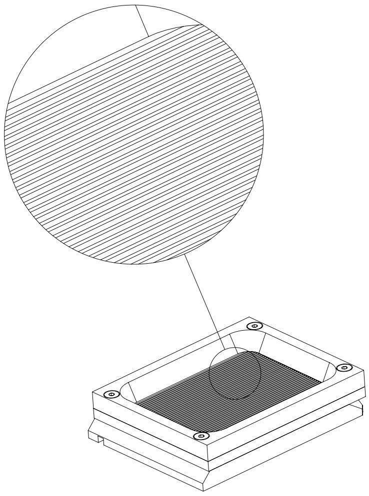


Figure 2-3: Example of a grooved platform

2.7.2. Advanced mode

If you are working with the Asyril HMI, you can configure your process with a “normal” mode or an “advanced” mode. In "advanced" mode, it is possible to differentiate the desired vibration sequence for the feeding and recirculation phases as:

- Feeding:
    - this sequence is normally used at the initial step or at the end of a production when none or not a lot of parts are in the field of view. Its goal is to transport the parts from bulk to the field of view
- Recirculation:
    - this sequence should prevent an overflow of parts in the field of view by preventing any part to come in the field of view
- Working:
    - it corresponds to the optimal state where the aim is to compensate the taken parts and to distribute the parts on the platform. The feeder should work most of the time with this manner. If not, the different thresholds and sequences should be modified. The cycle time is also directly related to this configuration.

In the “normal” mode, only the Working sequences are configurable. The Feeding and Recirculation ones are automatically set up.

2.8. Recipe

If you are using the Asyril HMI or the Asycube Library, the parameters of vibration and of the sequences are stored in two xml files that composed the recipe for each kind of parts as described in Table 2-10.

Label	Content	Dependency
.Fconf	The parameters of the different batchs (amplitude, frequency, duration…)	geometry and size of the parts
.Fproc	The sequence of vibration and the corresponding thresholds of parts	sequences depend on the geometry of the plate the threshold are related to the size of the parts.

Table 2-10 : recipe files

3. Asyril HMI

NOTE:

If a computer is included in your product, the "HMI" software is already installed.

This chapter explains how to install the HMI on your own computer if needed.

3.1. Prerequisite

To install and execute the HMI you need those elements:

- Computer with windows 7 (or 8) 64bits (a 32 bits version of HMI can be delivered on request)
- .Net 4.0 minimum installed.
- User access defined to be able to install and execute software.

NOTE:

Check that your version of the ".net framework" is up to date. You can download this version from the Microsoft website: http://www.microsoft.com/download/

3.2. Installing the HMI software on a specific computer

3.2.1. Pre-installing

Step 1	Insert the USB key into the computer on which the HMI should be installed
Step 2	Double-click on the SurfaceToolkitRuntime.msi executable file to launch the installation procedure
Step 3	Accept the license agreement and follow the instructions provided by the wizard

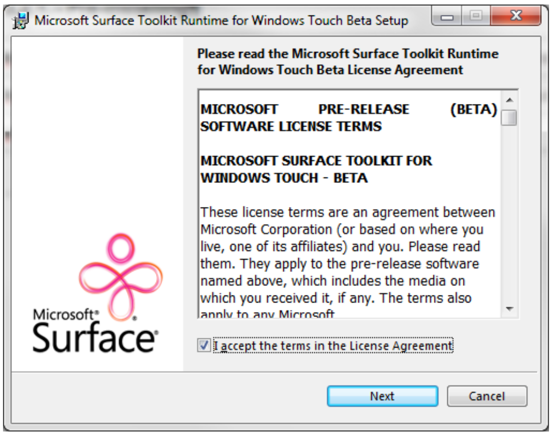


Figure 3-1: HMI setup wizard

Step 4	When installation is complete, click on "Finish" to close the wizard

3.2.2. Installing

Step 1	Insert the USB key into the computer on which the HMI should be installed
Step 2	Double-click on the setup.exe executable file to launch the installation procedure

Figure 3-2: HMI setup wizard

Step 3	Follow the instructions provided by the wizard
Step 4	When installation is complete, click on "close" to close the wizard

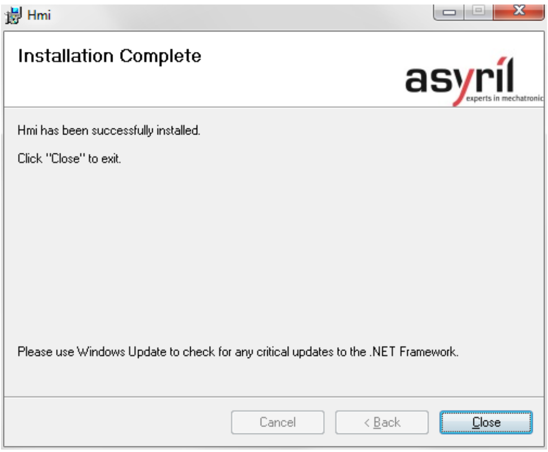


Figure 3-3: Installation complete

3.2.3. Starting HMI

Step 1	Click on the shortcut created on the desktop
Step 2	If needed, configure the HMI depending of your products on the configuration page

3.2.4. Configuring HMI

On the first start, HMI contains no product. You have to add your product in the configuration page to be able to work with it.

The following explanation shows you how to do that:

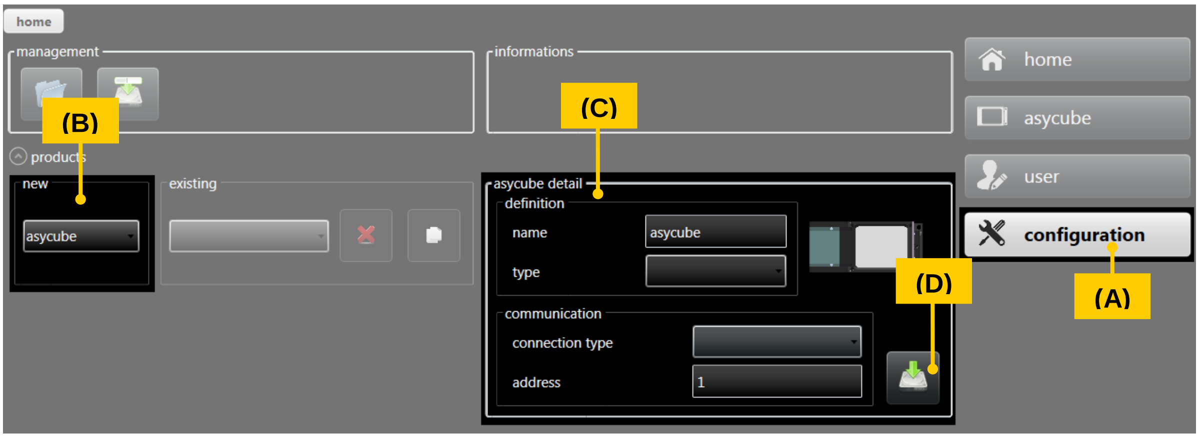


Figure 3-4: HMI Configuration page

Ref.	Designation	Description
(A)	*Configuration" button*	Press this button to display the Configuration screen.
(B)	*"New product" list*	Select in this list the product you want to add (an Asycube in this example).
(C)	*“Asycube parameters” window*	Fill all parameters of the product you choose.
(D)	*"Save" button*	Use this button to save the configuration changes. Then you need to restart HMI to apply your modifications.

Table 3-1 : HMI configuration page description

For more explanations, see HMI documentation.

3.2.5. Troubleshooting

Ref.	Problem	Solution
1	*HMI crashes on starting*	Try to start HMI with administrator access (right-click on shortcut, Properties/Compatibility, select “Run this program as an administrator”).
2	*HMI starts but no button are displayed*	The SurfaceToolKit is missing or improperly installed.

Table 3-2 : Troubleshooting listing

4. Asycube Library

4.1. Prerequisite

To integrate and use the Asycube Library you need those elements:

- Computer with windows 7 (or 8) any CPU,
- .Net 3.5 minimum installed.

NOTE:

Check that your version of the ".net framework" is up to date. You can download this version from the Microsoft website: http://www.microsoft.com/download/

4.2. Content of the Library and its documentation

The Asycube Library allows to

- Communicate with an Asycube
- Treat the recipe (read/write parameters)
- Configure and optimize the vibration parameters
- Configure and run the process

This chapter presents the main information about the architecture of the library, the easy-tuning method and the usability and configuration of the process. For all other information, methods, properties, please refer to the HTML Help of the library.
Simple examples of how to work with the asycube and its library are also included in the HTML Help.

4.3. Architecture

The Asycube Library is composed by two dll:

- *Asyril.IFeeder* represents a Feeder Interface. It contains generic property, enums, structures and functions for feeding systems.
- *Asyril.AsyCube* is a concrete implementation of the IFeeder Interface. It contains specific properties, structures, enums and methods to work with asycubes.

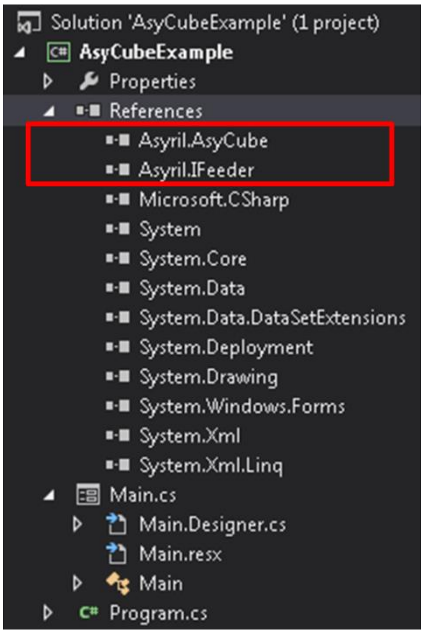


Figure 4-1 : The first step for working with the Library is to create a project and to add both dll *Asyril.IFeeder* and *Asyril.AsyCube* to the references.

The architecture of the *Asyril.AsyCube* library is similar to the physical device: it contains three parts, named Backlight, Platform and Reservoir. Additionally, a fourth part, called Process, defines the intelligence of the asycube.

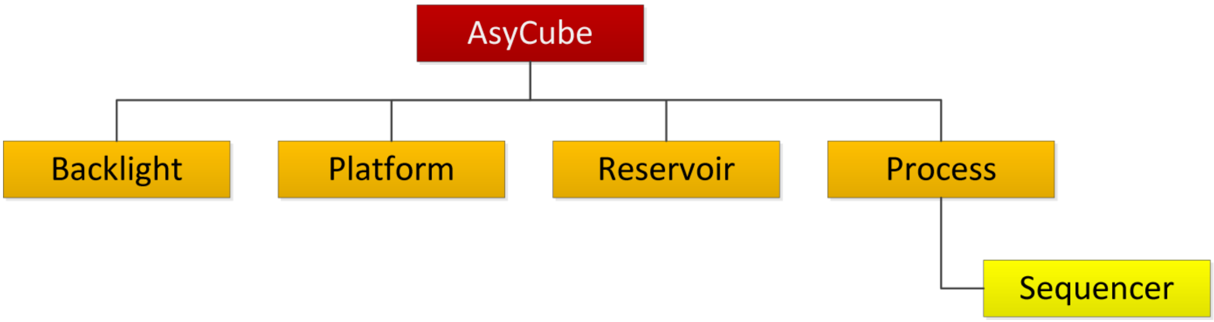


Figure 4-2 : Structure of the *Asyril.AsyCube* library

4.3.1. AsyCube

The AsyCube object represents the physical device. It handles the communication and the generic parameters of the asycube like for example the lifetime. It also proposes the load and save functions. The methods and properties are accessible over AsyCube.XXX.

Example:

- AsyCube.EnableSystem = true; //Enable the AsyCube
- AsyCube.Connect(); //Connect to the AsyCube

4.3.2. Backlight

The Backlight handels all parameters and methods that can change the behavior of it. The Backlight is accessible as a public field over the AsyCube object. The methods and properties are accessible over AsyCube.Backlight.XXX.

Example:

- AsyCube.Backlight.Intensity = 100; //Change the intensity of the Backlight
- AsyCube.Backlight.State = true; //Turn the Backlight on

4.3.3. Platform

The Platform proposes all methods that handle the behavior of the vibrating platform. It is accessible as a public field over the AsyCube object. The methods are accessible over AsyCube.Platform.XXX.

Example:

- AsyCube.Platform.GetBatch(BatchName.A); //Get the Batch A of the Platform
- AsyCube.Platform.Vibrate(BatchName.A,1000,false,false); //Start a vibration on the Platform

4.3.4. Reservoir

The Reservoir proposes all methods that handle the behavior of the reservoir. It is accessible as a public field over the AsyCube object. The methods are accessible over AsyCube.Reservoir.XXX.

Example:

- AsyCube.Reservoir.GetBatch(BatchName.A); //Get the Batch A of the Reservoir
- AsyCube.Reservoir.Vibrate(BatchName.A,1000,false,false); //Start a vibration on the Reservoir

4.3.5. Process

The Process represents the intelligence of the asycube. Different as the three other public fields, the Process represents an array of public fields. This solution ensures forward compatibility for further functionality but today only one Process per asycube can be called. The methods and properties can be called with AsyCube.Process[index].XXX.

Example:

- AsyCube.Process[0].Load("…\test.fproc") //Load a feeder process file

Some commands can or must be directly addressed to the Sequencer. In this case the properties and methods are called like AsyCube.Process[index].Sequencer.XXX.

Example:

- AsyCube.Process[0].Sequencer.AdvancedConfiguration = true; //Set AdvancedConfiguration mode
- AsyCube.Process[0].Sequencer.Prepare(); //Prepare the sequencer for the next run

For more information about the Process and the configuration of the sequencer, please refer to sections 4.4 and 4.5

4.4. Use of the process within a production feeding machine

Figure 4-3 presents the interaction between the Client, the Library and the Asycube during a feeding process.

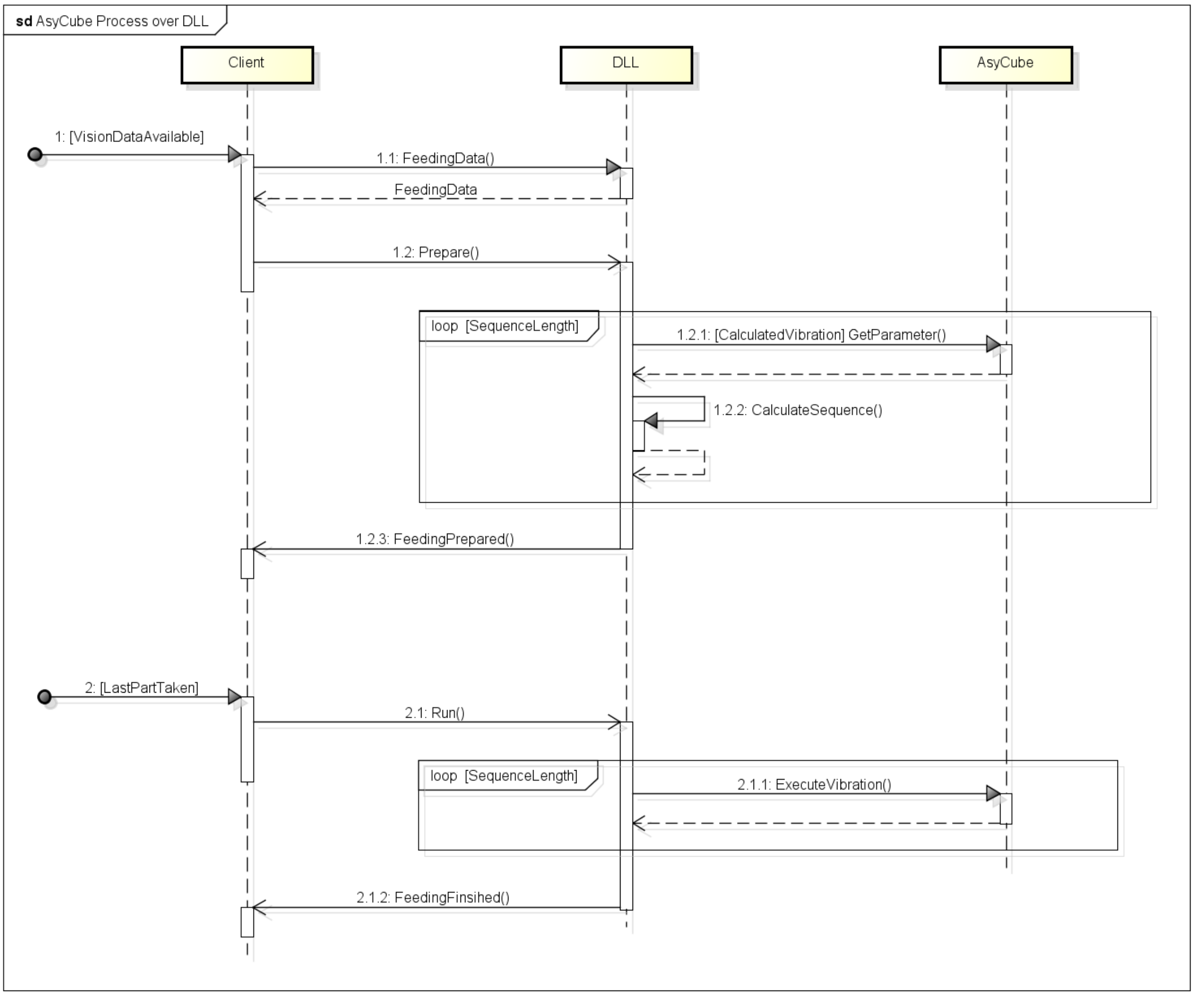


Figure 4-3 : Sequence diagram

4.4.1. Feeding Data

Data coming from the vision system has to be transferred to the sequencer to choose the optimal vibration sequence. This information can be of different ways:

- List of the positions of all parts and of good parts,
- List of the positions of all remaining parts,
- Center of mass of the remaining parts distribution and number of the remaining parts.

The direction of the feeder has to be related to the vision system. For the feeder, the coordinates are as in Figure 4-4, with the origin at the center of the plate and X-axis parallel to the reservoir (bulk). The part positions have to follow this coordinate system in order to manage the right vibration direction.

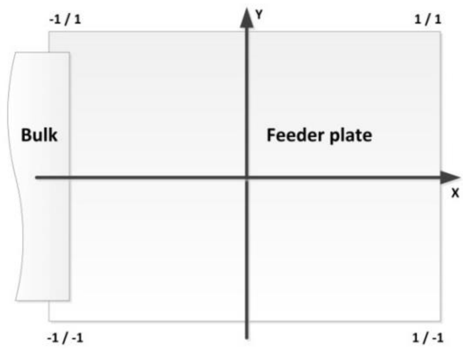


Figure 4-4 : Coordinate system of the vibratory platform

4.5. Configuration of the process

The Process is built as a sequencer that has three different sections, named Feeding, Working and Recirculation. They differentiate the desired vibration sequences depending on the number of parts on the platform. Each of them has its own assignment and is described here below.

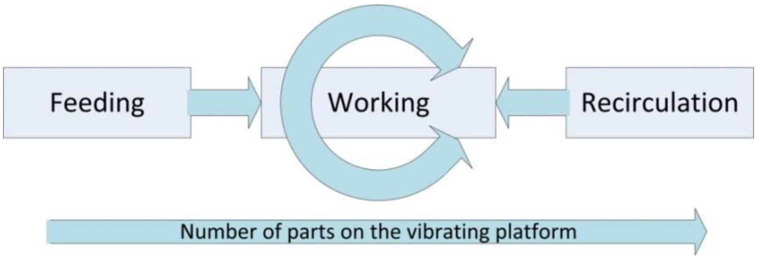


Figure 4-5 : sections of the process sequencer

- Working:
  - this section corresponds to the optimal state where the aim is to compensate the taken parts and to distribute the parts on the platform. The feeder should work most of the time within this section. If not, the different thresholds and sequences should be modified. The cycle time is also directly related to this section;
- Feeding:
  - this sequence is normally used at the initial step or at the end of a production when none or not a lot of parts are in the field of view. Its goal is to transport the parts from bulk to the field of view;
  - the number of parts found on the surface is less than a defined minimum threshold;
- Recirculation:
  - this sequence should prevent an overflow of parts in the field of view by preventing any part to come in the field of view;
  - the number of parts found on the surface is more than a defined maximum threshold.

4.5.1. Structure of the sequencer

Each section comprises one or more sequences that define a movement for a given number of parts in the field of view. The sequences are configured with subsequences of one or more vibration commands. Figure 4-6 shows the structure of the sections.

A single vibration command has four parameters (see Table 4-1 for the detailed list):

- the location of the vibration,
- the direction of the vibration,
- the corresponding specific batch,
- the duration of the vibration in milliseconds.

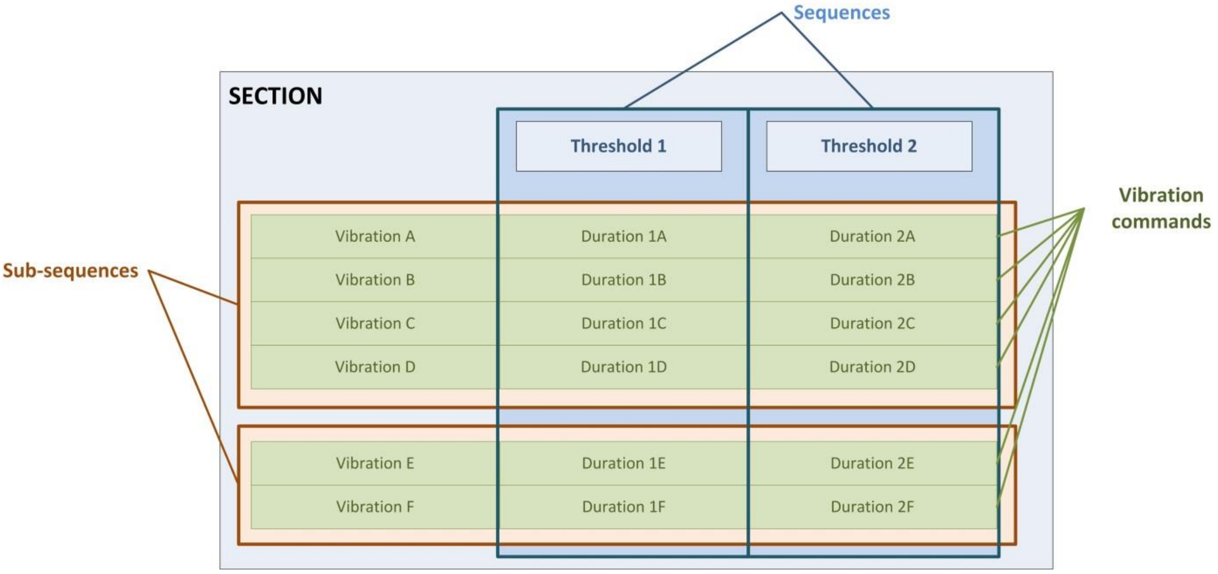


Figure 4-6 : structure of the sections in sequence – subsequence – vibration command.

Location	Direction	Batch	Usability
Reservoir	Forward	A	
	Backward	B	
	Custom	C – Z	Execute a custom vibration with one of the batches C to Z as defined.
Platform	Forward	A	
	Forward Left	B	
	Forward Right	C	
	Left	D	
	Right	E	
	Backward	F	
	Backward Left	G	
	Backward Right	H	
	Flip	I	
	Custom	J – Z	Execute a custom vibration with one of the batches J to Z as defined.
	Calculated	None	Execute a calculated vibration to optimize the parts distribution.
None	Stabilization	None	Wait a defined time (usually until the parts are stabilized).

Table 4-1 : list of the vibration command parameters

4.5.2. Parts thresholds

The different sections can have different number of subsequences with different vibrations commands:

- If the number of parts detected in the field of view is *smaller* than the threshold of the lowest sequence, then the subsequences of the lowest sequence are executed.
- If the number of parts detected in the field of view is *bigger* than the threshold of the highest sequence, then the subsequences of the highest sequence are executed.
- If the number of parts detected in the field of view is *between* two indicated threshold, then the duration values are linearly interpolated. This case can only appear within a section and supposes that the different section thresholds differ only by a value of 1.

An example of the different sequences is shown in Figure 4-7 with a graphical view of the executed vibration duration according to the number of parts detected on the platform.

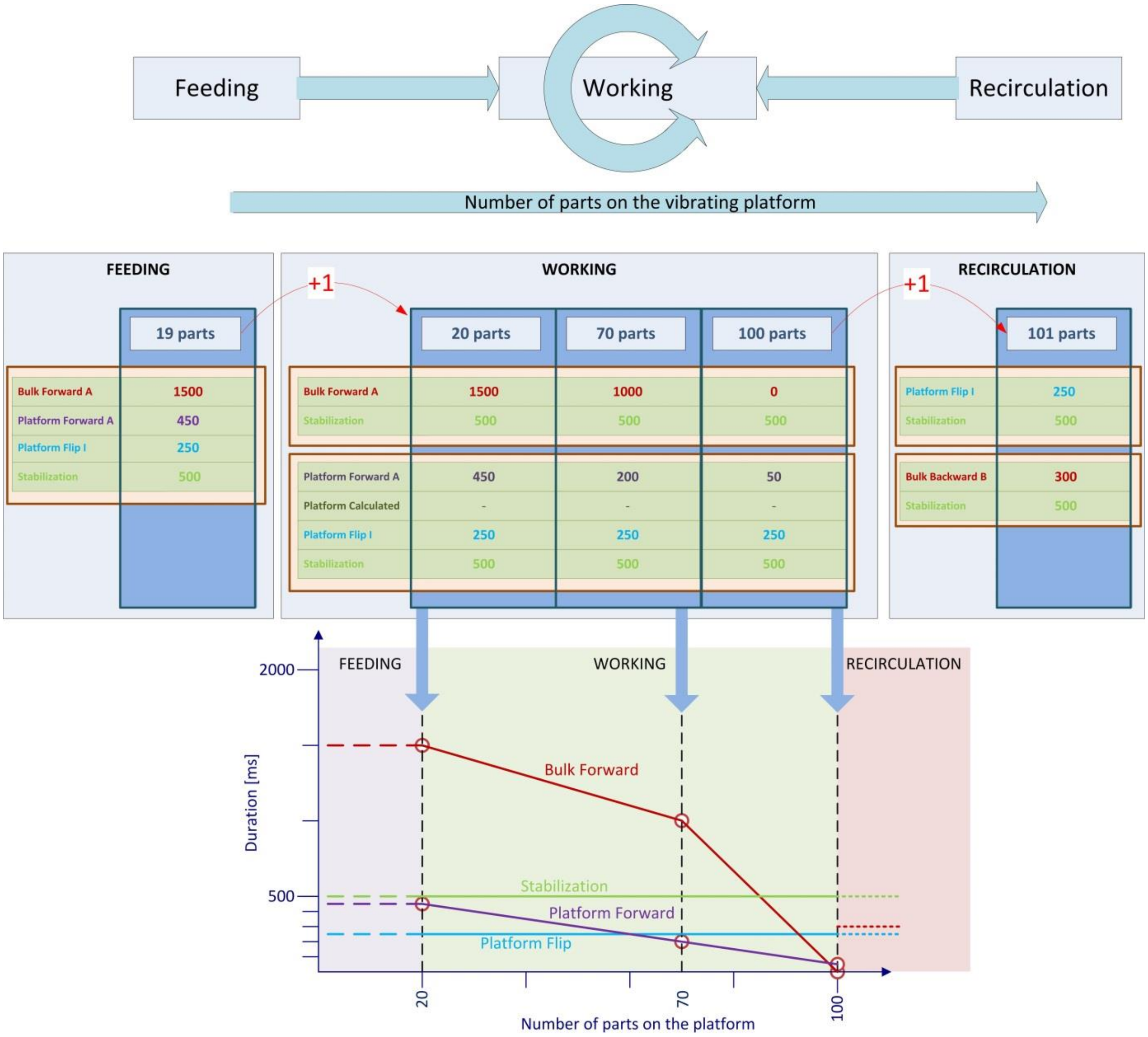


Figure 4-7 : Example of sequences for the different three sections and their corresponding execution vibration according to the number of parts on the platform.

4.5.3. Subsequence synchronization

Usually one subsequence corresponds to the vibrations of the platform and a second one to the ones of the bulk. Their execution may then be synchronized to reduce the vibration duration or to optimize the behavior as following (Figure 4-8):

- NONE: All subsequences will be executed each after the other, purely sequentially.
- STARTING: All the subsequences will start at the same time.
- STOPPING: All the subsequences will stop at the same time.

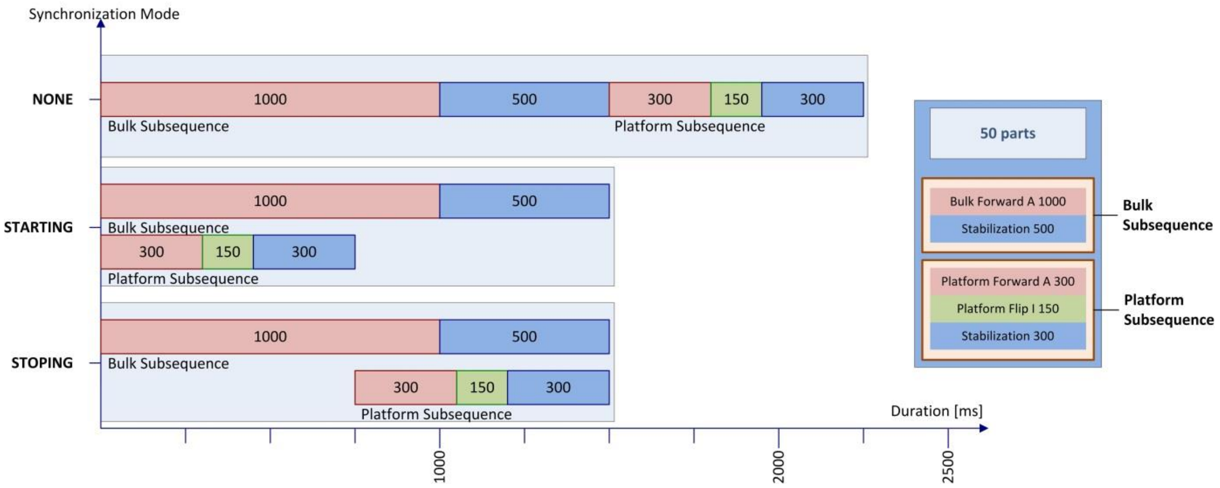


Figure 4-8 : comparison of the different synchronization modes

NOTE:

For configuration purpose of the sequencer, it is possible to insert a vibration command in a specified section and subsequence at a defined index or to add a vibration command at the end of the specified subsequence. Pay attention that the vibration command will be add/insert in all sequences of the concerned section and with a default duration value (0).

5. Communication with asycube

The host computer communicates with the asycube using protocol RS485, RS232 or Ethernet (TCP/IP). In all cases, the asycube is the server and the host computer is the client. The server (the asycube) sends packets only after a client request.

5.1. RS232 or RS485

The Asycube units are either factory configured for RS-232 or RS-485 “AsyLink” protocol operations. Depending on the configuration, the connection has to be made differently. RS-232 is used when only one Asycube system is integrated in the final machine whereas RS-485 is used when an Asycube-Network has to be established.

For more information on the installation of an Asycube RS-485 network (default installation), please refer to section 5.1.4 “Asycube Serial Network”.

Default RS232 or RS485 parameters are:

Baud rate	19200
Start bit	1
Data bits	8
Stop bit	1
Parity	None
Checksum	None
Allowed node address	• RS-232 : #0 • RS-485 : #1 to #15
Data Format	16-bits word 0 … 65536

5.1.1. Baudrate

Only one device can drive the line at a time. The drivers must therefore be put into a high-impedance mode (tri-state) when not in use. Some RS-485 hardware devices handle this automatically. If not, the 485 device software must be used as a control line to handle the driver. (The RTS handshake line will be used typically if your 485 device is controlled through a RS-232 serial port). A consequence of tri-standing the driver is a delay between the end of transmission and the moment when the driver is tri-stated. This turn-around delay is an important part of a two-wire network because during that time no other transmissions can occur.

RS232 notice:

A delay time must be inserted before the next message (T5 = 4ms) is sent. RTS (Line driver) is typically used as control but is not connected in the 485 Network.

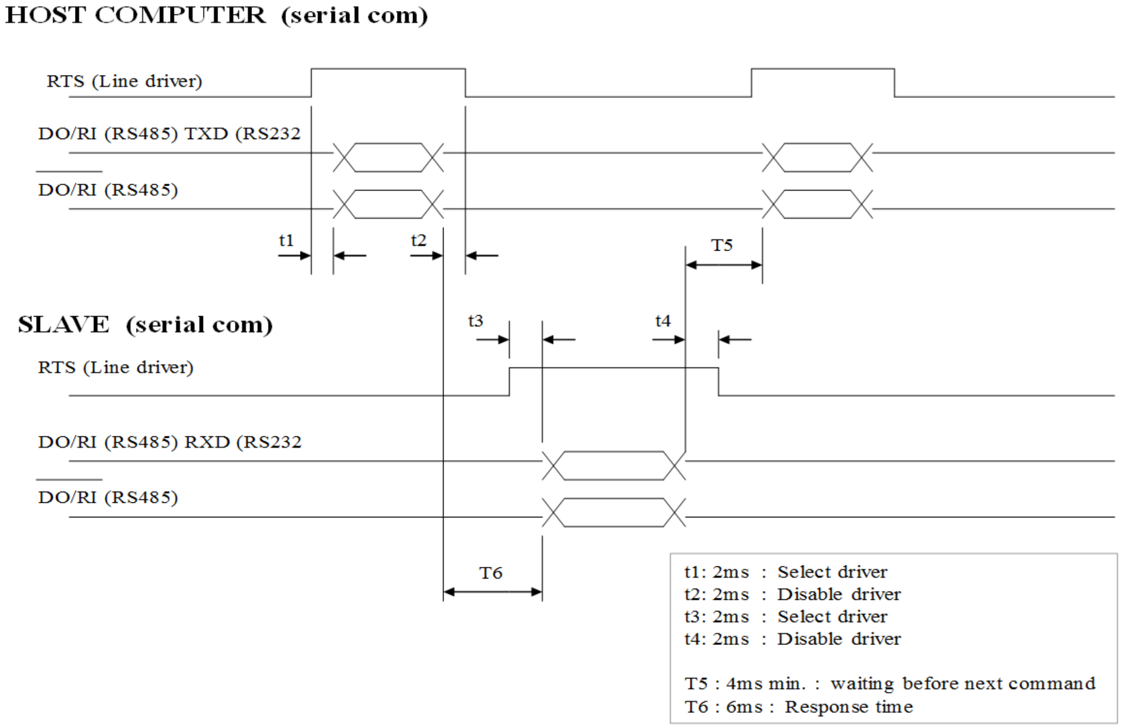


Figure 5-1: timeset RS232/485

5.1.3. Serial Timing

5.1.3.1. AsyCube RS-232 Timing

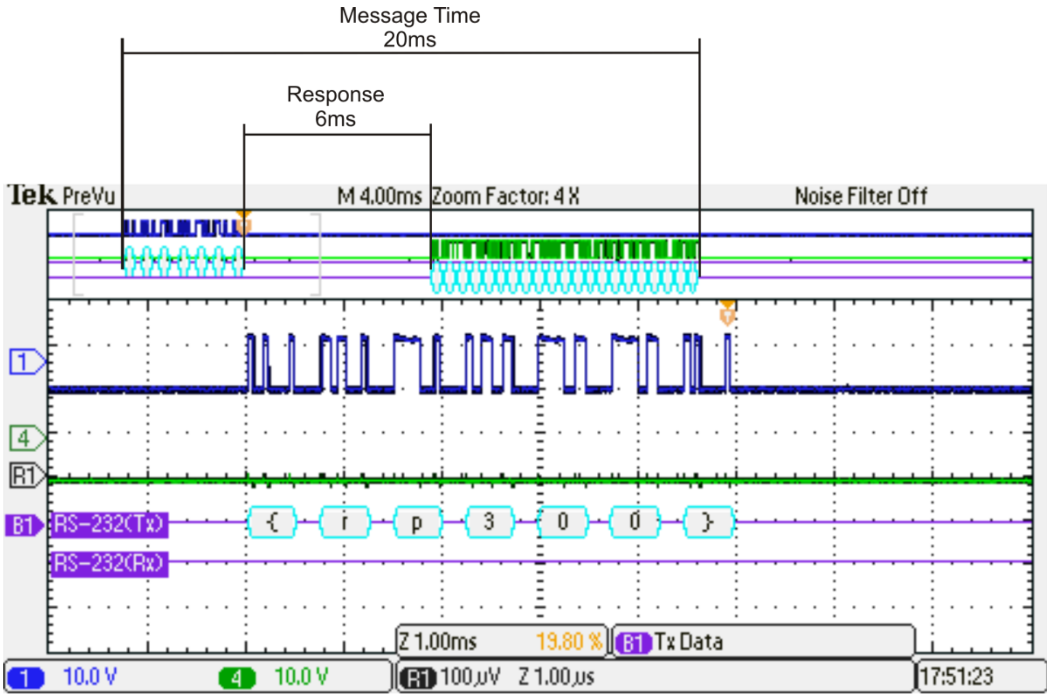


Figure 5-2: ASCII Command {rp300} packet

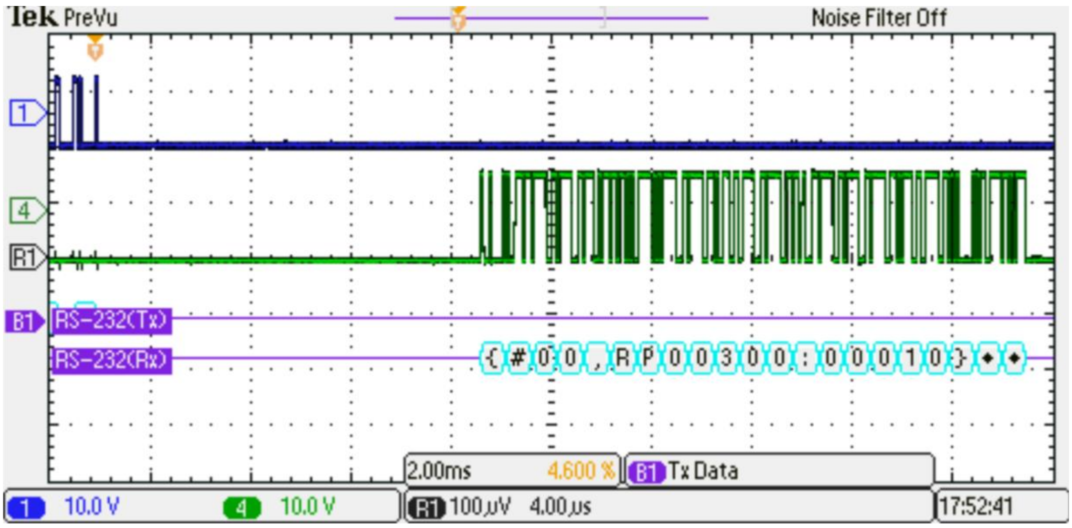


Figure 5-3: ASCII Answer {#00,RP00300:00010}CR LF

5.1.3.2. AsyCube RS-485 Timing

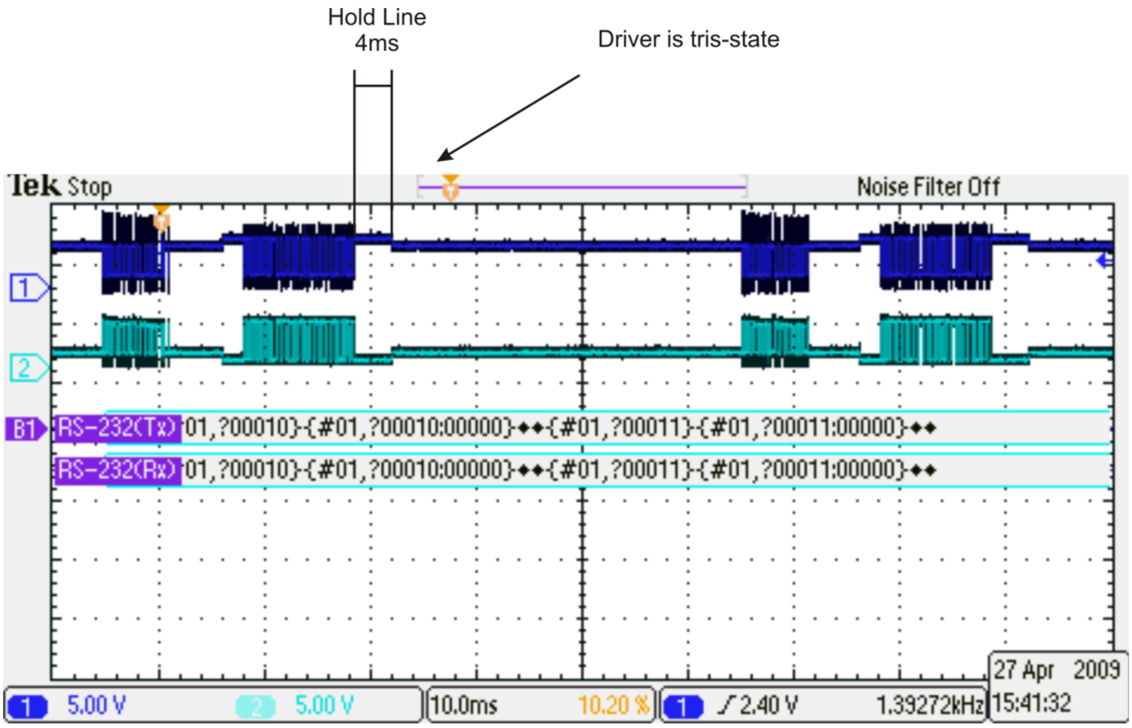


Figure 5-4: Positive signal and negative signal show the ASCII message

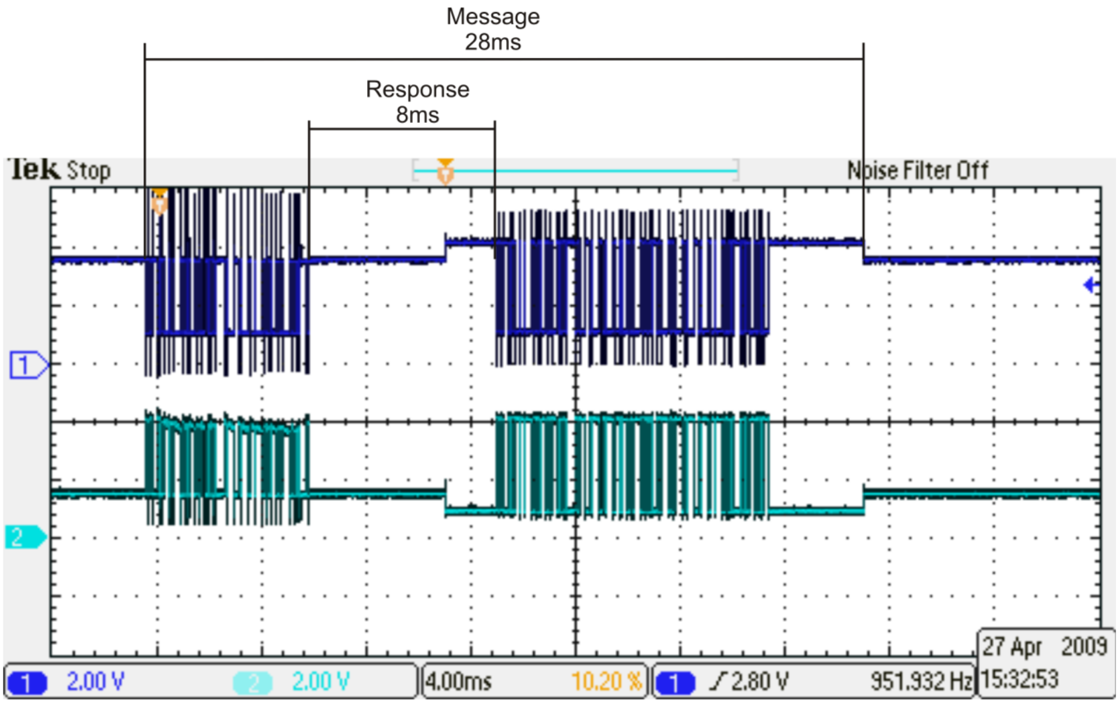


Figure 5-5: The RS-485 response time is 2ms longer than the RS-232

5.1.4. Asycube Serial Network

This technical note provides summarized guidelines and procedures to install and start-up a Asycube RS485 network to communicate at half-duplex.

RS485 gives you two main advantages over RS232: longer cable lengths and greater immunity to noise. In comparison to RS232, RS485 uses lower voltage and differentials signals. RS485 is based on balanced circuits that rely on twisted-pair wires. The same two wires are used to transmit and receive; therefore, within RS485 networks, only one device can transmit while all of the other devices "listen" (receiver mode)

The network topology is a multi-drop bus. In either configuration, devices are addressable (up to 15 nodes) allowing each node to communicate independently. Only one device can drive the line at a time, which implies that the drivers must be set into a high-impedance mode (tri-state) when not in use.

Use shielded twisted pair cables to network the devices. Do not cross positive and negative signals. Reversing the polarity will not damage the device, but it will not communicate. Do not forget the signal ground. The function of the signal ground wire is to tie the signal ground of each device nodes to one common ground.

- End Node: The device attached at both physical ends of the network, containing a network terminator
- In-Line Node: All devices connected to the network, which are not end node.

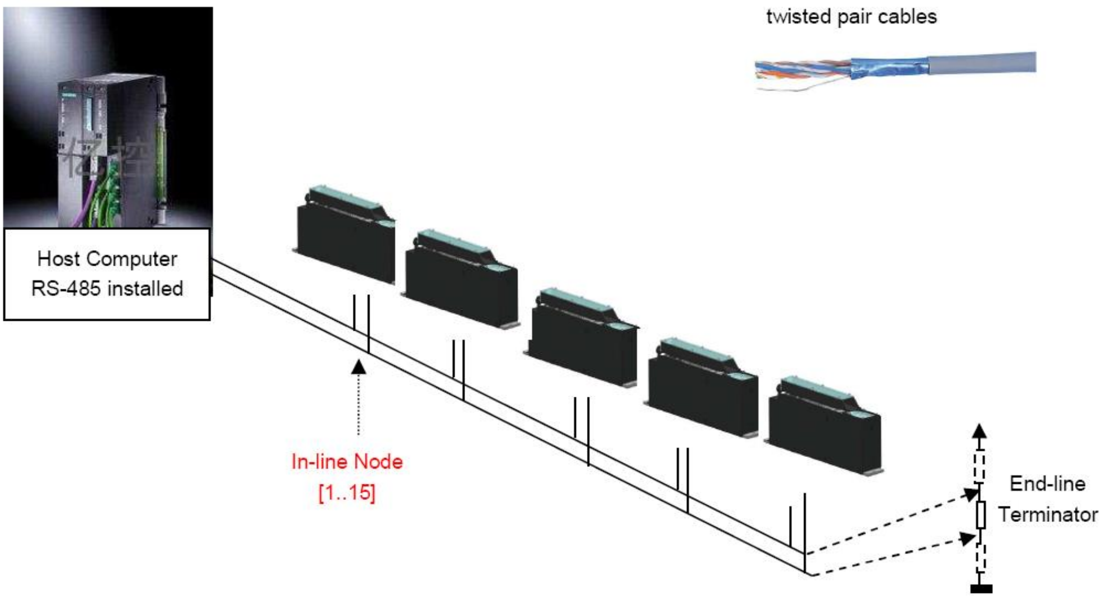


Figure 5-6: "In-Line Node" and "End Node"

5.2. Ethernet via a MOXA converter

In this case, the host computer communicates with the asycube using protocol Ethernet TCP/IP.

Default TCP/IP parameters are:

IP Address	192.168.127.254
Subnet Mask	255.255.255.0
TCP port	4001

These parameters are stored in the MOXA converter and can be modified using the procedure below:

1. Enter IP address in a web browser
2. Enter the password “AsyCube-1690”
3. Click on Operating Settings
4. Click on Port 1
5. Enter your tcp port desired in the Local TCP Port field
6. Validate the configuration using submit button
7. Click on Network Settings
8. Enter Your IP Address in the corresponding field
9. Enter Your Netmask in the corresponding field
10. Validate the configuration with submit button.

5.2.1. Timing

When you use a converter, the Asycube is also using RS-485 protocol to communicate with the converter. Because of that, you have to take care about the response time explained in the AsyCube RS-485 timing of 8ms between two commands (see chapter 5.1.3.2).

5.3. Asycube communication protocol

The host controller communications protocol uses only ASCII characters, and is designed for communication networks (two or more nodes). The host computer is always the master node. Slave nodes transmit only after receiving a message from the master.

Command / Response Format	Command: Begin, Node, Command , Data, End Response: Begin, Node, Response, Data, End + Return Line
Begin	The ASCII char "{" must be the first byte of the packet to allow detection of a new packet.
Node	The ASCII char "#" is the node address prefix. RS232 mode : The node is optional, 0 is the default node. RS485 network mode: The node field will contain two ASCII char from "[1 to 15]". Ethernet with MOXA converter: The node field will contain two ASCII char from "[1 to 15]".
Command	This field will contain two ASCII letter characters followed by the parameter number. These two letters specify the purpose of the message packed (for instance Read or Write Parameter). The value in this field that is sent in response by the AsyCube will specify the command to which the AsyCube is responding. The available commands are listed in Chapter 5.4.
Response	This field contains a fixed format that specifies the validation of the instruction. The AsyCube gives a response message for each corresponding instruction with the node address.
Data	This field contains from 0 to 5 ASCII chars that will be interpreted in various ways, depending on the parameter that appears in the command field.
End	The ASCII char "}" must be the last byte of the packet
+ Return Line	The ASCII char "carriage return 0x0D" and "line feed 0x0A" are the last two bytes returned by the AsyCube to allow the detection of the end of a packet. (Typically using on cursor return line with a remote like "hyper terminal")

" 0 " to " 9 " " a " to " z " " A " to " Z "	not case sensitive
" { "	begin of packet
" # "	node address prefix
" , "	node address separator
" : "	specifies read operation
" = "	specifies write operation
" } "	end of packet
CR	0x0D Carriage Return
LF	0x0A Line Feed

Table 5-1: ASCII Character

Examples:

1) In this example we want to know the actual parameter of register 300 (amplitude of first actuator of Platform Batch A) on the AsyCube with the address #1

   Command: {#1,rp300}

   Response: {#1,rp300:00100}CR LF

2) In this example we want to modify the amplitude of the first actuator (value=90) of the Platform Batch A (301) from the AsyCube with the address #1.

   Command: {#1,wp301=90}

   Response: {#1, wp301=00090} CR LF

5.4. Communication error code (Serial bit)

The serial response gives an error code in the form of an integer value. You have to convert the value to binary to obtain the Error bit affected. For example a response {#01,Er00004} means that the system don't recognized the first character of the command.

Binary	Error Bit	Message
[00001]	0	Message string syntax error!
[00002]	1	String to integer data convert error! , even/off according to read/write
[00004]	2	Unknown first Character of Command!
[00008]	3	Unknown second Character of Command!
[00016]	4	*« not used »*
[00032]	5	*« not used »*
[00064]	6	*« not used »*
[00128]	7	*« not used »*
[00256]	8	Receive buffer is full!
[00512]	9	Receive end of message "}" but receive buffer is full!
[01024]	10	Receive end of message "}" but missing begin of message "{"!
[02048]	11	
[04096]	12	Framing error detected!
[08192]	13	Parity error detected!
[16384]	14	Overflow error detected!
[32768]	15	Receive complete message timeout control!

Table 5-2: communication error code

6. Commands

All existing commands are described in the overview below. The command examples are written for the Asycube with address #1.

6.1. Level access

The system has three different levels to access parameters or to execute some commands. Actual level can be accessed by command {#1,?6}.

Level	Description	Response to query	Select the level
User	User access allows to access to simple commands and parameters to use the Asycube.	{#1,?6:00001}	{#1,WP7=0}
Integrator	Integrator access allows changing some special parameters for advanced configuration.	{#1,?6:00002}	{#1,WP7=1234}
Developer	Developer access allows to change all parameters but is exclusively used by Asyril	{#1,?6:00004}	*Reserved for Asyril*

Table 6-1: level access description

6.2. Access Single Parameters

Code	Label	Command	Response	Remark
WP	Write Parameter	{#1,WP303=90}	{#1,WP303=90}	
RP	Read Parameter	{#1,RP302}	{#1,RP302:90}	

Table 6-2: read and write commands

NOTE :

The even registers are readable parameters and the odd registers are the writable parameters. The register 302 and 303 e.g. stand for the first actuator frequency of the platform batch “A”. If the frequency of the first actuator signal needs to be changed, register 303 needs to be written to, if the frequency of the first actuator signal needs to be asked, register 302 has to be read.

6.3. Access Batch Parameters

6.3.1. Platform Batch Parameters

Code	Label	Command	Response	Remark
SCA..Z	Save Platform Batch A…Z parameters	{#1,SCA=(p1; p2;etc)}	{#1,SCA=(p1;p2;etc)}	
LCA..Z	Load Platform Batch A…Z parameters	{#1,LCA}	{#1,LCA:(p1;p2;etc)}	

Table 6-3: platform batch commands

P1, P2 are parameters given in a specific order and separate with a semicolon. This order is the following:

Amplitude1; Frequency1; Phase1; Waveform1;
Amplitude2; Frequency2; Phase2; Waveform2;
Amplitude3; Frequency3; Waveform3;
Duration

Examples:

- Write platform batch A:
  - Command : {#1,SCA=(90;70;0;1;88;71;90;2;85;72;3;1200)}
  - Response: {#1,SCA=(90;70;0;1;88;71;90;2;85;72;3;1200)}
- Read platform batch A :
  - Command : {#1,LCA}
  - Response: {#1,LCA :(90;70;0;1;88;71;90;2;85;72;3;1200)}

6.3.2. Reservoir Batch Parameters

Code	Label	Command	Response	Remark
SBA..Z	Save Reservoir Batch A…Z parameters	{#1,SBA=(p1; p2;etc)}	{#1,SBA=(p1;p2;etc)}	
LBA..Z	Load Reservoir Batch A…Z parameters	{#1,LBA}	{#1,LBA:(p1;p2;etc)}	

Table 6-4: reservoir batch commands

P1, P2 are parameters given in a specific order and separate with a comma. This order is the following:

Amplitude; Frequency; Waveform; Duration

Examples:

- Write reservoir batch A:
  - Command: {#1,SBA=(80;70;3;1200)}
  - Response: {#1,SBA=(80;70;3;1200)}

- Read reservoir batch A:
  - Command: {#1,LBA}
  - Response: {#1,LBA :(80;70;3;1200)}

NOTE :

To access batch parameters (platform and reservoir), the order of the parameters have to be strictly respected. These commands are useful to access all parameters of a batch in only one message.

6.4. Backlight

Code	Label	Command	Response	Remark
K1	Backlight On	{#1,K1}	{#1,K1}	After response received, the backlight is considered ON, but it depends if raising time of the backlight.
K0	Backlight Off	{#1,K0}	{#1,K0}	
K?	Backlight State	{#1,K?}	{#1,K?:0} {#1,K?:1}	0: Backlight is off 1: Backlight is on
KF	Backlight is flashing	{#1,KF}	{#1,KF}	Duration = Parameter 100

Table 6-5: backlight commands

6.5. System States

The Asycube has to different working states:

- Service mode is the normal mode, to work with the Asycube.
- Standby mode is a special mode. In this mode the power of vibrations, outputs, backlight, etc. are off. Operator can send commands without any physical action on the Asycube. This mode can be useful for tests and debug.

Code	Label	Command	Response	Remark
HC	Halt Platform Vibrations	{#1,HC}	{#1,HC}	Stop all actuators
HB	Halt Outputs Activation	{#1,HB}	{#1,HB}	Stop outputs
H1	System in service	{#1,H1}	{#1,H1}	Set system in service
H0	System in standby	{#1,H0}	{#1,H0}	Set system in standby
H?	System State	{#1,H?}	{#1,H?:0} {#1,H?:1}	0: System in standby 1: System in service

Table 6-6: system states commands

6.6. Platform Vibrations (C for Cube)

Code	Label	Command	Response	Remark
CA..Z	Platform vibrate for a time [ms]	{#1,CF100}	{#1,CF100}	Batch F vibrates for 100ms
CA0..Z0	Platform vibrate forever	{#1,CF0}	{#1,CF0}	Batch F vibrate forever. Stop vibration with command HC.
CA..Z	Platform vibrate for a pre-defined delay	{#1,CF}	{#1,CF}	Delay depends on selected Batch. In this case the delay is equal the register 950 value. If value = 0 -> forever
C?	Read selected Platform batch	{#1,C?}	{#1,C?:F}	Batch F is selected. The selected batch is the last batch executed.
C??	Read selected Platform Batch and state	{#1,C??}	{#1,C??:F3}	Batch F is selected and the state is 3. States: 0: Actuator disabled 1: Actuator enable but stopped 3: Vibrating 5: Actuator stopped over temperature !: Undefined actuator state
CA..Z ?	Read Platform Batch state	{#1,CF?}	{#1,CF:0}	State of Batch F is 0. States: 0: Actuator disabled 1: Actuator enable but stopped 3: Vibrating 5: Actuator stopped over temperature !: Undefined actuator

Table 6-7: platform vibrations commands

6.7. Reservoir activation (B for Bulk)

Code	Label	Command	Response	Remark
BA..Z	Outputs activation for a time [ms]	{#1,BF100}	{#1,BF100}	Batch F vibrates for 100ms
BA0..Z0	Outputs activation forever	{#1,BF0}	{#1,BF0}	Batch F vibrates forever. Use the command HB to stop the vibration.
BA..Z	Outputs activation for a pre-defined delay	{#1,BF}	{#1,BF}	Delay depends on selected Batch. In this case the delay is equal the register 1250 value. If value = 0 -> forever
B?	Read selected Output batch	{#1,B?}	{#1,B?:F}	Batch F is selected. The selected reservoir batch is the last batch executed.
B??	Read selected Output Batch and state	{#1,B??}	{#1,B??:F3}	Batch F is selected and the state is 3. States: 0: Actuator disabled 1: Actuator enabled but stopped 3: Vibrating 5: Actuator stopped over temperature !: Undefined actuator state
BA..Z ?	Read Output Batch state	{#1,BF?}	{#1,BF:0}	State of Batch F is 0. States: 0: Actuator disabled 1: Actuator enabled but stopped 3: Vibrating 5: Actuator stopped over temperature !: Undefined actuator state

Table 6-8: outputs activation commands

6.8. Flash Operation

Code	Label	Command	Response	Remark
DF	Data Flash Memory	{#1,DF}	{#1,DF}	100’000 Program Cycles
DE	Data Erase Memory	{#1,DE}	{#1,DE}	** only developer (Asyril)
DR	Data Restore Memory	{#1,DR}	{#1,DR}	* only integrator
DY	Restore Data with Factory values	{#1,DY}	{#1,DY}	* only integrator
DP	Restore only batches data with Factory values	{#1,DP?}	{#1,DP?:0}	* only integrator
D..?	Data Flash Memory State	{#1,DF?}	{#1,DF?:0}	States: 0: Operation completed 1: Operation in progress 3: Sector erase timeout 4: Page program timeout 5: Flash erase timeout 6: Checksum error 7: Nothing to flash restore 16: Flash operation busy 128: Operation need password

Table 6-9: flash operations commands

6.9. States

Code	Label	Command	Response	Remark
?0	NOP	{#1,?0}	{#1,?0}	
?2	RS485 Node Info	{#1,?2}	{#1,?2:1}	Return node info
?6	Login State	{#1,?6}	{#1,?6}	State: 1: User 2: Integrator 4: Developer (Asyril)
?8	Soft High Version	{#1,?8}	{#1,?8:2}	Return highest value of the software version.
?10	Soft Middle Version	{#1,?10}	{#1,?10:2}	Return middle value of the software version.
?12	Soft Low Version	{#1,?12}	{#1,?12:0}	Return lowest value of the software version.
?40	Backlight Flash Remain Time	{#1,?40}	{#1,?40:00010}	Answer gives the remaining time until the end of the flash of the backlight. The value is in ms.
?42	Platform Remain Time	{#1,?42}	{#1,?42:00010}	Answer gives the remaining time until the end of the platform vibration. The value is in ms
?44	Reservoir Remain Time	{#1,?44}	{#1,?44:00010}	Answer gives the remaining time until the end of the reservoir vibration. The value is in ms

Table 6-10: states commands

6.10. General

Code	Label	Command	Response	Remark
V?	Read Software Version	{#1,V?}	{#1,(c) asycube VX.X.X}	
#	Change the command timeout to 10 sec	{#1,#}	{#1,}	After that the timeout to enter a manual command is delayed to 10 sec. This is used to work with another software (p.e. hyperterminal)

Table 6-11: general commands

6.11. Warnings

To read the warnings, send {#1,rp2} command.

Define	Value	Comment
WARNING_PLATFORM	0x0001	Above max temperature platform
WARNING_RESERVOIR	0x0002	Above max temperature reservoir
WARNING_3	0x0004	Not used
WARNING_4	0x0008	Not used
WARNING_5	0x0010	Not used
WARNING_6	0x0020	Not used
WARNING_LED	0x0040	Not used
WARNING_FLASH	0x0080	Flash operation fail

Table 6-12: warnings list

6.12. Alarms

To read the alarms, send {#1,rp4} command.

Define	Value	Comment
ALARM_1	0x0001	
ALARM_2	0x0002	
ALARM_3	0x0004	
ALARM_4	0x0008	
ALARM_5	0x0010	
ALARM_6	0x0020	
ALARM_BACK_LT	0x0040	Backlight timeout reached
ALARM_8	0x0080	

Table 6-13: alarms list

7. Parameters

7.1. Configuration

*: integrator write; **: developer write

Register	Parameters	Command	Comment
0	NOP	rd/wr	
2	warning	rd & clear	
4	alarm	rd & clear	
6	password	wr	
…			
22	life time [day]	**	if auto-flashing enabled
24	life time [hour]	**	if auto-flashing enabled
26	life time [second]	**	if auto-flashing enabled
28	auto-flashing 20Min.	*	default: enable
30	actuators life time [hour]	**	
32	actuators life time [sec]	**	
34	actuators life time [msec]	**	
36	Actuators number of vibrations [nb] – 0-32767	**	first 2 bytes
38	Actuators number of vibrations [nb] – i*32768	**	bytes 3 and 4
…			
72	Synchro backlight logic	*	0 : logic positive / 1 : logic negative
…			
92	Type 1: AsyCube	**	0 : Unknown 1: Mezzo 2: Forte 3: Fortissimo 4: Largo_A5
94	Type 2: AsyCube	**	
96	Color of Backlight	**	0: Green 1: Red 2: Blue 3: IR 4: UV 5: White 99: None
…			
100	Backlight Flash Time		[ms]
102	Backlight PWM		[%]
…			

122	Platform integral limit time	**	Default 60 [sec]
124	Reservoir integral limit time	**	Default 60 [sec]
…			
142	UART1 driver RS485	**	Default true
144	UART1 rx timeout RS485	*	Default: 1'000 [ms]
146	UART2 rx timeout RS232		Default: 1'000 [ms]
148	UART1 tx timeout RS485	*	Default: 2 [ms]
150	UART2 tx timeout RS232	*	Default: 0 [ms]
152	Backlight timeout	**	0= disable timeout function 30 = 30 sec with PWM 100%, 60 sec with PWM 50%, etc.
…			
158	Gain amplitude actuator 1	*	(int) [2.55]
160	Gain amplitude actuator 2	*	(int) [2.55]
162	Gain amplitude actuator 3	*	(int) [2.55]
164	Gain amplitude actuator 4	*	(int) [2.55]
166	Offset amplitude actuator 1	*	(int) +/- n 1/256
168	Offset amplitude actuator 2	*	(int) +/- n 1/256
170	Offset amplitude actuator 3	*	(int) +/- n 1/256
172	Offset amplitude actuator 4	*	(int) +/- n 1/256
174	Offset frequency actuator 1	*	(int) +/- n 0.25 Hz
176	Offset frequency actuator 2	*	(int) +/- n 0.25 Hz
178	Offset frequency actuator 3	*	(int) +/- n 0.25 Hz
180	Offset frequency actuator 4	*	(int) +/- n 0.25 Hz

Table 7-1: configuration parameters

7.2. Platform Batches

Register	Define	Batch	Units	Range
300	Amplitude 1	Platform "A"	[%]	0 to 100 %
302	Frequency 1		[Hz]	0 to 350 Hz
304	Phase 1		[°deg.]	0 to 360 degrees
306	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
308	Amplitude 2		[%]	0 to 100 %
310	Frequency 2		[Hz]	0 to 350 Hz
312	Phase 2		[°deg.]	0 to 360 degrees
314	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
316	Amplitude 3		[%]	0 to 100 %
318	Frequency 3		[Hz]	0 to 350 Hz
320	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
322	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
940	Duration		[ms]	
324	Amplitude 1	Platform "B"	[%]	0 to 100 %
326	Frequency 1		[Hz]	0 to 350 Hz
328	Phase 1		[°deg.]	0 to 360 degrees
330	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
332	Amplitude 2		[%]	0 to 100 %
334	Frequency 2		[Hz]	0 to 350 Hz
336	Phase 2		[°deg.]	0 to 360 degrees
338	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
340	Amplitude 3		[%]	0 to 100 %
342	Frequency 3		[Hz]	0 to 350 Hz
344	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
346	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
942	Duration		[ms]	
348	Amplitude 1	Platform "C"	[%]	0 to 100 %
350	Frequency 1		[Hz]	0 to 350 Hz
352	Phase 1		[°deg.]	0 to 360 degrees
354	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
356	Amplitude 2		[%]	0 to 100 %
358	Frequency 2		[Hz]	0 to 350 Hz
360	Phase 2		[°deg.]	0 to 360 degrees
362	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn

Register	Define	Batch	Units	Range
364	Amplitude 3		[%]	0 to 100 %
366	Frequency 3		[Hz]	0 to 350 Hz
368	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
370	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
944	Duration		[ms]	
372	Amplitude 1	Platform "D"	[%]	0 to 100 %
374	Frequency 1		[Hz]	0 to 350 Hz
376	Phase 1		[°deg.]	0 to 360 degrees
378	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
380	Amplitude 2		[%]	0 to 100 %
382	Frequency 2		[Hz]	0 to 350 Hz
384	Phase 2		[°deg.]	0 to 360 degrees
386	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
388	Amplitude 3		[%]	0 to 100 %
390	Frequency 3		[Hz]	0 to 350 Hz
392	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
394	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
946	Duration		[ms]	
396	Amplitude 1	Platform "E"	[%]	0 to 100 %
398	Frequency 1		[Hz]	0 to 350 Hz
400	Phase 1		[°deg.]	0 to 360 degrees
402	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
404	Amplitude 2		[%]	0 to 100 %
406	Frequency 2		[Hz]	0 to 350 Hz
408	Phase 2		[°deg.]	0 to 360 degrees
410	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
412	Amplitude 3		[%]	0 to 100 %
414	Frequency 3		[Hz]	0 to 350 Hz
416	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
418	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
948	Duration		[ms]	
420	Amplitude 1	Platform "F"	[%]	0 to 100 %
422	Frequency 1		[Hz]	0 to 350 Hz
424	Phase 1		[°deg.]	0 to 360 degrees

Register	Define	Batch	Units	Range
426	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
428	Amplitude 2		[%]	0 to 100 %
430	Frequency 2		[Hz]	0 to 350 Hz
432	Phase 2		[°deg.]	0 to 360 degrees
434	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
436	Amplitude 3		[%]	0 to 100 %
438	Frequency 3		[Hz]	0 to 350 Hz
440	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
442	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
950	Duration		[ms]	
444	Amplitude 1	Platform "G"	[%]	0 to 100 %
446	Frequency 1		[Hz]	0 to 350 Hz
448	Phase 1		[°deg.]	0 to 360 degrees
450	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
452	Amplitude 2		[%]	0 to 100 %
454	Frequency 2		[Hz]	0 to 350 Hz
456	Phase 2		[°deg.]	0 to 360 degrees
458	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
460	Amplitude 3		[%]	0 to 100 %
462	Frequency 3		[Hz]	0 to 350 Hz
464	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
466	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
952	Duration		[ms]	
468	Amplitude 1	Platform "H"	[%]	0 to 100 %
470	Frequency 1		[Hz]	0 to 350 Hz
472	Phase 1		[°deg.]	0 to 360 degrees
474	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
476	Amplitude 2		[%]	0 to 100 %
478	Frequency 2		[Hz]	0 to 350 Hz
480	Phase 2		[°deg.]	0 to 360 degrees
482	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
484	Amplitude 3		[%]	0 to 100 %
486	Frequency 3		[Hz]	0 to 350 Hz
488	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
490	Mode		[0..2]	0=normal, 1=evacuate, 2=fill

Register	Define	Batch	Units	Range
954	Duration		[ms]	
492	Amplitude 1	Platform "I"	[%]	0 to 100 %
494	Frequency 1		[Hz]	0 to 350 Hz
496	Phase 1		[°deg.]	0 to 360 degrees
498	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
500	Amplitude 2		[%]	0 to 100 %
502	Frequency 2		[Hz]	0 to 350 Hz
504	Phase 2		[°deg.]	0 to 360 degrees
506	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
508	Amplitude 3		[%]	0 to 100 %
510	Frequency 3		[Hz]	0 to 350 Hz
512	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
514	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
956	Duration		[ms]	
516	Amplitude 1	Platform "J"	[%]	0 to 100 %
518	Frequency 1		[Hz]	0 to 350 Hz
520	Phase 1		[°deg.]	0 to 360 degrees
522	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
524	Amplitude 2		[%]	0 to 100 %
526	Frequency 2		[Hz]	0 to 350 Hz
528	Phase 2		[°deg.]	0 to 360 degrees
530	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
532	Amplitude 3		[%]	0 to 100 %
534	Frequency 3		[Hz]	0 to 350 Hz
536	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
538	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
958	Duration		[ms]	
540	Amplitude 1	Platform "K"	[%]	0 to 100 %
542	Frequency 1		[Hz]	0 to 350 Hz
544	Phase 1		[°deg.]	0 to 360 degrees
546	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
548	Amplitude 2		[%]	0 to 100 %
550	Frequency 2		[Hz]	0 to 350 Hz
552	Phase 2		[°deg.]	0 to 360 degrees

Register	Define	Batch	Units	Range
554	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
556	Amplitude 3		[%]	0 to 100 %
558	Frequency 3		[Hz]	0 to 350 Hz
560	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
562	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
960	Duration		[ms]	
564	Amplitude 1	Platform "L"	[%]	0 to 100 %
566	Frequency 1		[Hz]	0 to 350 Hz
568	Phase 1		[°deg.]	0 to 360 degrees
570	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
572	Amplitude 2		[%]	0 to 100 %
574	Frequency 2		[Hz]	0 to 350 Hz
576	Phase 2		[°deg.]	0 to 360 degrees
578	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
580	Amplitude 3		[%]	0 to 100 %
582	Frequency 3		[Hz]	0 to 350 Hz
584	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
586	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
962	Duration		[ms]	
588	Amplitude 1	Platform "M"	[%]	0 to 100 %
590	Frequency 1		[Hz]	0 to 350 Hz
592	Phase 1		[°deg.]	0 to 360 degrees
594	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
596	Amplitude 2		[%]	0 to 100 %
598	Frequency 2		[Hz]	0 to 350 Hz
600	Phase 2		[°deg.]	0 to 360 degrees
602	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
604	Amplitude 3		[%]	0 to 100 %
606	Frequency 3		[Hz]	0 to 350 Hz
608	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
610	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
964	Duration		[ms]	
612	Amplitude 1	Platform "N"	[%]	0 to 100 %
614	Frequency 1		[Hz]	0 to 350 Hz

Register	Define	Batch	Units	Range
616	Phase 1		[°deg.]	0 to 360 degrees
618	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
620	Amplitude 2		[%]	0 to 100 %
622	Frequency 2		[Hz]	0 to 350 Hz
624	Phase 2		[°deg.]	0 to 360 degrees
626	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
628	Amplitude 3		[%]	0 to 100 %
630	Frequency 3		[Hz]	0 to 350 Hz
632	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
634	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
966	Duration		[ms]	
636	Amplitude 1	Platform "O"	[%]	0 to 100 %
638	Frequency 1		[Hz]	0 to 350 Hz
640	Phase 1		[°deg.]	0 to 360 degrees
642	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
644	Amplitude 2		[%]	0 to 100 %
646	Frequency 2		[Hz]	0 to 350 Hz
648	Phase 2		[°deg.]	0 to 360 degrees
650	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
652	Amplitude 3		[%]	0 to 100 %
654	Frequency 3		[Hz]	0 to 350 Hz
656	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
658	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
968	Duration		[ms]	
660	Amplitude 1	Platform "P"	[%]	0 to 100 %
662	Frequency 1		[Hz]	0 to 350 Hz
664	Phase 1		[°deg.]	0 to 360 degrees
666	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
668	Amplitude 2		[%]	0 to 100 %
670	Frequency 2		[Hz]	0 to 350 Hz
672	Phase 2		[°deg.]	0 to 360 degrees
674	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
676	Amplitude 3		[%]	0 to 100 %
678	Frequency 3		[Hz]	0 to 350 Hz
680	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn

Register	Define	Batch	Units	Range
682	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
970	Duration		[ms]	
684	Amplitude 1	Platform "Q"	[%]	0 to 100 %
686	Frequency 1		[Hz]	0 to 350 Hz
688	Phase 1		[°deg.]	0 to 360 degrees
690	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
692	Amplitude 2		[%]	0 to 100 %
694	Frequency 2		[Hz]	0 to 350 Hz
696	Phase 2		[°deg.]	0 to 360 degrees
698	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
700	Amplitude 3		[%]	0 to 100 %
702	Frequency 3		[Hz]	0 to 350 Hz
704	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
706	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
972	Duration		[ms]	
708	Amplitude 1	Platform "R"	[%]	0 to 100 %
710	Frequency 1		[Hz]	0 to 350 Hz
712	Phase 1		[°deg.]	0 to 360 degrees
714	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
716	Amplitude 2		[%]	0 to 100 %
718	Frequency 2		[Hz]	0 to 350 Hz
720	Phase 2		[°deg.]	0 to 360 degrees
722	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
724	Amplitude 3		[%]	0 to 100 %
726	Frequency 3		[Hz]	0 to 350 Hz
728	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
730	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
974	Duration		[ms]	
732	Amplitude 1	Platform "S"	[%]	0 to 100 %
734	Frequency 1		[Hz]	0 to 350 Hz
736	Phase 1		[°deg.]	0 to 360 degrees
738	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
740	Amplitude 2		[%]	0 to 100 %
742	Frequency 2		[Hz]	0 to 350 Hz

Register	Define	Batch	Units	Range
744	Phase 2		[°deg.]	0 to 360 degrees
746	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
748	Amplitude 3		[%]	0 to 100 %
750	Frequency 3		[Hz]	0 to 350 Hz
752	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
754	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
976	Duration		[ms]	
756	Amplitude 1	Platform "T"	[%]	0 to 100 %
758	Frequency 1		[Hz]	0 to 350 Hz
760	Phase 1		[°deg.]	0 to 360 degrees
762	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
764	Amplitude 2		[%]	0 to 100 %
766	Frequency 2		[Hz]	0 to 350 Hz
768	Phase 2		[°deg.]	0 to 360 degrees
770	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
772	Amplitude 3		[%]	0 to 100 %
774	Frequency 3		[Hz]	0 to 350 Hz
776	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
778	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
978	Duration		[ms]	
780	Amplitude 1	Platform "U"	[%]	0 to 100 %
782	Frequency 1		[Hz]	0 to 350 Hz
784	Phase 1		[°deg.]	0 to 360 degrees
786	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
788	Amplitude 2		[%]	0 to 100 %
790	Frequency 2		[Hz]	0 to 350 Hz
792	Phase 2		[°deg.]	0 to 360 degrees
794	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
796	Amplitude 3		[%]	0 to 100 %
798	Frequency 3		[Hz]	0 to 350 Hz
800	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
802	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
980	Duration		[ms]	
804	Amplitude 1	Platform "V"	[%]	0 to 100 %

Register	Define	Batch	Units	Range
806	Frequency 1		[Hz]	0 to 350 Hz
808	Phase 1		[°deg.]	0 to 360 degrees
810	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
812	Amplitude 2		[%]	0 to 100 %
814	Frequency 2		[Hz]	0 to 350 Hz
816	Phase 2		[°deg.]	0 to 360 degrees
818	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
820	Amplitude 3		[%]	0 to 100 %
822	Frequency 3		[Hz]	0 to 350 Hz
824	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
826	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
982	Duration		[ms]	
828	Amplitude 1	Platform "W"	[%]	0 to 100 %
830	Frequency 1		[Hz]	0 to 350 Hz
832	Phase 1		[°deg.]	0 to 360 degrees
834	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
836	Amplitude 2		[%]	0 to 100 %
838	Frequency 2		[Hz]	0 to 350 Hz
840	Phase 2		[°deg.]	0 to 360 degrees
842	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
844	Amplitude 3		[%]	0 to 100 %
846	Frequency 3		[Hz]	0 to 350 Hz
848	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
850	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
984	Duration		[ms]	
852	Amplitude 1	Platform "X"	[%]	0 to 100 %
854	Frequency 1		[Hz]	0 to 350 Hz
856	Phase 1		[°deg.]	0 to 360 degrees
858	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
860	Amplitude 2		[%]	0 to 100 %
862	Frequency 2		[Hz]	0 to 350 Hz
864	Phase 2		[°deg.]	0 to 360 degrees
866	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
868	Amplitude 3		[%]	0 to 100 %
870	Frequency 3		[Hz]	0 to 350 Hz

Register	Define	Batch	Units	Range
872	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
874	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
986	Duration		[ms]	
876	Amplitude 1	Platform "Y"	[%]	0 to 100 %
878	Frequency 1		[Hz]	0 to 350 Hz
880	Phase 1		[°deg.]	0 to 360 degrees
882	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
884	Amplitude 2		[%]	0 to 100 %
886	Frequency 2		[Hz]	0 to 350 Hz
888	Phase 2		[°deg.]	0 to 360 degrees
890	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
892	Amplitude 3		[%]	0 to 100 %
894	Frequency 3		[Hz]	0 to 350 Hz
896	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
898	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
988	Duration		[ms]	
900	Amplitude 1	Platform "Z"	[%]	0 to 100 %
902	Frequency 1		[Hz]	0 to 350 Hz
904	Phase 1		[°deg.]	0 to 360 degrees
906	Waveform 1		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
908	Amplitude 2		[%]	0 to 100 %
910	Frequency 2		[Hz]	0 to 350 Hz
912	Phase 2		[°deg.]	0 to 360 degrees
914	Waveform 2		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
916	Amplitude 3		[%]	0 to 100 %
918	Frequency 3		[Hz]	0 to 350 Hz
920	Waveform 3		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
922	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
990	Duration		[ms]	

Table 7-2: platform batches parameters

7.3. Reservoir Batches

Register	Define	Batch	Units	Range
1000	Amplitude	Bulk "A"	[%]	0 to 100%
1002	Frequency		[Hz]	0 to 350 Hz
1004	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1006	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1240	Delay		[ms]	
1008	Amplitude	Bulk "B"	[%]	0 to 100%
1010	Frequency		[Hz]	0 to 350 Hz
1012	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1014	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1242	Delay		[ms]	
1016	Amplitude	Bulk "C"	[%]	0 to 100%
1018	Frequency		[Hz]	0 to 350 Hz
1020	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1022	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1244	Delay		[ms]	
1024	Amplitude	Bulk "D"	[%]	0 to 100%
1026	Frequency		[Hz]	0 to 350 Hz
1028	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1030	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1246	Delay		[ms]	
1032	Amplitude	Bulk "E"	[%]	0 to 100%
1034	Frequency		[Hz]	0 to 350 Hz
1036	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1038	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1248	Delay		[ms]	
1040	Amplitude	Bulk "F"	[%]	0 to 100%
1042	Frequency		[Hz]	0 to 350 Hz
1044	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1046	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1250	Delay		[ms]	

Register	Define	Batch	Units	Range
1048	Amplitude	Bulk "G"	[%]	0 to 100%
1050	Frequency		[Hz]	0 to 350 Hz
1052	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1054	Mode	Bulk "G"	[0..2]	0=normal, 1=evacuate, 2=fill
1252	Delay		[ms]	
1056	Amplitude	Bulk "H"	[%]	0 to 100%
1058	Frequency		[Hz]	0 to 350 Hz
1060	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1062	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1254	Delay		[ms]	
1064	Amplitude	Bulk "I"	[%]	0 to 100%
1066	Frequency		[Hz]	0 to 350 Hz
1068	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1070	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1256	Delay		[ms]	
1072	Amplitude	Bulk "J"	[%]	0 to 100%
1074	Frequency		[Hz]	0 to 350 Hz
1076	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1078	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1258	Delay		[ms]	
1080	Amplitude	Bulk "K"	[%]	0 to 100%
1082	Frequency		[Hz]	0 to 350 Hz
1084	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1086	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1260	Delay		[ms]	
1088	Amplitude	Bulk "L"	[%]	0 to 100%
1090	Frequency		[Hz]	0 to 350 Hz
1092	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1094	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1262	Delay		[ms]	
1096	Amplitude	Bulk "M"	[%]	0 to 100%

Register	Define	Batch	Units	Range
1098	Frequency		[Hz]	0 to 350 Hz
1100	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1102	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1264	Delay		[ms]	
1104	Amplitude	Bulk "N"	[%]	0 to 100%
1106	Frequency		[Hz]	0 to 350 Hz
1108	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1110	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1266	Delay		[ms]	
1112	Amplitude	Bulk "O"	[%]	0 to 100%
1114	Frequency		[Hz]	0 to 350 Hz
1116	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1118	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1268	Delay		[ms]	
1120	Amplitude	Bulk "P"	[%]	0 to 100%
1122	Frequency		[Hz]	0 to 350 Hz
1124	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1126	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1270	Delay		[ms]	
1128	Amplitude	Bulk "Q"	[%]	0 to 100%
1130	Frequency		[Hz]	0 to 350 Hz
1132	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1134	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1272	Delay		[ms]	
1136	Amplitude	Bulk "R"	[%]	0 to 100%
1138	Frequency		[Hz]	0 to 350 Hz
1140	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1142	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1274	Delay		[ms]	
1144	Amplitude	Bulk "S"	[%]	0 to 100%
1146	Frequency		[Hz]	0 to 350 Hz

Register	Define	Batch	Units	Range
1148	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1150	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1276	Delay		[ms]	
1152	Amplitude	Bulk "T"	[%]	0 to 100%
1154	Frequency		[Hz]	0 to 350 Hz
1156	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1158	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1278	Delay		[ms]	
1160	Amplitude	Bulk "U"	[%]	0 to 100%
1162	Frequency		[Hz]	0 to 350 Hz
1164	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1166	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1280	Delay		[ms]	
1168	Amplitude	Bulk "V"	[%]	0 to 100%
1170	Frequency		[Hz]	0 to 350 Hz
1172	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1174	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1282	Delay		[ms]	
1176	Amplitude	Bulk "W"	[%]	0 to 100%
1178	Frequency		[Hz]	0 to 350 Hz
1180	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1182	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1284	Delay		[ms]	
1184	Amplitude	Bulk "X"	[%]	0 to 100%
1186	Frequency		[Hz]	0 to 350 Hz
1188	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1190	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1286	Delay		[ms]	
1192	Amplitude	Bulk "Y"	[%]	0 to 100%
1194	Frequency		[Hz]	0 to 350 Hz
1196	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn

Register	Define	Batch	Units	Range
1198	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1288	Delay		[ms]	
1200	Amplitude	Bulk "Z"	[%]	0 to 100%
1202	Frequency		[Hz]	0 to 350 Hz
1204	Figure		[0..3]	0=no signal, 1=sinus, 2=rp up, 3= rp dn
1206	Mode		[0..2]	0=normal, 1=evacuate, 2=fill
1290	Delay		[ms]	

Table 7-3: reservoir batches parameters

8. Technical support

8.1. For better service …

You have read the the related manuals without finding answers to your questions? Before calling the support service, note the following information for your system:

- serial number and product key of your material
- software version
- alarm or error message displayed on the screen

8.2. Contact

You can find lot of information on our website: www.asyril.ch

You can also contact us by mail or call our support service:

support@asyril.ch +41 26 653 7190

Review history

Rev.	Date	Author	Comments
1.0	09.12.2011	SiA	Initial Version
2.0	14.11.2014	DaM	Initial version with new template and asycube library section
2.1	01.06.2015	HsJ	Update for new firmware version 1.5.0

asyril sa
z.i. le vivier
ch-1690 villaz-st-pierre
switzerland
tel. +41 26 653 71 90
fax +41 26 653 71 91
info@asyril.ch
www.asyril.ch