

Commissioning Manual

AGILOX ONE



Obsah

Commissioning Manual AGILOX ONE	1
Revision History	2
Version 2021Q1	2
1. General	3
1.1 Introduction	3
1.2 Commissioning Manual	3
1.3 Copyright	3
1.4 Target Group	3
1.5 Typographic Conventions	4
1.6 Description of Warnings and Safety Information	4
2. Safety	5
3. Description	6
3.1 AGILOX System Overview	6
4. Commissioning	7
4.1 Maintenance Operation Mode	8
4.1.1 Description	8
4.1.2 Activation	9
4.2 Init Setup	10
4.2.1 Union	10
4.2.2 WLAN Network	11
4.2.3 IP - Address	11
4.3 Visualization	12
4.3.1 Context Menus	12
4.3.2 Stations	14
4.3.3 Vehicle Information	21
4.3.4 Visualization Toggles	22
4.3.5 Manual Operation	24
4.3.6 Reports	25
4.3.7 Misc. Tools and Settings	30
4.4 Teach Contours	52
4.4.1 Initial Mapping and Acquisition	52
4.5 Single Teach	56
4.5 Map	57
4.5.1 Initial Definition of Driving Paths	59
4.5.2 Map layers	61
4.5.3 Color Schema	61
4.5.4 SVG Elements	62
4.5.5 Action Settings	65
4.6 Collections	70
4.6.1 agioloxio	70
4.6.2 barrier	70
4.6.3 collective	74
4.6.4 floor	74
4.6.5 heartbeat	75
4.6.6 loadcarrier	76
4.6.7 mapping	77
4.6.8 supply_dispose	78

4.6.9 toggle	83
4.7 Workflow Definition Language	84
4.7.1 Workflow Variables	84
4.7.2 Workflow Object	86
4.7.3 Workflow Order Object	87
4.7.4 Workflow Event Object	90
4.7.5 Workflow Condition Object	94
4.7.6 Workflow Action Object	99
4.7.7 Workflow Target Object	104
4.7.8 Webservice Request	106
4.7.9 AGILOXIO Request Definition	107
4.7.10 Barcode Object	109
4.7.11 Special Workflows	111
4.8 JSON API	112
4.8.1 Basic usage	112
4.8.2 Most Common Application	113
4.8.3 General GET endpoints	115
4.8.4 System overview	117
4.8.5 Order Pool	126
4.8.6 POST Orders - Create/Edit orders	131
4.8.7 POST Occupation - Change occupation of stations	135
4.8.8 Post AGILOX vehicle	135
4.8.9 Mapping	137
5. User Guide	138
5.1 Create Special Stationareas	139
5.1.1 Narrow Aisle	139
5.1.2 Block Storage	139
5.2 Control High-Speed Door	141
5.2.1 Electrical Signals	141
5.2.2 Configuration AGILOX	141
5.3 Crossing with Entry Control	143
5.3.1 Electrical Signals	143
5.3.2 Configuration AGILOX	143
5.4 Conveyor station	145
5.4.1 Electrical Signals	145
5.4.2 Configuration AGILOX	145
5.5 Pallet Dispensers	148
5.5.1 Electrical Signals	148
5.6 Lift System	149
5.6.1 Mechanical Requirements	149
5.6.2 Electrical Requirements	150
5.6.3 Signal Exchange	150
5.6.4 Electrical Signals	150
5.6.5 Configuration AGILOX	150
5.7 Workflow	151
5.7.1 Workflow STATION -> STATION	151
5.7.2 Workflow STATION -> AREA	151
5.7.3 Workflow with description	152
5.7.4 Workflow with weight check	154
5.7.5 Workflow Loop between two Stations	155

5.8 Workflow Drop at Conveyor Station	156
5.9 Workflow with Schedule	159
5.10 Workflow with Barcode	160
5.10.1 Basic Usage	160
5.10.2 Request mit Barcode	161
5.11 Workflow with AGILOX JSON-API	162
5.11.1 Workflow with Target from Host System	162
5.11.2 Workflow with Event Based Webservice POST	166
5.12 Workflow with supply/dispose	167
5.12.1 Dispose Workflow with target determination via Barcode	167
5.13 Dashboard Overview	169
5.14 Workflow Dispose Area	171
ANNEX - AGILOX Feature Feed	174
Commissioning Manual Version 2021Q1	174
Archive	175

Translation of Original Commissioning Manual AGILOX ONE

AGILOX North America, Inc.

Management:

Dipl.-Ing. (FH) Franz Humer, M.A.
Dirk Erlacher, M.A.

110 Samaritan Drive, Suite 206
Cumming, GA 30040
United States of America

Mail: office@agilox.net

Technical Support: support@agilox.net

Locations Worldwide:

<https://www.agilox.net/en/unternehmen#sales-contact>

Revision History

Version 2021Q1

- Updated Target Group [1.4 - Target Group](#)
- Additions to [5.5 - Pallet Dispenser](#)
- Detailed description of feedback structure in [4.8.4 - System Overview](#)
- Changes according to
[Feature Feed](#)

1. General

1.1 Introduction

The AGILOX system has been designed and manufactured in line with state-of-the-art standards. To ensure its safe operation, all persons using it for their work must be properly trained and instructed. In addition to this requirement, please read this Operation Manual carefully and in its entirety.



INFORMATION

As our system is continually evolving, its manufacturer reserves the right to modify the illustrations contained in this documentation.

1.2 Commissioning Manual

This Commissioning Manual is based on the [Operation Manual](#) and contains extensive information for commissioning and administration. However it does not replace the Operation Manual.

The Operation Manual contains all relevant information for users and their staff regarding the operation of the AGILOX system as well as its installation and maintenance. Additionally, it contains important safety information and notes. It must be read, understood and obeyed in its entirety by all responsible persons and forms the basis of this Commissioning Manual.

1.3 Copyright

AGILOX North America, Inc. remains the sole holder of the copyright for this Operation Manual. For any unauthorized use of this Manual for competition-related purposes in whole or in part, the express approval of the manufacturer is required. Furthermore, these instructions must not be made available to any third parties.

1.4 Target Group

This commissioning manual is aimed at specifically trained system administrators.

Administrators are persons who operate and optimize the system. They can expand their own system with the existing range of functionalities and make changes to the operating areas.

Administrator requirements:

- Minimum age: 18 years
- Must have a basic technical and logistical understanding
- Must have read and understood both the operation manual and the commissioning manual
- Must have been trained on the AGILOX system by AGILOX North America, Inc., an AGILOX system partner or an in-house responsible person
- This training has to follow the training guidelines of and must be documented in the "Training Protocol Administrator".

1.5 Typographic Conventions

Standard formats:

bold	important text, important passages
CAPITAL LETTERS	operating states
Courier	programming codes and examples

List formats:

1.	step-by-step instructions with a fixed sequence
2.	
•	lists without any fixed sequence
→	consequence or result of an instruction

1.6 Description of Warnings and Safety Information



DANGER

Non-compliance will lead to severe irreversible injuries or death.
Colorcode: #ff0000



WARNING

Non-compliance may lead to severe irreversible or fatal injuries.
Colorcode: #f5860b



CAUTION

Non-compliance may lead to minor or mid-level injuries.
Colorcode: #f6e807



NOTICE

Non-compliance may lead to damage to property.
Colorcode: #00b0f0



INFORMATION

Important information
Colorcode: #33cc33

2. Safety



CAUTION

The Operation Manual contains all relevant safety information for users and their staff regarding the operation of the AGILOX system as well as its installation and maintenance.

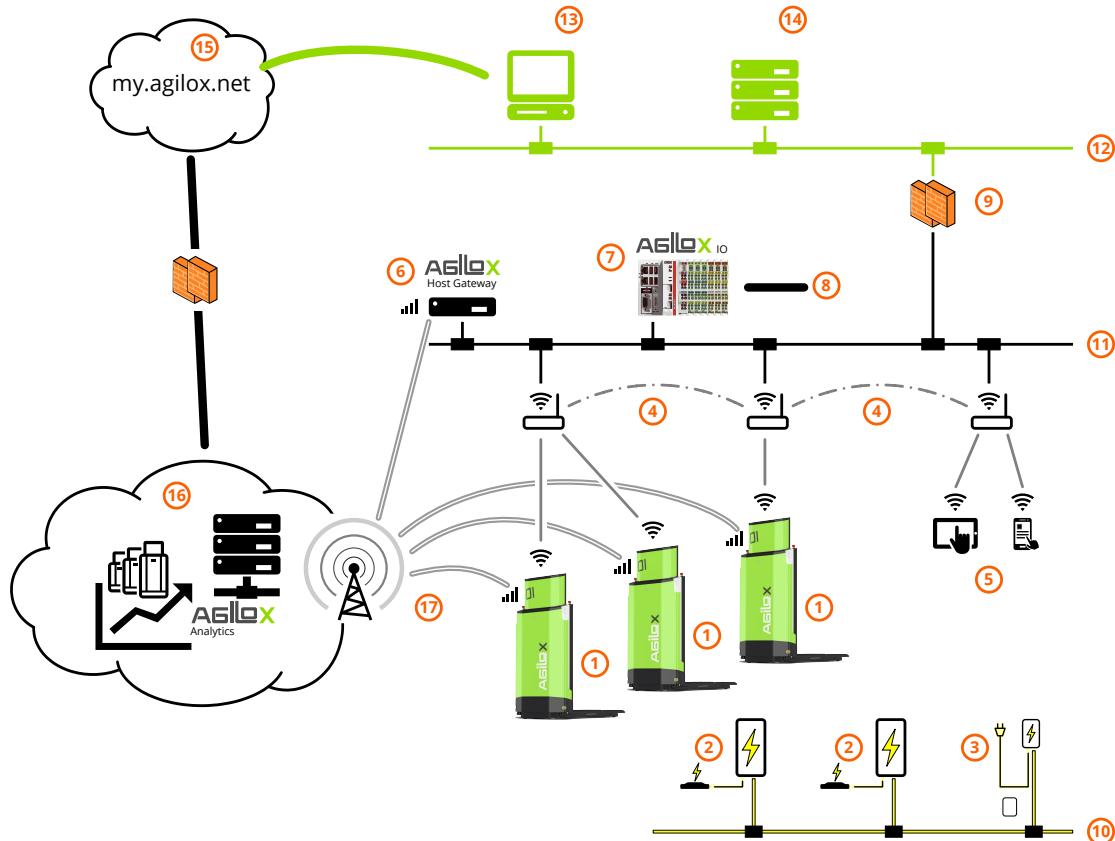
The instructions and requirements documented in chapter [2 - Safety](#) of the Operation Manual must be read, understood and complied with over the entire service life of the system.

All operators are required to study them even before commissioning their system.

3. Description

3.1 AGILOX System Overview

The AGILOX system is composed of vehicles, infrastructure, gateway-instances, control terminals, access to AGILOX Analytics and various aids.



No.	Description	No.	Description
1	AGILOX Vehicle	9	Firewall (<i>provided by customer</i>)
2	AGILOX Charging Station (ADSLO)	10	Energy supply (<i>provided by customer</i>)
3	AGILOX Mobile Charging Device (ADMOL)	11	AGILOX LAN (<i>provided by customer</i>)
4	WLAN Infrastructure (<i>provided by customer</i>)	12	Customer LAN (<i>provided by customer</i>)
5	Handheld operating device (<i>provided by customer</i>)	13	Office PC (<i>provided by customer</i>)
6	AGILOX Host Gateway (AGIPC, <i>optional</i>)	14	Host System(s) (<i>provided by customer</i>)
7	AGILOX IO (AGIO, <i>optional</i>)	15	Internet
8	Input / Output Signals	16	AGILOX Analytics (<i>optional</i>)
		17	AGILOX M2M Communication (<i>optional</i>)



INFORMATION

For detailed information on each system, see the **AGILOX ONE Operation Manual, Chapter 3. Description**.

4. Commissioning



WARNING

Risk of crushing

Crushing hazards can occur in the following situations:

- Incorrect operation in manual mode causes the vehicle to hit a person.
- Due to incorrect machine configuration (fork length), the rear personal protection system (ultrasonic sensors in the fork tips) is switched off too early and persons are hit.
- When entering a station, a person passes between the vehicle and the pick-up station (via forks) and is squeezed between the vehicle and the load.

This can lead to serious injuries.

- Manual operation is protected by the authorization system.
- Manual operation may only be carried out by adequately trained and authorized persons.
- Only those personnel who are directly required for the work may be in the danger zone of the machine. In addition, it must be ensured that the smallest possible number of personnel are in the danger zone.
- Use safety shoes on the travel path / in the travel area.

4.1 Maintenance Operation Mode



DANGER

Danger due to deactivated safety features!

In maintenance mode, all safety features will be disabled, posing a risk to the operational safety of AGILOX vehicles and potentially leading to severe injuries and/or damage to property.

- Pay particular attention whenever commands are executed
- The maintenance operation mode must only be activated by specially trained staff
- Prevent accidental or unauthorized activation of the vehicle's maintenance mode by pulling the key from the lock of the key switch

4.1.1 Description

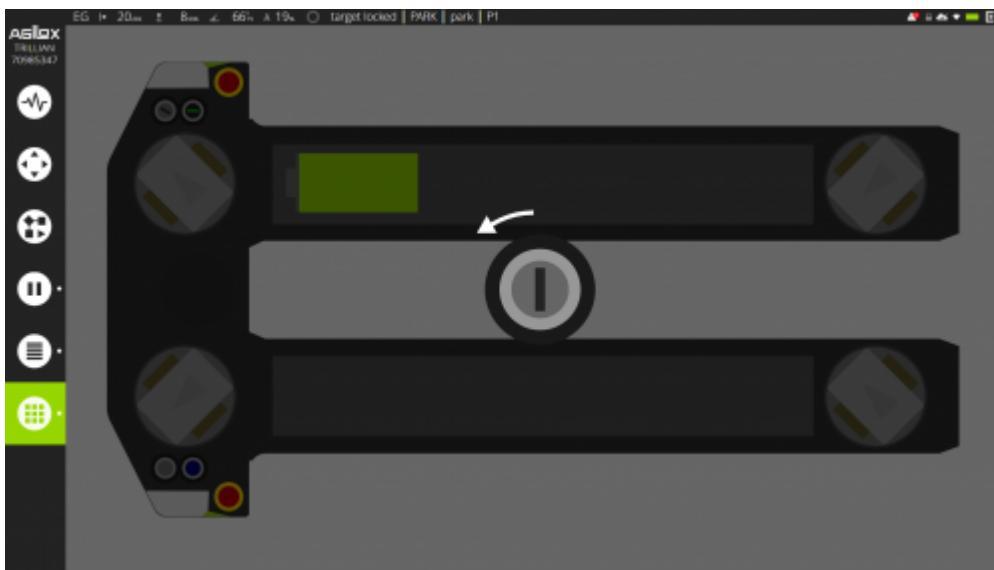
In maintenance mode, users can control all functionalities of an AGILOX vehicle while all of its safety features are disabled. The maintenance mode is mostly required for maintenance work carried out by AGILOX North America, Inc. Trained and authorized maintenance and repair staff is allowed to activate the vehicle's maintenance operation mode in exceptional circumstances only, for example for troubleshooting purposes.

4.1.2 Activation

1. Initiate 'Maintenance Mode' in the AGILOX user interface:



2. Turn the key switch to 'Maintenance Operation Mode' (left-hand position).



3. Press the Start pushbutton to confirm the change of operation mode.
4. Connect a terminal device to the 'AGILOX-<Serial Number>' WLAN hotspot.
→ Password: <Serial Number>



→ The AGILOX vehicle is now in MAINTENANCE OPERATION MODE See [4.3.7.4 - Maintenance Mode \(Vehicle Maintenance\)](#)

4.2 Init Setup

The Init setup allows to assign the AGILOX vehicle to a new or an existing union and to change major network settings.



NOTICE

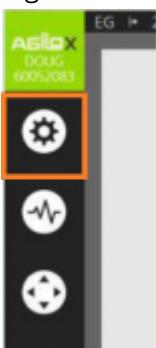
Before performing an Init Setup a Backup of the system should be made!

To start an Init Setup the maintenance mode must be activated. (see Manual [4.1 - Maintenance Operation Mode](#))

After that log in to the visualization as Administrator.

Press and hold both push buttons (Start and Acknowledge) for 5s .

A gears icon appears on the HMI:



Clicking on this icon allows following adjustments:

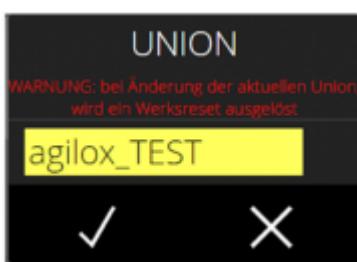
4.2.1 Union

Assign this AGILOX vehicle to a union. If the AGILOX vehicle should be assigned to an existing union, enter the union name here.



NOTICE

If the current union of the vehicle is changed a factory reset of the vehicle is triggered.



INFORMATION

If an AGILOX vehicle with a new software version enters the union, it does not communicate with the other vehicles at first.

The other vehicles of the union recognize the newer software version of the new vehicle and automatically start a software update by downloading the new software version from the new vehicle. After about 2-3s seconds all vehicles of the swarm are up to date and communicate with each other.

4.2.2 WLAN Network

Network settings can be made.

The desired WLAN-network can be selected and the authentication can be made.

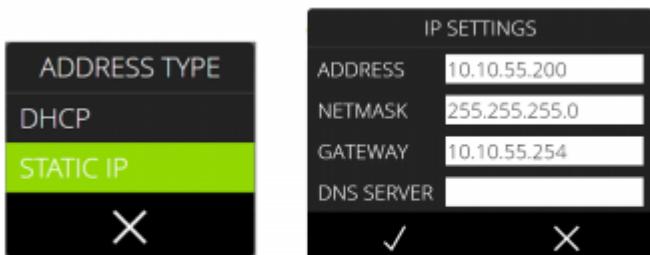


All vehicles and systems of the same union should be connected to the same WLAN Network.

4.2.3 IP - Address

Preferably choose STATIC IP , as with DHCP the IP Address is assigned dynamically and when changing the IP address some settings might not work properly any more.

If DHCP is chosen, the following adjustments can be made:



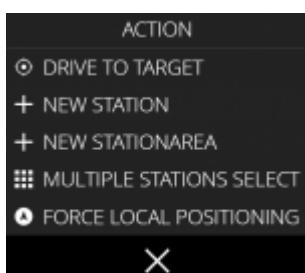
Enter the desired IP ADDRESS and the IP address of the GATEWAY of the corresponding network, adjust the NETMASK and confirm the changes. All vehicles and systems of the same union should have ip addresses of the same network segment.

4.3 Visualization

4.3.1 Context Menus

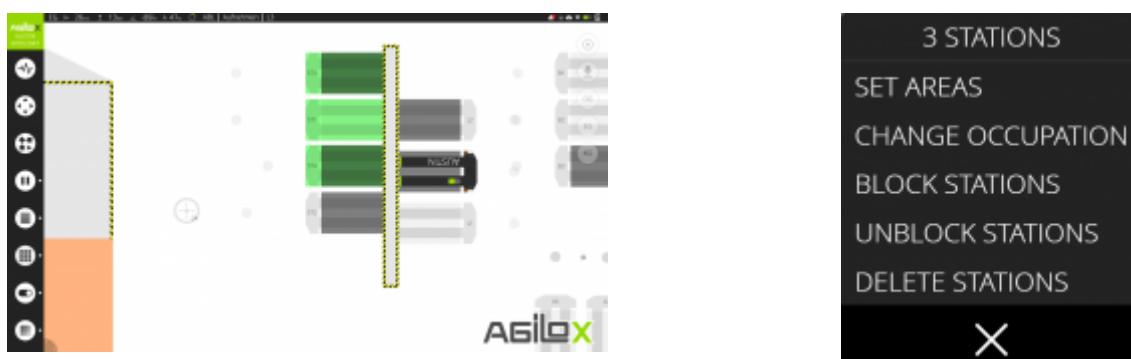
There are various context menus available in the visualization:

4.3.1.1 Context menu on free area:



Description	Information
Drive to Target	Tell AGILOX vehicle to drive to this position.
New Station	Create new station on this position. See 4.3.2 - Stations for details.
New Station Area	Create new station area.
Multiple stations select	Allows selecting multiple stations on the map. Selected stations are shown in green. And a different context menu with further options is shown. See "Multiple Station Select" below.
Force Local Positioning	In case an AGILOX vehicle lost its position, it can be given an approximate position with this option. The AGILOX vehicle will then try to find its position in this area within a 5m radius.

Multiple Stations Select



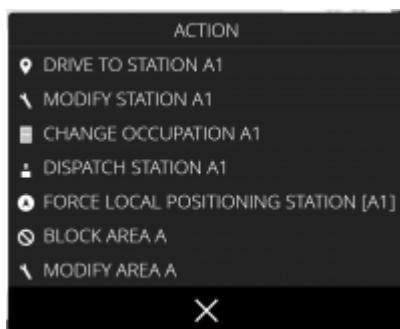
Description	Information
3 Stations	Number of stations selected. In this case 3 Stations.
Set Areas	Assign selected stations to an existing area.
Change Occupation	Set Occupation to empty / occupied / Box Carrier
Block / Unblock Stations	Block / Unblock selected stations
Delete Stations	Delete selected stations

4.3.1.2 Context menu on SVG elements



See [4.5 - Map](#) for more details.

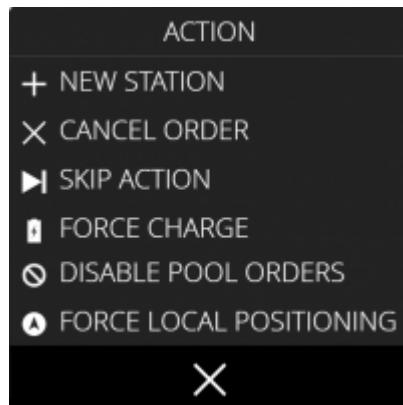
4.3.1.3 Context menu on station:



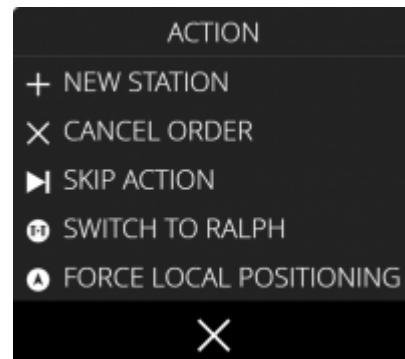
Description	Information
Drive to Station <Station Name>	Tell AGILOX vehicle to drive to the station.
Modify Station <Station Name>	Edit current station. See 4.3.2 - Stations for details.
Change Occupation <Station Name> 	Change occupation of the station. E.g. If a load carrier has been dropped or removed manually, the AGILOX vehicle can only detect that the occupation has changed once it passes by again. In addition, it does not know, whether it is a pallet or the Box Carrier.
Dispatch Station <Station Name>	Dispatch station manually. A destination can be chosen in the next step. There is no prior check if the load carrier fits for this AGILOX vehicle. This is only checked once the vehicle is on the station already.
Force Local Positioning Station [<Station Name>]	In case an AGILOX vehicle lost its position, it can be given an approximate position with this option. The AGILOX vehicle will then try to find its position in this area within a 5m radius.
Block AREA <Area Name>	Disable all stations of current station area.
Modify AREA <Area Name>	Edit current station area.

4.3.1.4 Context menu on AGILOX vehicle:

AGILOX vehicle currently connected to on hmi:



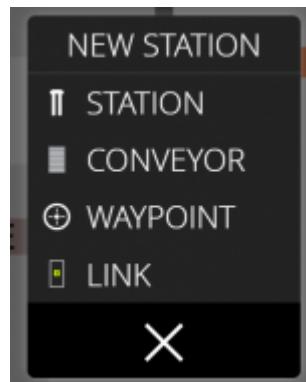
AGILOX vehicle not connected to on hmi:



Description	Information
Cancel Order	Cancel current order
Skip Action	Skip current action
Force Charge	Tell AGILOX vehicle to drive to a charging station and start charging the battery.
Disable / Enable Pool Orders	Disable/Enable pool orders for this AGILOX vehicle
Switch to <AGILOX Name>	Switch to selected AGILOX vehicle.
Force Local Positioning	In case an AGILOX vehicle lost its position, it can be given an approximate position with this option. The AGILOX vehicle will then try to find its position in this area within a 5m radius.
Allow Charge Break (Only for AGILOX vehicles connected to and on charging station)	Allow AGILOX vehicle to break charging earlier to start an order.

4.3.2 Stations

There are four different types for stations:

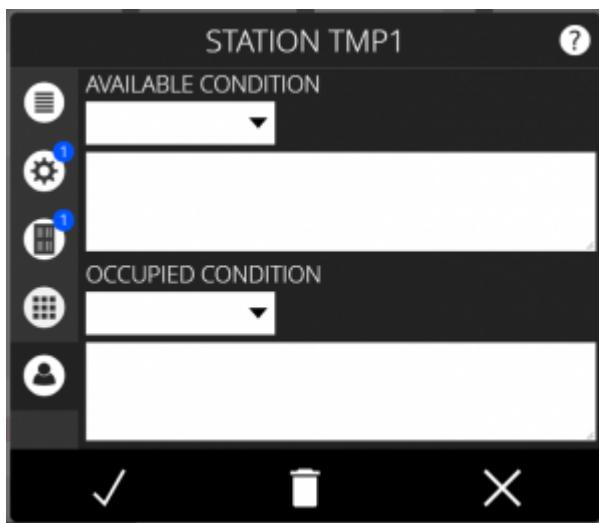


4.3.2.1 Station



No.	Description	Information
1	Station Name	Edit station name
2	Station Parameters	<p>Station parameters such as</p> <ul style="list-style-type: none"> • Position X • Position Y • Position angle • Sort order in station area • Entering liftheight • Relative lift: Leaving lift height = entering lift height + relative lift • Preposition Y: AGILOX is approaching this point before entering a station. Charging Stations mostly need a negative value. • Prepre-Position Waypoint: for narrow aisle, where the AGILOX vehicle cannot turn in front of the station, an existing Waypoint can be selected, where the vehicle can turn and then drive parallel to the station • Tolerance X • VIA: an existing waypoint can be selected, the AGILOX vehicle will always drive over the waypoint first, when driving to that station
3	Menu Bar	<p>Menu bar to navigate through the station parameters.</p> <p>The blue number shows how many selections are made where multiple choice is possible.</p>
4	Options	<ul style="list-style-type: none"> • Station disabled • Charging possible • Parking allowed • Pallet detection: defines whether the pallet can be used as a reference for entering. The orientation, distance, and size of the pallet will be detected. This is possible up to an entering lift height of 300mm, but it always needs to be a floor station. • Occupation detection: defines whether the occupation of the station can and should be detected by the sensors of the AGILOX vehicle.

No.	Description	Information
5	Loadcarrier Type	Allowed load carrier types for this station, multiple choice possible.
6	Station Area	Station area(s) this station belongs to, multiple choice possible.
7	Admin Settings	See description below.
8	Safety Settings	<ul style="list-style-type: none"> Safe Drop: If enabled, station entering for dropping off a pallet is done with the protective field "positioning" until 200mm before the end of the station. Safe Drop+: If enabled, the complete station entering for dropping off a pallet is done with the protective field "positioning" <p>See Operation Manual 6.4.5 - SAFE DROP / SAFE DROP+</p>



Admin Settings

Admin Settings allow to define conditions for availability of the station and for station occupied. The drop-down menu allows to select states of AGILOX IOs or toggles. Additionally, specific conditions can be defined when selecting "Condition" again, after selecting the state/toggle. The selected item will then be prefilled in the box below and condition as in [4.7.5 - Workflow Condition Object](#) can be defined.

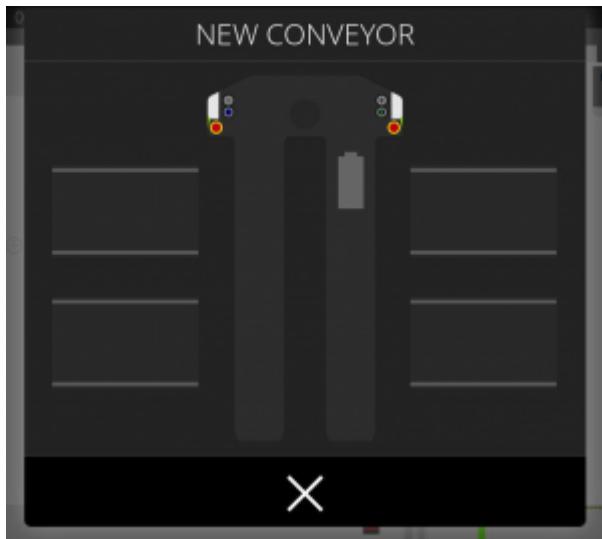
As long as the availability condition is not fulfilled and in addition no event is defined for this station and no occupied condition via an AGILOX IO, the station is set "empty" and the occupancy is not checked by the AGILOX vehicle. This is necessary to be able to drive over the station.

For AGILOX IO signals as occupied condition, "latency" can be defined in first level. The input has to be present for the defined time in seconds to be "true" (debouncing of input).

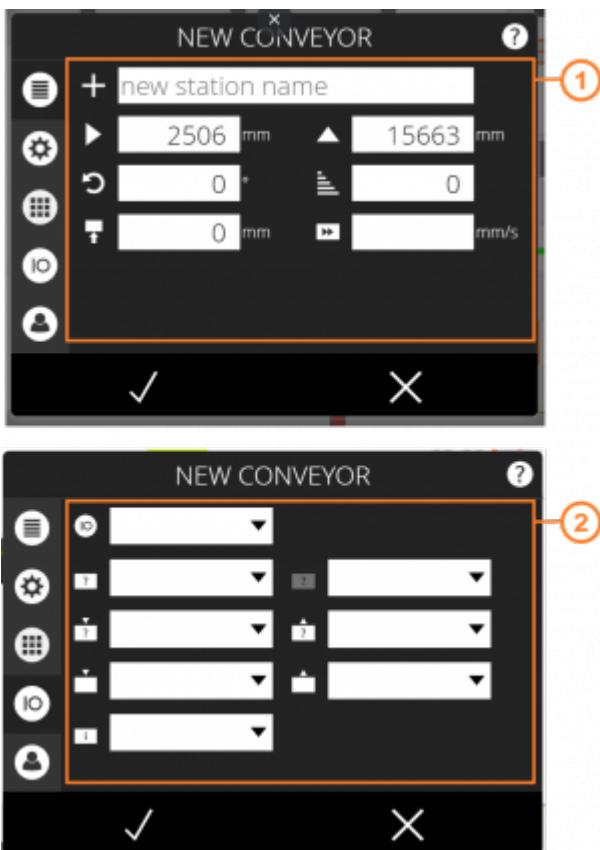
```
{
  "latency":5,
  "agiloxio": {
    "test": {
      "state": {
        "Drop high": true
      }
    }
  }
}
```

4.3.2.2 Conveyor

To define a new conveyor station, the AGILOX vehicle needs to pick up the AGILOX Box Carrier first. After that drive the AGILOX vehicle together with the Box Carrier right in front of the conveyor in manual mode. Once the AGILOX vehicle is in the right position for hand-overs, the conveyor station can be created via the context menu on the area of movement - "New Station". The following window will pop up:



The position of the conveyor station can be picked and the hmi automatically fills in the correct coordinates for the conveyor station.



In contrast to Station, the following adjustments can be made:

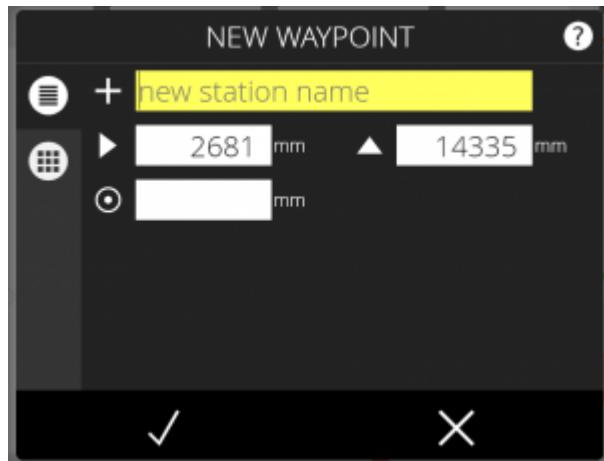
No.	Description	Information
1	General	Edit station name and parameters such as <ul style="list-style-type: none"> • Position X • Position Y • Position angle • Sort order in station area • Entering liftheight • Conveyor speed
2	AGILOX IO Settings	Assignment of AGILOX I/Os <ul style="list-style-type: none"> • AGILOX IO • Occupied flag • Capacity • Ready for pickup flag • Ready for drop flag • Start pickup • Start drop • Announce pickup/drop - optional Tooltips show which one is which



CAUTION

Check the correct assignment of all I/O signals before automatic operation!
Wrong assignment can lead to boxes falling off the conveyor.

4.3.2.3 Waypoint



In contrast to Station, the following adjustments can be made:

- Station Name
- Position X
- Position Y
- Waypoint radius

4.3.2.4 Link



In contrast to Station, the following adjustments can be made:

No.	Description	Information
1	General	Edit station name and parameters such as <ul style="list-style-type: none"> • Position X • Position Y • Position angle • Preposition Y • Prepre-Position Waypoint (see station parameters above) • VIA (see station parameters above)
2	AGILOX IO Settings	Assignment of AGILOX I/Os <ul style="list-style-type: none"> • Leave request • Leave wait condition • Lock request • Lock wait condition Tooltips show which one is which
3	Link to Station	Assignment of linked Stations Station names, the link station links to on the different floors

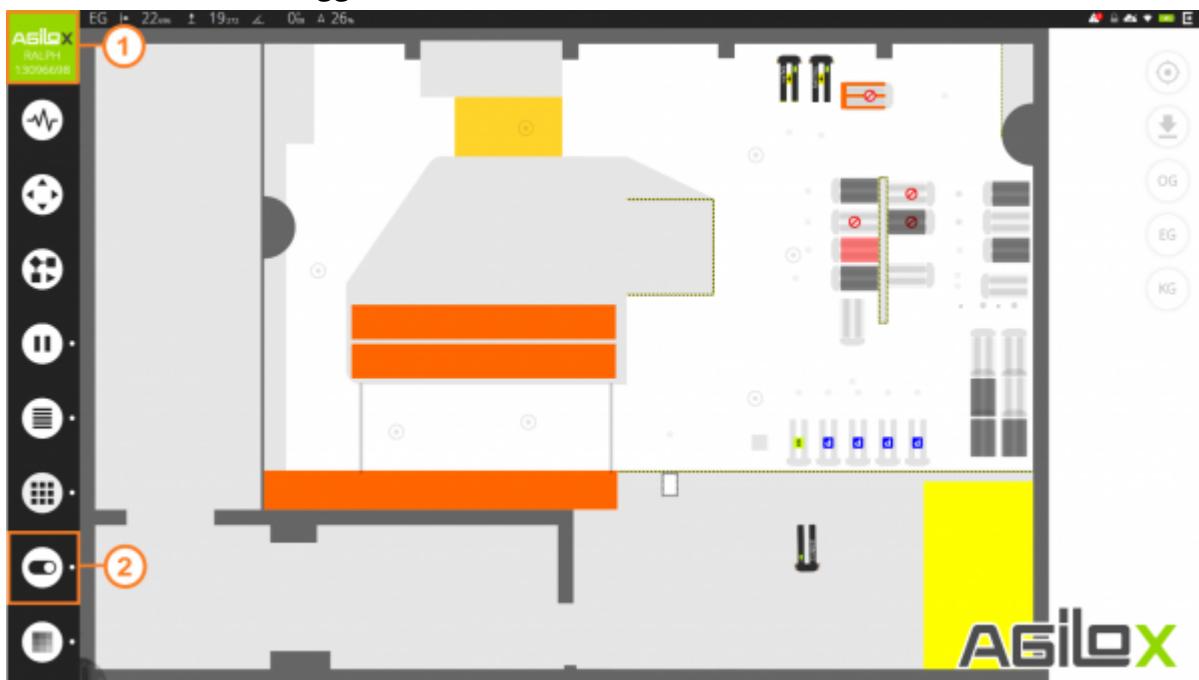
4.3.3 Vehicle Information

For ADMIN users there are additional settings in vehicle information:



No	Description	Information
1	Toggle LED test mode	Start/Stop LED test mode. All lamps of the AGILOX vehicle will be switched on (Signal lamps, floor spot, camera light, ...) and can be checked for their functionality.
2	Toggle audio test mode	Start/Stop audio test mode. A pilot tone will be sounded when clicking on the loudspeaker symbol and the speaker can be checked for its functionality.
3	Barcode	Barcode test mode can only be activated when the vehicle is in maintenance mode. After clicking on this symbol, a live image of the camera is shown and the camera/barcode can be checked.

4.3.4 Visualization Toggles

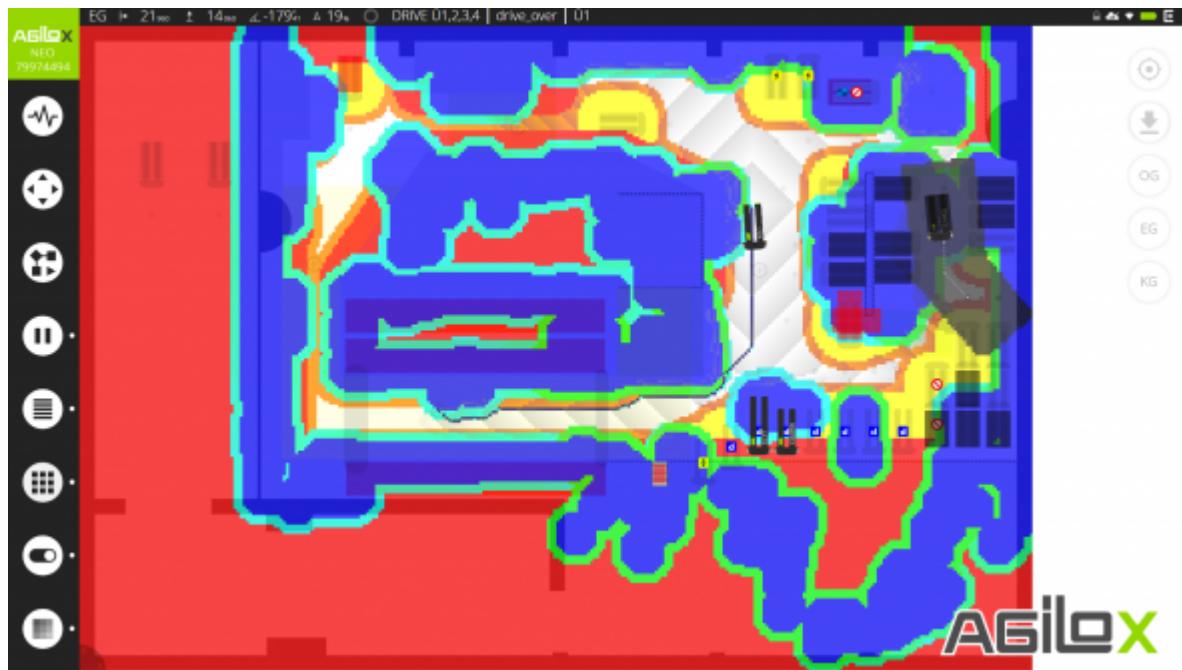


No.	Description
1	Visualization
2	Visualization toggles



No.	Description	Information
1	Show live route map in visualization	
2	Show map in visualization	
3	Erase map areas	For erasing Contours and Scan data.

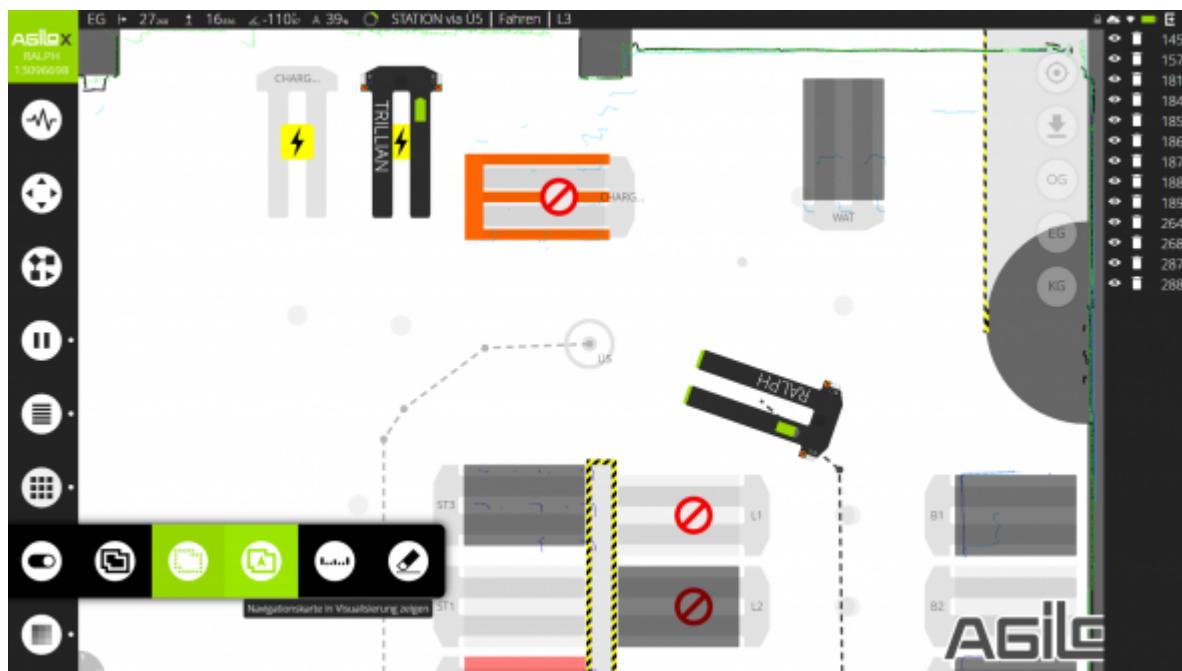
4.3.4.1 Show Live Route Map in Visualization



This map overlay shows the basis for finding a route. Additionally, to the route map above, all actual detected obstacles are shown live on the map as blue areas.

The function worsens performance and should merely be used for diagnostic purposes.

4.3.4.2 Contours



Shows the initially acquired ('learned') map AGILOX vehicles use as a basis for determining their position. Drawn in as black lines.

This illustration shows contours in combination with the scan data from [Operation Manual 6.2.6.3 - Display Scan Data](#)

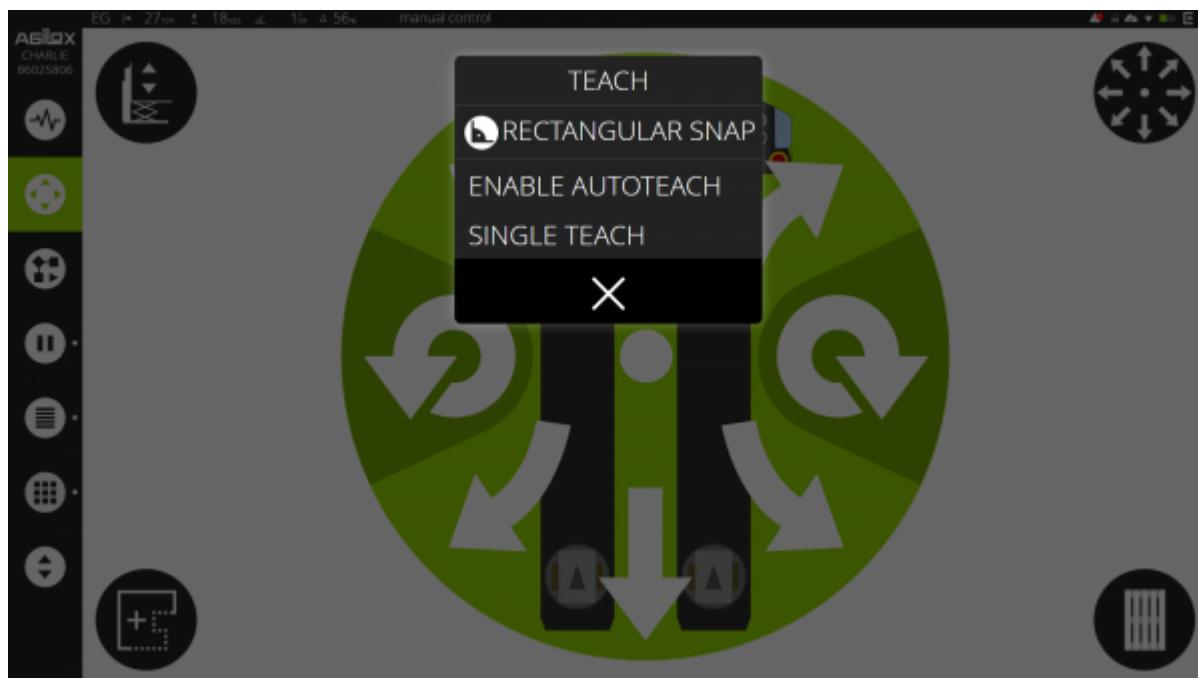
4.3.5 Manual Operation

To enable the manual operation mode of an AGILOX vehicle, the key switch has to be turned to Manual Operation. (see [Operation Manual 6.1.4 - Manual Operation Mode](#))



No.	Description	Information
1	Manual operation	
2	Add new contour from navigation-scanner	

4.3.5.1 Add new Contour



No.	Description	Information
1	Rectangular Snap	Force rectangular teaching: Tries to find rectangular contours. And adjust map accordingly.
2	Enable Autoteach	If activated and map quality lower 25% a teach will be triggered.
3	Single Teach	Teach triggered at current position.

See [4.4. - Teach Contours](#) for detailed information.

4.3.6 Reports

- ①
- ②
- ③
- ④



No.	Description	Information
1	Log	Show log file of selected AGILOX vehicle.
2	Failures	Show failures of selected AGILOX vehicle.
3	Notification	
4	Network tester	

4.3.6.1 Vehicle Log



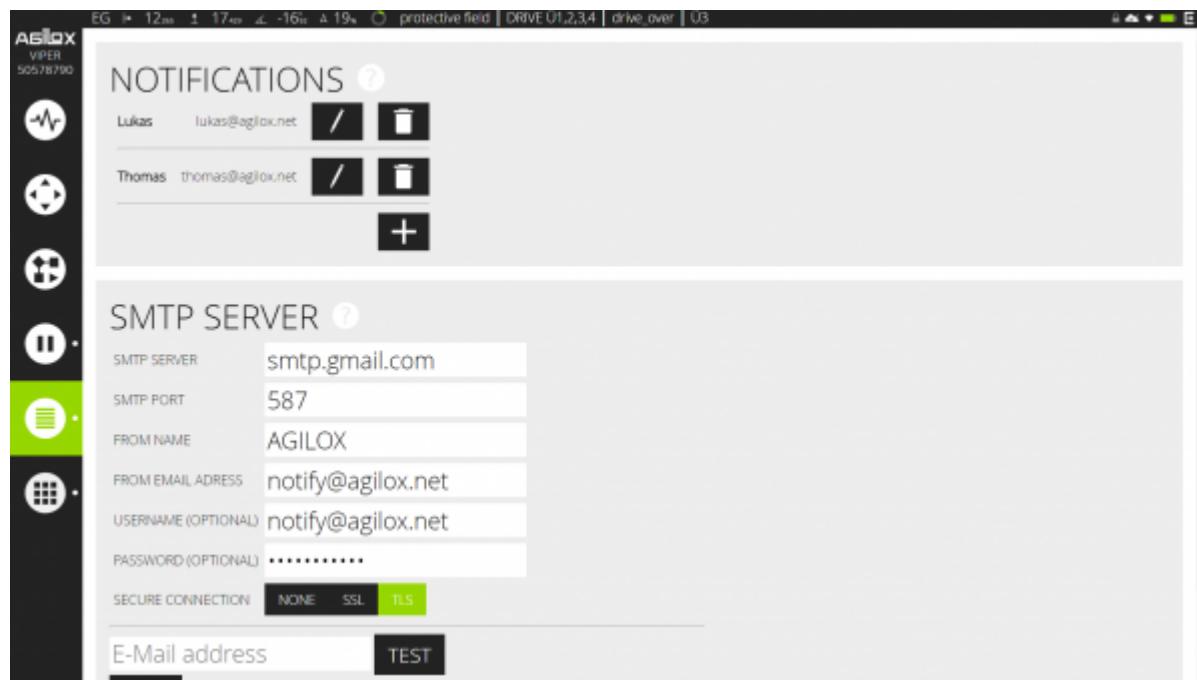
No.	Description	Information
1	Download log	Download log file
2	Module visibility	Enable and disable different information

The yellow bar at the top is a search box for a text search in the log file.

4.3.6.2 Failures

All Failures of the selected AGILOX vehicle are shown in the failure log. Search function works the same as in Vehicle log files.

4.3.6.3 Notification



The AGILOX System can send Notifications to defined e-mail addresses that include information about pending issues.

SMTP Server Settings for E-Mail sending can also be made and tested here.

4.3.6.4 Network Tester

UPTIME

10:08:30 up 3 min, 0 users, load average: 1.92, 1.31, 0.56

WIRELESS

```
Interface wlpis0
  ifindex 7
  wdev 0x1
  addr 04:ce:56:ac:83:0c
  ssid AGuest
  type managed
  wiphy 0
  channel 36 (5180 MHz), width: 40 MHz, center1: 5190 MHz
  txpower 22.00 dBm
  multicast TXQ:
    qsz-byt qsz-pkt flows drops marks overlim hashcol tx-bytes      tx-packets
    0       0       0       0       0       0       0       0       0
```

NETWORK

#kernel	Interface	RX Pkts/Rate	TX Pkts/Rate	RX Data/Rate	TX Data/Rate
		RX Errs/Drop	TX Errs/Drop	RX Over/Rate	TX Coll/Rate
	lo	11 0	11 0	2372 0	2372 0
		0 0	0 0	0 0	0 0
	erp2s0	211 0	61 0	198730 0	6254 0
		0 0	0 0	0 0	0 0
	erp0s31fg	807 0	7 0	1133K 0	549 0
		0 0	0 0	0 0	0 0

Network status of AGILOX Vehicle with ping function.

TRACEROUTE

10.10.55.254 TRACE

PORT SCAN

10.10.55.254 80 SCAN

DNS LOOKUP

70710457.agilox.net DNS LOOKUP

Webservice

http://localhost:8100/ci/self/battery/level

POST-DATA

CALL

Furthermore the connection to a webservice from vehicle's view and webservice POSTs can be tested.

4.3.7 Misc. Tools and Settings

In the system's misc. tools and settings, users can by default switch between the AGILOX vehicles of a UNION (7) and also log out (11), if they wish to do so. Administrators have access to additional functionalities.



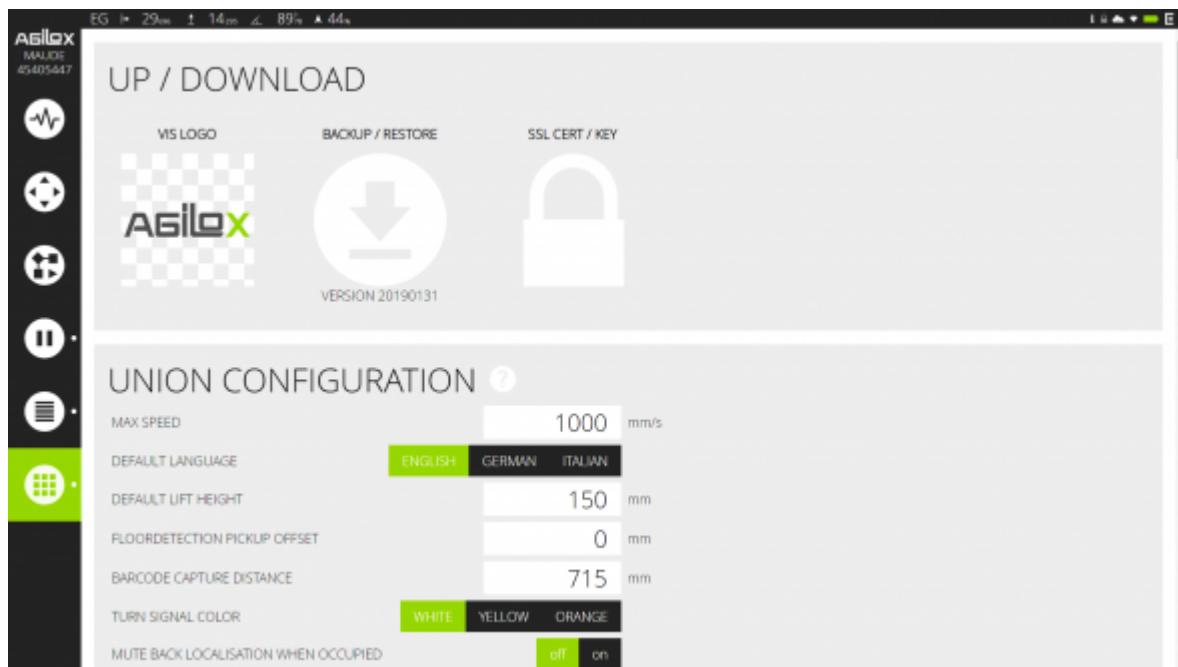
If the hmi is connected to a host gateway, the following will be displayed:



No.	Description
1	System Settings
2	System Database Editor
3	Workflow Settings
4	Vehicle Maintenance
5	AGILOX IO
6	Map Settings
7	Shutdown AGILOX / Shutdown or Reboot Host Gateway

4.3.7.1 System Settings

The system setting options can be used for project-specific adjustments, for adapting UNION configurations as well as backing up and restoring software.



Logo (VIS LOGO)

- Removal or replacement of the customized logo displayed in the visualization, e.g. customer logos

Backup copies and restoring (BACKUP / RESTORE)



INFORMATION

It is recommended to perform a backup before implementing any modifications

- Click to generate and download backup copies
- Use drag & drop to update and restore backup copies



NOTICE

Risk of damage to property

As soon as a backup copy has been loaded into an AGILOX vehicle, it will be immediately distributed to all the AGILOX vehicles being part of the same UNION by swarm intelligence! The entire UNION process can be permanently impaired by loading an incorrect backup copy!

- Only use this functionality after adequate training

SSL CERT/KEY

- Upload SSL Certificate or key with drag & drop function.
- See [4.8.1.4 - JSON API - Security](#) for details.

Setting the AGILOX vehicle parameters (UNION CONFIGURATION)

UNION CONFIGURATION

MAX SPEED	1400	mm/s					
DEFAULT LANGUAGE	ENGLISH	GERMAN	ITALIAN	CZECH	FRENCH	CHINESE	DANISH
DEFAULT LIFT HEIGHT	155 mm						
LOWERED LIFT HEIGHT	0 mm						
PALLET DETECTION PICKUP OFFSET	5 mm						
BARCODE CAPTURE DISTANCE	500 mm						
BATTERY PLAN CHARGE LEVEL	40 %						
TURN SIGNAL COLOR	WHITE	YELLOW	ORANGE				
STRICT LOAD CARRIER DETECTION	off	on					
BEND AUDIO SIGNAL	off	on					
DISABLE AUTOMATIC CARRIER SWAP	off	on					
 	-						
CI TIMEOUT	60 s						
HARDWARE BUTTON CANCEL TIME	5 s						
INTERLEAVE ADDITIONAL HULL	0mm	100mm					
LOGIN FOR WORKFLOW	off	on					
LOGIN FOR VISUALIZATION	off	on					
DISABLE HTTP PORT 80	off	on					
DISABLE COLLISION AVOIDANCE FALL CHECK	off	on					
INVALID CHANNEL ENTERING DENIED	off	on					
PASSWORD POLICY	EASY	MEDIUM					



INFORMATION

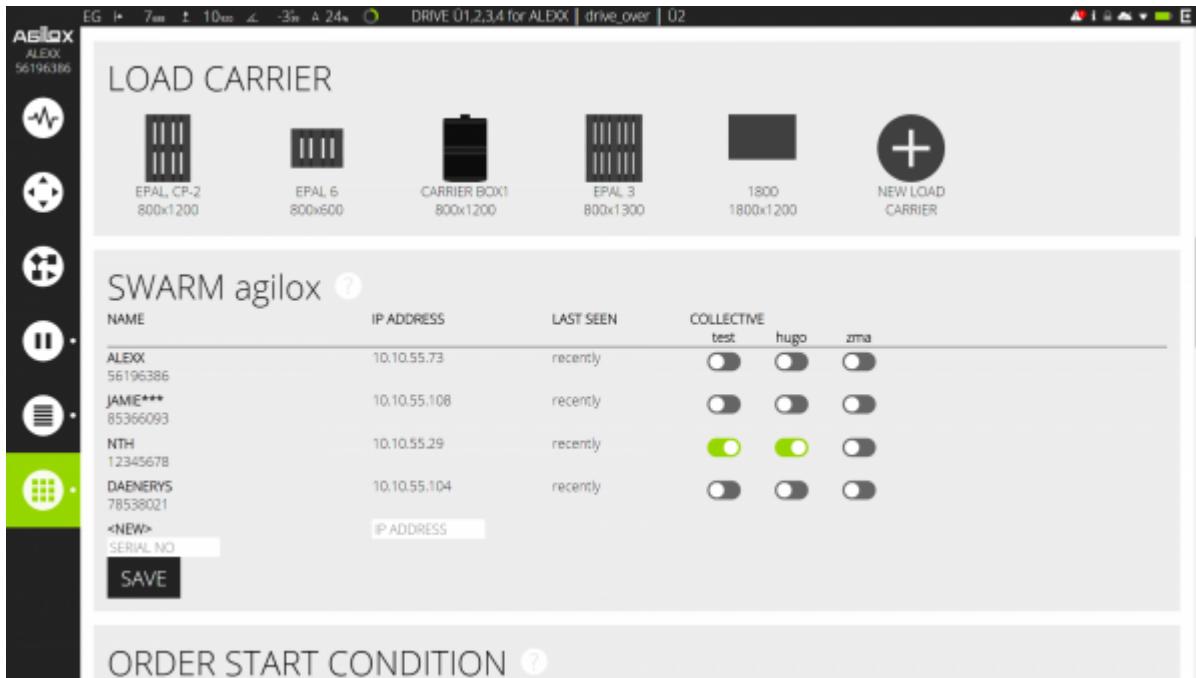
As a rule, these parameters must not be modified.

Any changes of Parameters will affect all AGILOX vehicles being part of the UNION.

Setting options:

Description	Information	Range	Default
MAX SPEED	Set maximum speed in mm/s.	50 - 1400 mm/s	1400 mm/s
DEFAULT LANGUAGE	Set default language for AGILOX user interface. Users may set their preferred language in user management (comes in effect when logged in).		English
DEFAULT LIFT HEIGHT	Lift height AGILOX is using while driving and being occupied. Should be defined in a way the load is just above the safety scanner level. Note: if too low, parts of the load might reach into the plane observed by safety scanners, which causes AGILOX to stop.	100 - 300 mm	150 mm
LOWERED LIFT HEIGHT	Setting for minimum lift height while driving, to guarantee to be seen by other third party vehicles. Adjustable in 5mm steps.	0 - 100 mm	20 mm
PALLET DETECTION PICKUP OFFSET	Adjust the distance between AGILOX vehicle and the pallet when picking it up, so how far into the pallet should the AGILOX vehicle go before picking it up. Higher Value means more distance to the pallet at pickup. Is used if pallet feet are not level with the pallet itself	-50 - +50 mm	0 mm
BARCODE CAPTURE DISTANCE	AGILOX stops at this given distance to the load, to capture a barcode photo if specified in the workflow.	350 - 1200 mm	715 mm
BATTERY PLAN CHARGE LEVEL	Once the battery level is below the adjusted plan charge level, the AGILOX vehicle will charge its battery after the order is completed and before starting a new order.	40 % - 70 %	60 %
ACCELERATION	If the PLC version allows this, the acceleration of the AGILOX vehicles can be adjusted here.	100 - 450 mm/s ²	350 mm/s ²
TURN SIGNAL COLOR	Set color for visual signal when AGILOX vehicle turns	white / yellow / orange	white
STRICT LOAD CARRIER DETECTION	See Operation Manual 6.4.3 - Load Carrier Detection Strict detection can handle noncontinuous pallet feet but is very strict on tolerances. The default "lazy" mode whereas needs continuous pallet feet but handles pallet placement more tolerant. For stations exclusively used for load carrier type EPAL, however, tolerant load carrier detection is always used (even if strict load carrier detection in system settings is set to ON)	OFF / ON	OFF
BEND AUDIO SIGNAL	If set to ON, the driving backwards sound is also emitted when the AGILOX vehicle changes direction.	OFF / ON	OFF

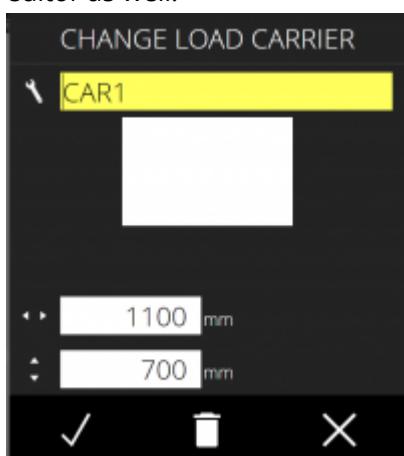
Description	Information	Range	Default
DISABLE AUTOMATIC CARRIER SWAP	If set to ON, the AGILOX vehicle will not change to the Box Carrier automatically unless an order requiring the Box Carrier is designated for this vehicle only.	OFF / ON	OFF
CI TIMEOUT	The CI Timeout determines how long the system waits for suddenly vanished AGILOX vehicles.	60 s - 3600 s	900 s
HARDWARE BUTTON CANCEL TIME	Defines how long the white "start" push button needs to be pressed for the order to be cancelled.	5 s - 3600 s	5 s
INTERLEAVE ADDITIONAL HULL	Defines how much space is left between an AGILOX vehicle including its load and another AGILOX vehicle passing by.	100mm / 0mm	100 mm
LOGIN FOR WORKFLOW	If set to ON, users need to login to use the workflow view (sometimes referred to as "easybuttons")	OFF / ON	OFF
LOGIN FOR VISUALIZATION	If set to ON, users need to login to even see the visualization.	OFF / ON	OFF
DISABLE HTTP PORT 80	HTTP port 80 can be disabled to only use HTTPS.	OFF / ON	OFF
DISABLE COLLISION AVOIDANCE FALL CHECK	Fall check of obstacle avoidance sensors can be deactivated. This can be useful for some backlight conditions, where the fall check can be error-prone.	OFF / ON	OFF
STATION ENTERING CURRENT MONITORING	See Operation Manual 6.4.6 - Current Monitoring during Station Entering If set to ON, the AGILOX vehicle is monitoring the motor currents when entering a station. If the current suddenly increases unexpectedly, the AGILOX vehicle stops immediately.	OFF / ON	OFF
INVALID CHANNEL ENTERING DENIED	If this system setting is enabled, a channel with an invalid station is not used for any action. The channel is still checked every time, the AGILOX vehicle wants to access it. If the channel with the invalid station is the only target of the order, the event "no_station_left" can be used to react to the situation.	OFF / ON	OFF
PASSWORD POLICY	EASY: no password policy MEDIUM: • min. 10 characters • user name and password must not be the same • user name must not be part of password • must contain at least one small letter, capital letter, number and special character	EASY / MEDIUM	EASY



Load Carrier

Specific load carrier types can be defined, changed or deleted here. The standard load carrier types EPAL, EPAL6 or AGILOX Box Carrier can not be changed or deleted.

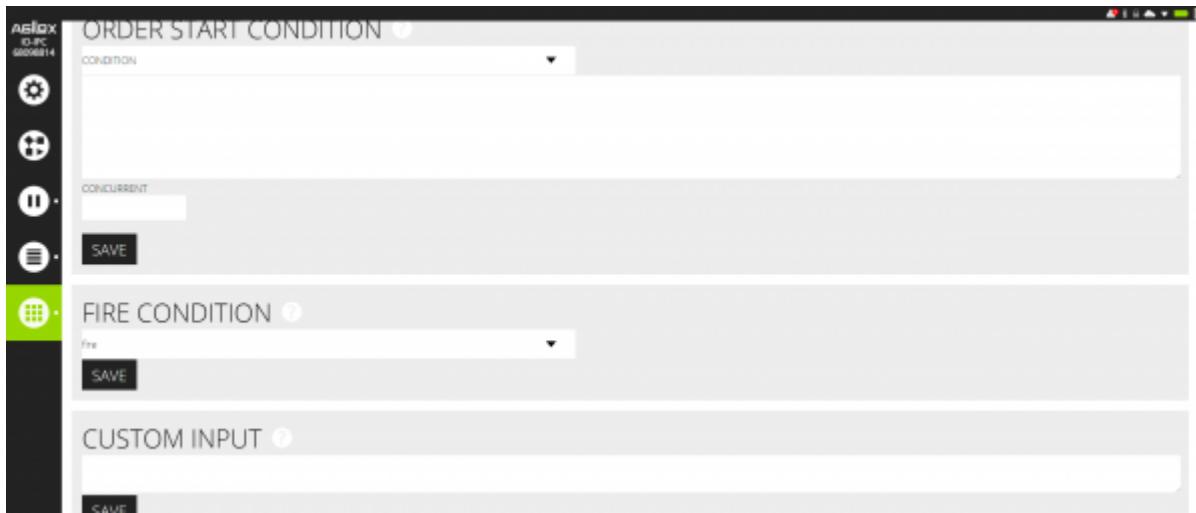
Any changes here will automatically change the collection "loadcarrier" ([4.6.6 - loadcarrier](#)) in system database editor as well.



Swarm <name>

Shows the AGILOX swarm with all AGILOX vehicles, their IP addresses and serial numbers. New vehicles can be added to the swarm here.

Each AGILOX vehicle can be assigned to one or more collectives. Collectives can be created in the collection "collective" ([4.6.3 - collective](#)) in the database editor.



Order Start Condition

A general order start condition can be defined. No order, except for internal orders (charge, park, dedeadlock), will be started, if the condition is not true. "Concurrent" defines how many AGILOX vehicles of the unit are allowed to do orders at the same time.

Fire Condition

A condition for the fire alarm can be defined here. If this condition is fulfilled, the fire alarm is triggered.

Custom Input

Starting from AGILOX Motion Version 3.26.1 an input on the Beckhoff PLC in the control cabinet of the AGILOX vehicle is provided for the customer for customer-specific sensors on the AGILOX vehicle:

-PLC1-DE1 → I7

If this input is true (i.e. +24V present) the actions defined here in system specifications will be executed. At the moment the same functions as on action areas are possible ([4.5.5 - Action Settings](#)).

The first level definition needs to be "action". Additionally the input can also be inverted by adding the attribute "inverted":true. (Default: false)

Example:

```
{  
    "action": {  
        "speed_max": 100  
    },  
    "inverted":true  
}
```



Order Tuning

Tuning and sorting orders by their priority, creation time and distance.

Each AGILOX is calculating a rating(score for each pending order. Those sliders change the significance of various parameters.

Slider to the left means "not important at all", slider to the right means "very important".

Note: usually there's no need to change any of those sliders. But if you want to force AGILOX to strictly do the orders in the order they were placed, you could put the "creation time" slider to the far right and all others maximum left.

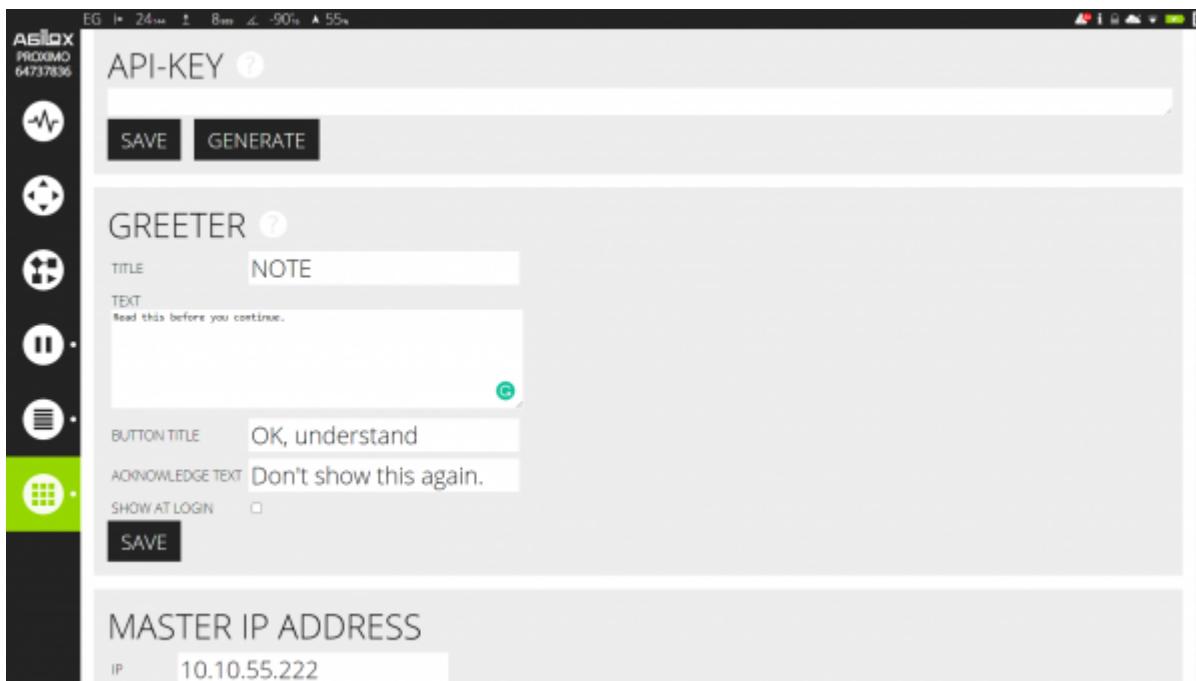
Slider	Description
PRIORITY	Dynamically set directly on the order by host-system or once defined in a workflow. See 4.7.2 - Workflow Order Object
CREATION TIME	Older orders will normally be processed before newer ones. But priority or distance might override the age.
DISTANCE	The AGILOX vehicle might prefer an order, if it is very near to its actual position. Enforce this, by putting the slider to the far right.
CARRIER CHANGE	The AGILOX vehicle might prefer orders, that don't need a carrier change. Enforce this, by putting the slider to the far left.
FLOOR CHANGE	The AGILOX vehicle prefers orders, that don't need a floor change. Enforce this, by putting the slider to the far left.

Audio Setup

Description	Information
VOLUME	Set volume for audio signals on all AGILOX vehicles of the union.
DRIVING BACKWARDS SOUND	Set pitch for driving backwards sound.

Time

Set timeserver for system time and timezone.



API-KEY

An API key can be created and saved. This key must then be used in every order for the entire union. See [4.8.1.4 - JSON API - Security](#) for details.



NOTICE

Risk of damage to property

The entire UNION process may be permanently impaired/disrupted by such editing.

- This functionality should only be used by persons with adequate skills

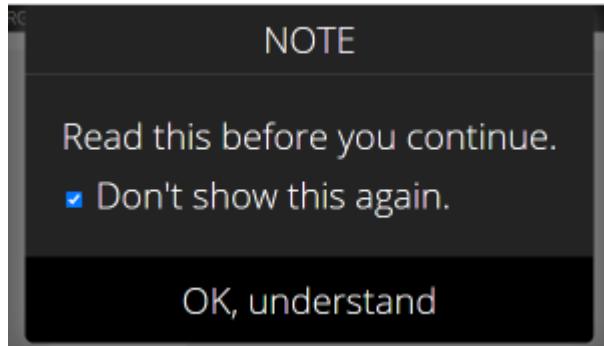
Greeter

A greeter (information window) can be created, which is displayed after login, or important information can be added to the login window.

Attribute	Description
TITLE	Title for the greeter
TEXT	Information text
BUTTON TITLE	Text shown on the OK button of the greeter. If empty, a simple check item will be shown.
ACKNOWLEDGE TEXT	A Text can be entered here (e.g. "Don't show this information again."). If the checkbox next to this text is checked, the greeter will not be shown again the next time this user logs in. If the text is left blank, the information will be shown again at every login.
SHOW AT LOGIN	If this is activated, the information is shown already at the login window. The checkbox must be checked to be able to log in. Button title and acknowledge text are not used for this.

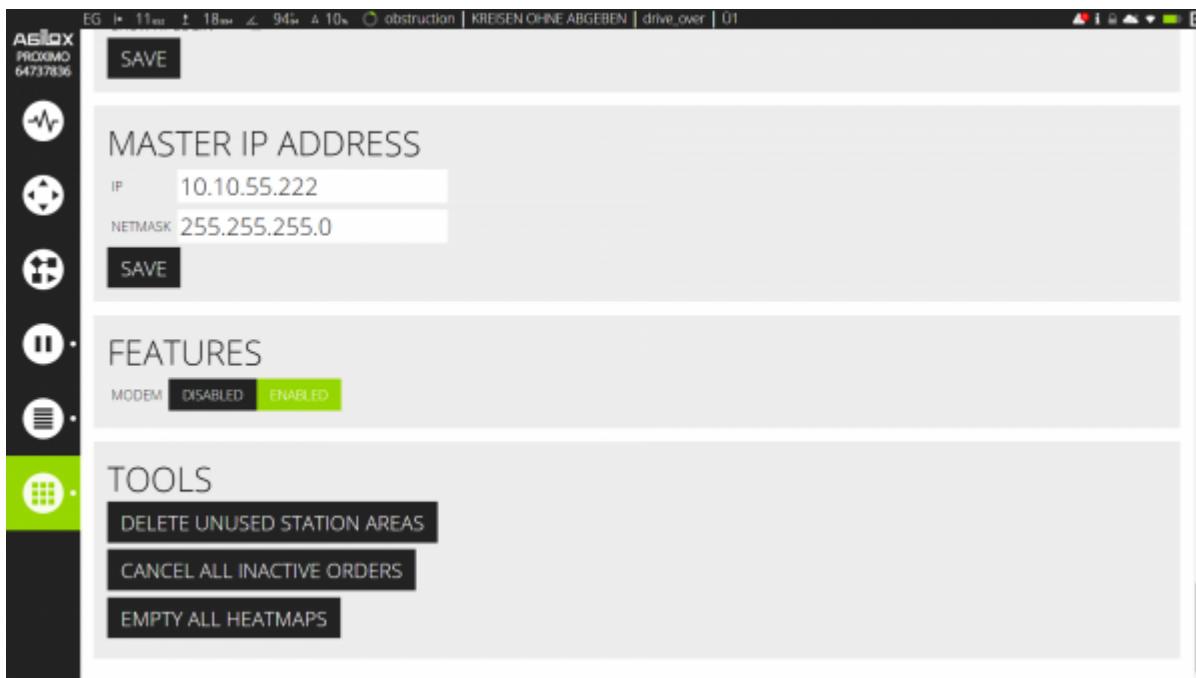
Example

The above settings lead to the following information window:



Example for "SHOW AT LOGIN"





Master IP Address

A Master IP Address can be set here.

It is always possible to connect to an AGILOX vehicle with this IP Address (master). The master is the AGILOX vehicle with the highest serial number.

Features

This is to enable / disable the internal modem.

Tools

All unused station areas can be deleted, inactive orders can be canceled and all heatmaps can be emptied.

4.3.7.2 System Database Editor



NOTICE

Risk of damage to property

The entire UNION process may be permanently impaired/disrupted by such editing.

- This functionality should only be used by persons with adequate skills

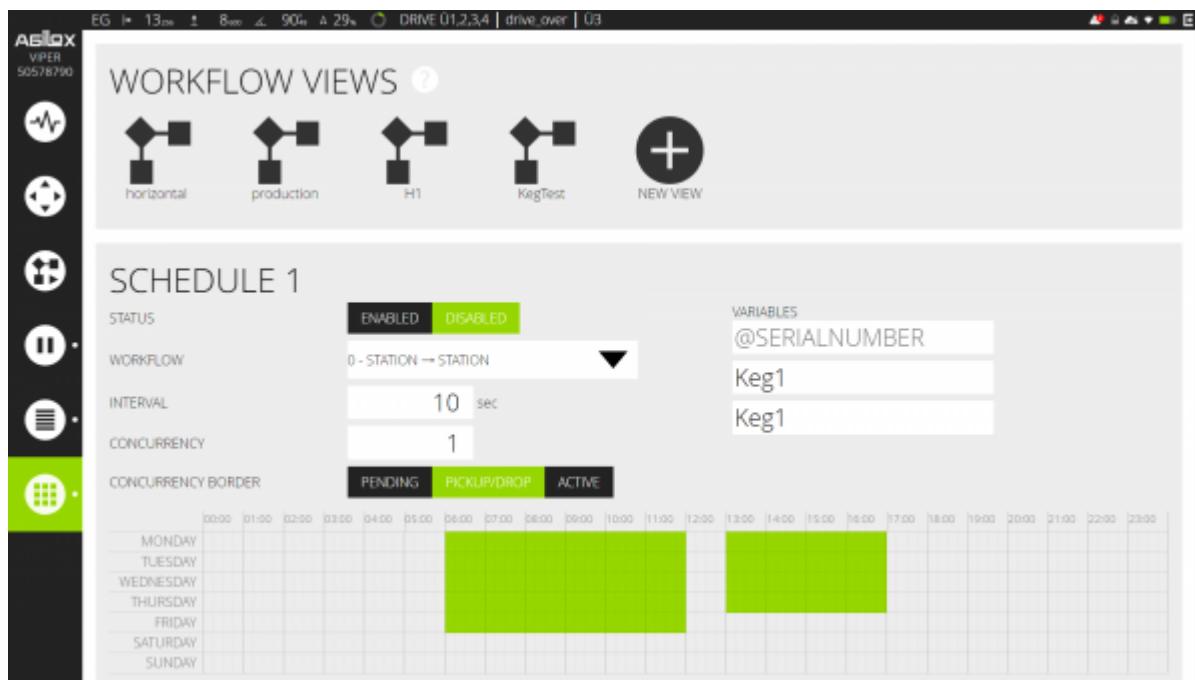
The screenshot shows the agloxi Neo mobile application interface. The top bar displays "EG 22m S -127m A 46m" and "manual control". The left sidebar contains icons for search, barrier, floor, supply_dispose, and Workflow (highlighted in green). The main content area shows a JSON configuration for a workflow named "STATION to STATION". The configuration includes two stations, "station": "STATION-STATION", with status 0, carrier "repal", and order 1. It defines two recipient types: "B": serialnumber: "SOURCENUMBER" and "D": serialnumber: "DESTINATIONNUMBER". The workflow consists of two actions: "pickup" at station "SOURCE;Source;station" and "drop" at station "DESTINATION;Destination;station". The second section of the configuration is for "STATION to AREA", with status 0, carrier "repal", and order 2. It also defines two recipient types: "B": serialnumber: "SOURCENUMBER" and "D": serialnumber: "DESTINATIONAREA". The workflow consists of "pickup" at station "SOURCE;Source;station" and "drop" at station "DESTINATION;Destination;stationarea". The bottom of the screen shows a footer with "Local Control" and a battery icon.

```
1: {
2:   "0": {
3:     "name": "STATION-STATION",
4:     "status": 0,
5:     "carrier": "repal",
6:     "order": 1,
7:     "recipient": [
8:       {
9:         "B": {
10:           "serialnumber": "SOURCENUMBER"
11:         }
12:       },
13:       {
14:         "action": {
15:           "B": {
16:             "action": "pickup",
17:             "station": "@SOURCE;Source;station"
18:           },
19:           "D": {
20:             "action": "drop",
21:             "station": "@DESTINATION;Destination;station"
22:           }
23:         }
24:       }
25:     ],
26:     "1": {
27:       "name": "STATION to AREA",
28:       "status": 0,
29:       "carrier": "repal",
30:       "order": 2,
31:       "recipient": [
32:         {
33:           "B": {
34:             "serialnumber": "SOURCENUMBER"
35:           }
36:         },
37:         {
38:           "action": {
39:             "B": {
40:               "action": "pickup",
41:               "station": "@SOURCE;Source;station"
42:             },
43:             "D": {
44:               "action": "drop",
45:               "stationarea": "@DESTINATION;Destination;stationarea"
46:             }
47:           }
48:         ]
49:       }
50:     }
51:   }
52: }
```

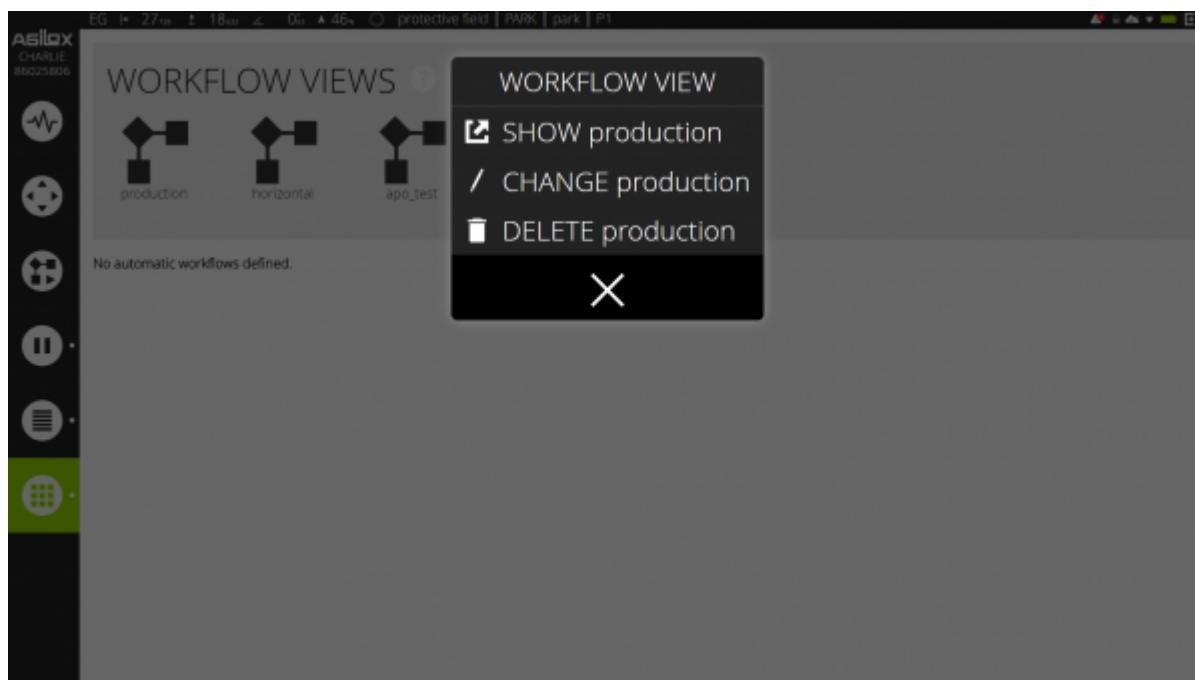
The database editor can be used to edit configurations for AGILOX IO, workflows and supply-dispose function. See [4.7 - Workflow Definition Language](#), [4.6 - Collections](#) and [AGILOX IO Operation Manual 5.3 - Collection agiloxio](#) for details.

4.3.7.3 Workflow Settings

Workflow views for operators (easy buttons) can be created and adjusted here and schedules can be created.

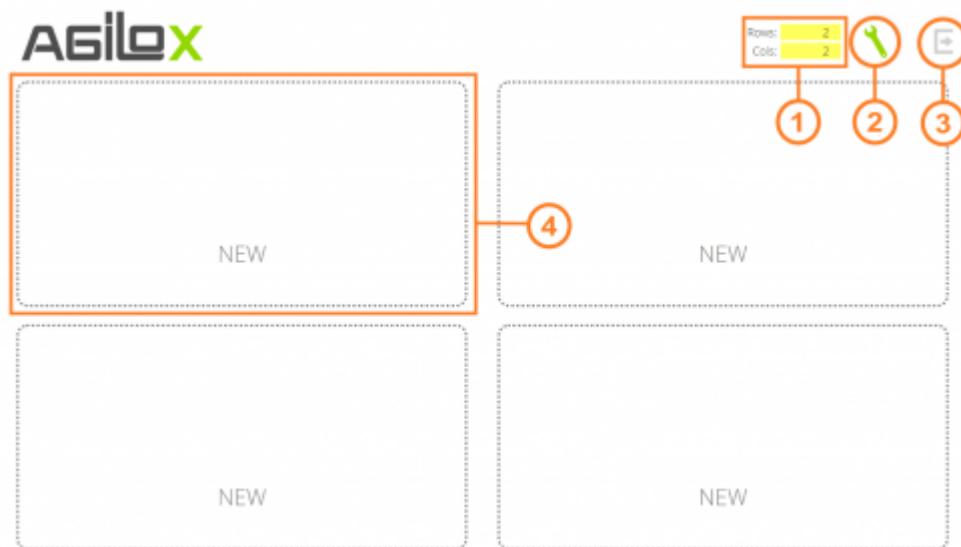


When clicking on a workflow view, the view can be shown, changed or deleted.



Create new views or edit existing views:

The name of the workflow view can be adjusted at first. After confirmation the workflow view opens and can be edited.



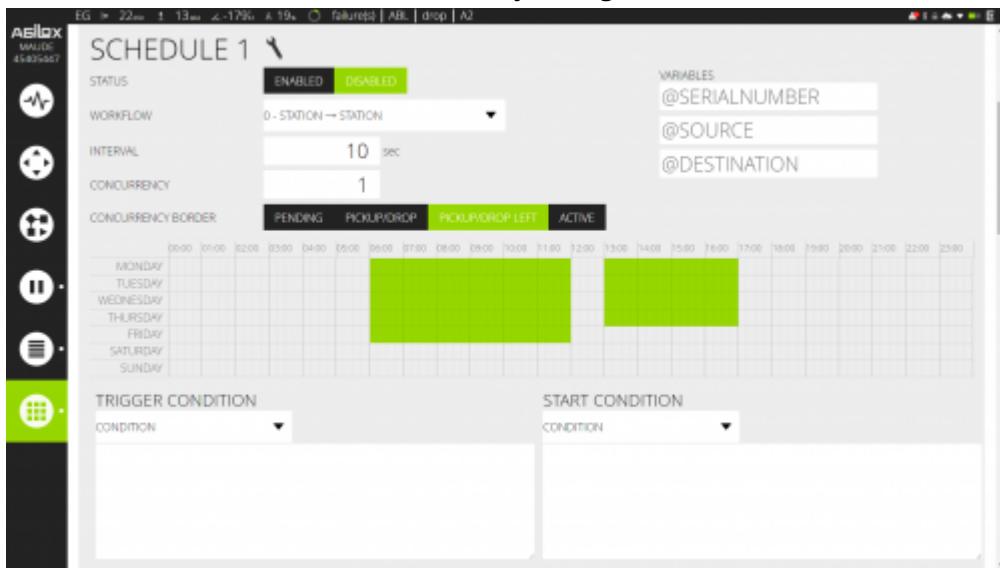
No.	Information
1	The number of rows and columns for the workflow view can be adjusted here.
2	Switch between edit view and user view.
3	Log out / Log in
4	Edit buttons

Edit buttons


No.	Description	Information
1	Function	<p>General function of the button: Workflow see the description below. Workflow List: Several workflows can be chosen which will be shown as a list on that button. A display name can also be defined for the button. Toggle: A toggle can be chosen later can be switched on and off with the button. (4.6.9 - Collection toggle) Station: A station can be chosen. The Occupation will be shown in the workflow view (grey: empty, white: occupied) and if the user level allows this, the occupation can be changed with this button (only ADMIN). The following menu allows to choose if the occupation should be changed to empty only or occupied only or if both should be possible. Station Area: same as station above, but for station areas or single stations of a station area</p>
2	Workflow	The desired workflow can be chosen via a drop-down menu. The name of the button can be changed, the countdown for cancelling the order can be adjusted, and a customized image for the button can be uploaded.
3	Variables	The variables for the workflow can be defined here.

Create new Schedule

Schedules can be created to automatically start generic workflows.



Description	Information
Schedule Name	The name of the schedule can be changed by clicking on the pen next to it.
Status	Enable / disable the schedule
Workflow	Choose any workflow from a drop-down menu
Variables	Set the required Variables for the workflow
Interval	Set the time between creating another order with that same workflow (with same variables). e.g. if set to 30 seconds, the system tries to create another order every 30 seconds
Concurrency	How many orders with this workflow can be created simultaneously
Concurrency Border	<ul style="list-style-type: none"> PENDING: If interval time has passed and there are fewer orders pending than adjusted in concurrency, another order can be created immediately. PICKUP/DROP: If interval time has passed and there are fewer orders active than adjusted in concurrency, another order can be created, after the first pickup/drop of the previous order. PICKUP/DROP LEFT: If interval time has passed and there are fewer orders active than adjusted in concurrency, another order can be created, after the AGILOX vehicle has left the first pickup/drop station of the previous order. ACTIVE: If interval time has passed and there are fewer orders <u>active</u> than adjusted in concurrency, another order can be created.
Week Schedule	Set timeframes for the order schedule
Trigger Condition	Conditions for creating orders can be defined. If the condition is TRUE, the order will be created but not started until the start condition is TRUE. The drop-down menu allows to select states of AGILOX IOs or toggles. Additionally, specific conditions can be defined when selecting "Condition" again, after selecting the state/toggle. The selected item will then be prefilled in the box below and conditions as in 4.7.5 - Workflow Condition Object can be defined.

Description	Information
Start Condition	Conditions for starting the workflow can be defined. The drop-down menu allows to select states of AGILOX IOs or toggles. Additionally, specific conditions can be defined when selecting "Condition" again, after selecting the state/toggle. The selected item will then be prefilled in the box below and conditions as in 4.7.5 - Workflow Condition Object can be defined.

4.3.7.4 Maintenance Mode (Vehicle Maintenance)

The maintenance mode of AGILOX vehicles allows an unlimited use of all functionalities of the vehicle as well as directly controlling individual components (motors, lifting equipment). To be able to activate the maintenance operation mode, the following conditions must be met:

- The operation mode key switch on the AGILOX vehicle must be in the left-hand position
- Link only via internal AGILOX vehicle WLAN (see [4.1 - Maintenance Operation Mode](#))
- The Acknowledgment pushbutton must be pressed for each step:
When performing a movement, one hand has to click and stay on the touch screen. The Acknowledgment pushbutton lights up and needs to be pressed with the other hand.
→ AGILOX vehicle is doing the maintenance move.
If one of the two hands got released or the network connection got lost, the other hand needs to let go as well and both hands have to start from the beginning again.



NOTICE

Risk of damage to property

Improper use (e.g. controlling just one of the two lifting sides) may damage AGILOX vehicles.

- This functionality should only be used by persons with adequate skills / training

Maintenance mode for the lifting equipment



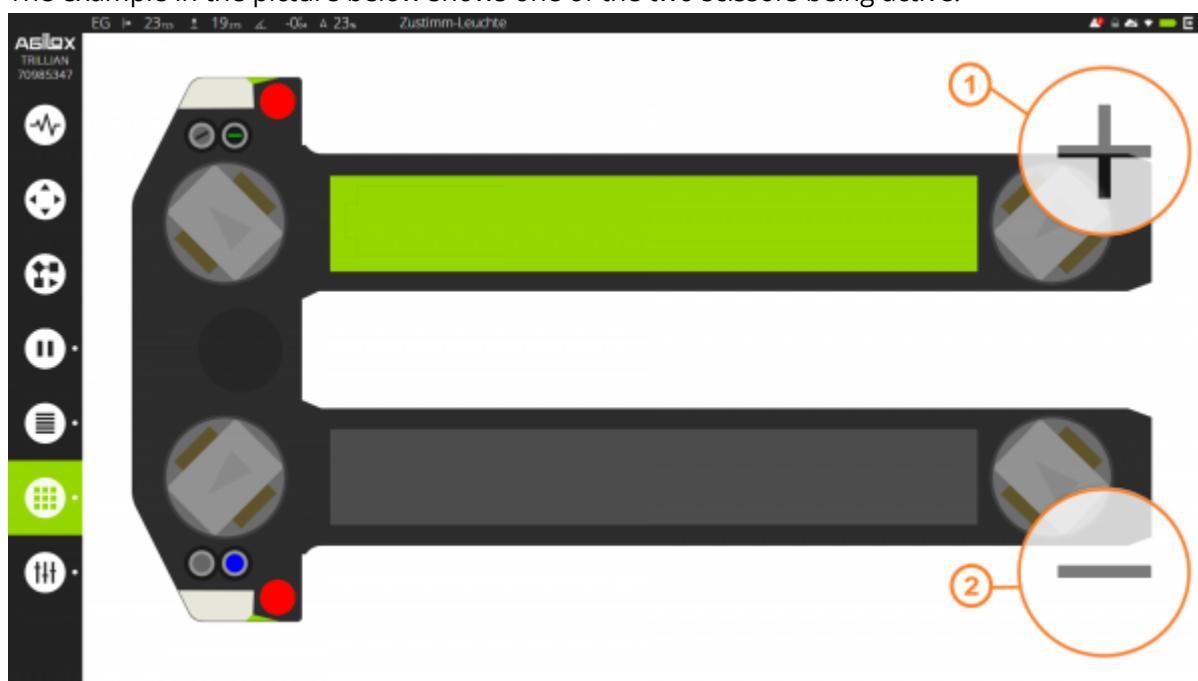
WARNING

The lifting equipment of an AGILOX vehicle may give rise to a risk of crushing. As no safety features are active, severe injuries may ensue.

- Be sure to use safety shoes in danger areas
- No persons must be present near the AGILOX vehicle while its lifting equipment is in motion

One or both scissors of the lifting equipment can be controlled by clicking on them.

The example in the picture below shows one of the two scissors being active.



NOTICE

Risk of damage to property

Raising the scissors above their upper stop or lowering them below their lower stop may result in the mechanical destruction of vehicle parts.

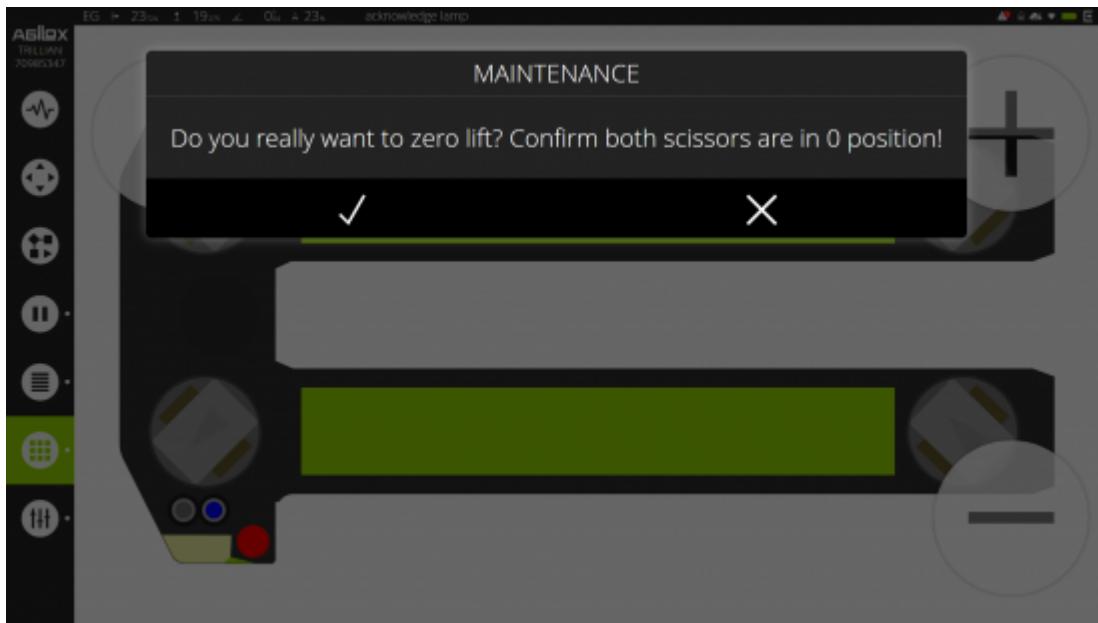
- Only raise/lower the scissors as far as they can go

- Press the Plus symbol (1) to raise the scissor lift(s)
- Press the Minus symbol (2) to lower the scissor lift(s)

Please note that the Acknowledgment pushbutton must be pressed the entire time!

Adjust Zero Position for Lift

1. Both forks must be in the same height above the floor
2. Move forks to the lowest position in manual mode
3. Switch to maintenance mode and activate both scissors
4. Set the zero position for the lifting mechanisms by clicking on the displayed '0'



NOTICE

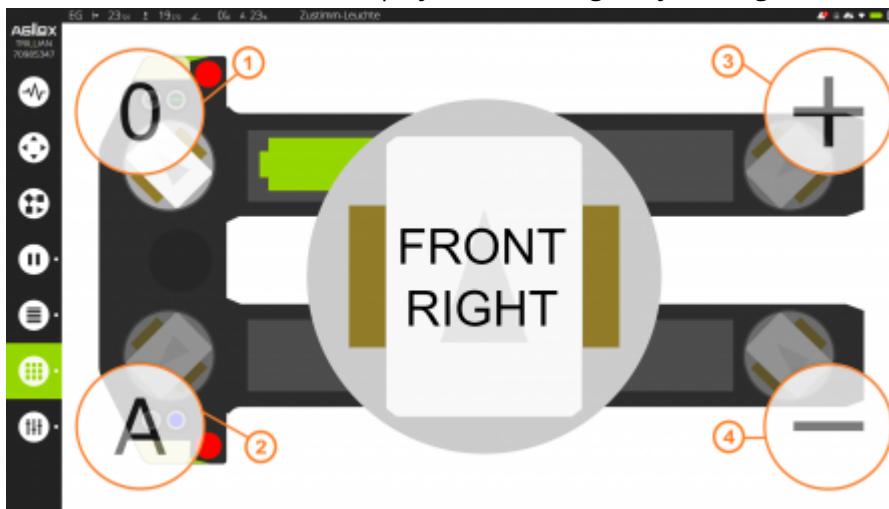
Risk of damage to property

An incorrect lifting starting level may impair the correct insertion of the vehicle fork into transport structures such as pallets, resulting in hazardous situations.

- Be mindful of the vehicle's lifting starting level

Maintenance Mode for the Drive Unit

The four drive units can be displayed and enlarged by clicking on one of them.



One or both drive wheels can be activated by clicking and controlled either individually or simultaneously.

- Clicking on the Plus symbol (3) will turn one or both wheels forwards
- Clicking on the Minus symbol (4) will turn one or both wheels backwards
- The '0' symbol (1) can be used to set the initial rotation angle of the drive unit.
- The 'A' symbol (2) can be used to repeat the Autosetup procedure for the drive unit.

Set Initial Rotation Angle for the Drive Unit

To set the initial rotation angle, i.e. the 0° angle for traveling in a straight line, follow these steps:

1. Make sure that the AGILOX vehicle rests on a stable surface.
2. Secure the drive unit by means of a plug-in bolt.



3. Press the '0'-Symbol



NOTICE

Risk of damage to property

Incorrect parameterization of the initial rotation angle may lead to malfunctions and mechanic damage.

- The AGILOX vehicle must rest on a stable, level surface during parameterization

Autosetup for Drive Unit

During the autosetup process, several testing and measuring procedures will be initiated to automatically determine and save the optimum parameters for the drive unit.

To perform an Autosetup, follow these steps:

1. The AGILOX vehicle must be in an elevated position, with the drive units safely secured against dropping out. Both wheels must be able to turn freely.
2. Select drive unit and press the 'A'-Symbol
3. Confirm warning that vehicle needs to be in elevated position.
4. Press the Acknowledgment pushbutton the entire time until its light goes off and the autosetup is completed.

The navigation scanner on top of the AGILOX vehicle will show 'Autosetup'.



NOTICE

Risk of damage to property

An incorrect Autosetup procedure may lead to malfunctions and mechanic damage.

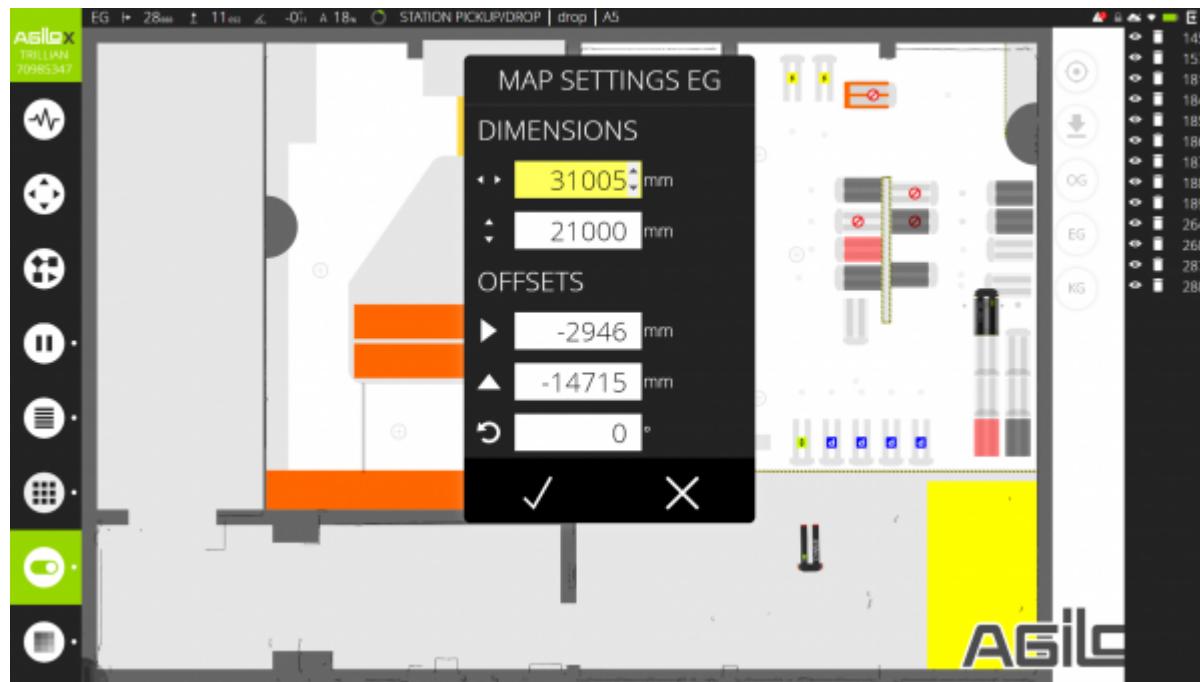
- AGILOX vehicle must be in an elevated position, with the drive units safely secured against dropping out
- Both wheels must be able to turn freely

4.3.7.5 AGILOX IO

The screenshot shows the AGILOX IO test interface. At the top, there is a header with vehicle information (EG, address, version, etc.) and a toolbar with various icons. Below the header, the title "AGILOX IO test" is displayed, along with the address "tcp://10.0.55.201:8001", version "1", and status "ALL OK (09:03:34)".
INPUTS: A row of eight buttons labeled 0 through 7. Buttons 0, 1, and 2 are highlighted in green, while others are dark grey.
OUTPUTS: A row of eight buttons labeled 0 through 7. Buttons 0, 1, and 2 are highlighted in green, while others are dark grey. Below each button, there is descriptive text: button 0 is "timer 0 set 8 unset 16"; button 1 is "timer 1 set 9 unset 17"; button 2 is "timer 2 set 10 unset 18"; button 3 is "timer 3 set 11 unset 19"; button 4 is "timer 4 set 12 unset 20"; button 5 is "timer 5 set 13 unset 21"; button 6 is "timer 6 set 14 unset 22"; button 7 is "timer 7 set 15 unset 23".
At the bottom left, there is a section titled "NEW AGILOX IO" with a "CREATE" button.

See Operation Manual AGILOX IO for detailed information.

4.3.7.6 Map Settings



Map parameters such as size or rotation can be adjusted under 'Map Settings'. When planning to expand the area of movement of an AGILOX vehicle (e.g. when an extension is added to a factory building), the map needs to be expanded as a first step as well.

In case of any such expansions, the map's size in millimeters, the offset of its point of origin as well as the rotation of the map need to be modified in the map settings.



INFORMATION

These settings will be adjusted by AGILOX North America, Inc. or its system partner upon initial commissioning in a project-specific way.



WARNING

Risk of injury

Any modifications will affect the entire UNION and may permanently impair/disrupt its progress/function.

- Only implement such modifications after adequate training

4.3.7.7 Switching off the AGILOX Vehicle (Shutdown AGILOX)

The AGILOX vehicle will be shut down immediately.



INFORMATION

The AGILOX vehicle can only be restarted by means of the main breaker of the AGILOX vehicle.

4.4 Teach Contours

The instructions and requirements for Area Mapping and Acquisition in [Operation Manual 2.10.5 - AGILOX vehicle - Area Mapping and Aquisition](#) must be complied with at any time.

Before the system can be commissioned, system operators have to provide adequate areas of movement in conformity with specific requirements. Special attention should be paid to the minimum width of areas of movement as well as to the floor space requirements at load acquisition, load depositing and load transfer stations. As regards the design of areas of movement, please refer to [Operation Manual ANNEX IV - Operating Areas](#) and [Operation Manual 2.3.4 - Floor Conditions of Area of Movement](#) for a specification of the applicable framework conditions.

In case of areas not in conformity with contour requirements, for example spaces with large wood paneling, system operators must get in touch with the manufacturer in order to coordinate measures designed to improve the visual characteristics of contours at the height of the vehicle's navigation scanner.



NOTICE

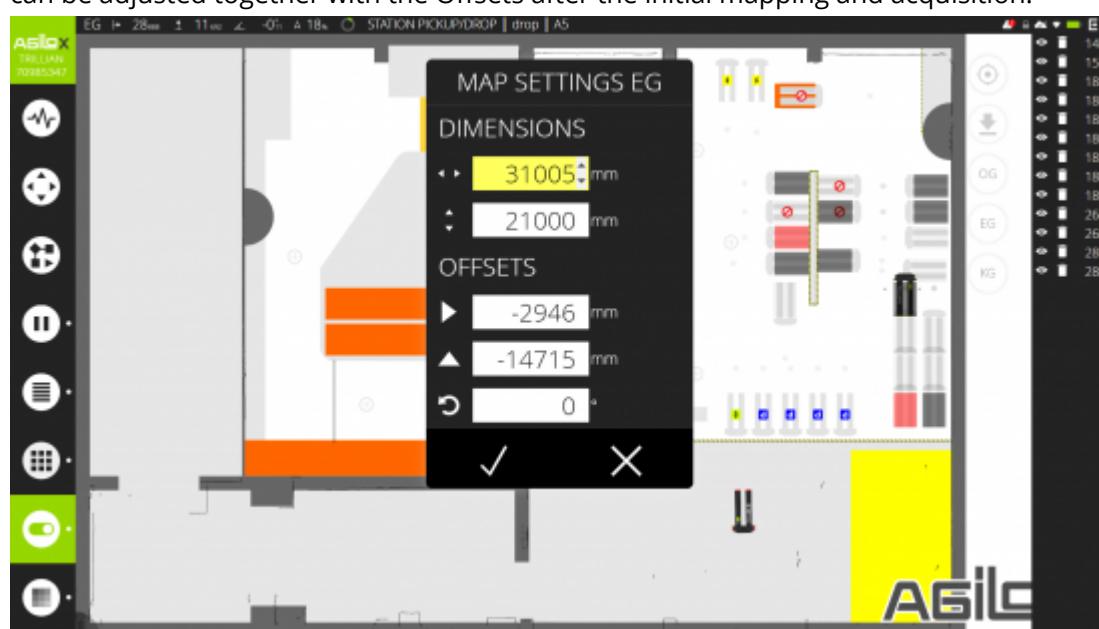
System operators always bear the sole responsibility for the ultimate safety concept. We recommend clarifying any issues in cooperation with authorities and bodies such as employers' liability insurance associations, government accident insurance agencies, the Works Council, the in-house health and safety department or the fire prevention department well in advance.

4.4.1 Initial Mapping and Acquisition

Upon commissioning, the first AGILOX vehicle will prepare a map of its surroundings (contours) and make it available to all the other AGILOX vehicles. The map can be edited and adjusted using a vector-drawing program such as Inkscape (see [4.5 - Map](#)).

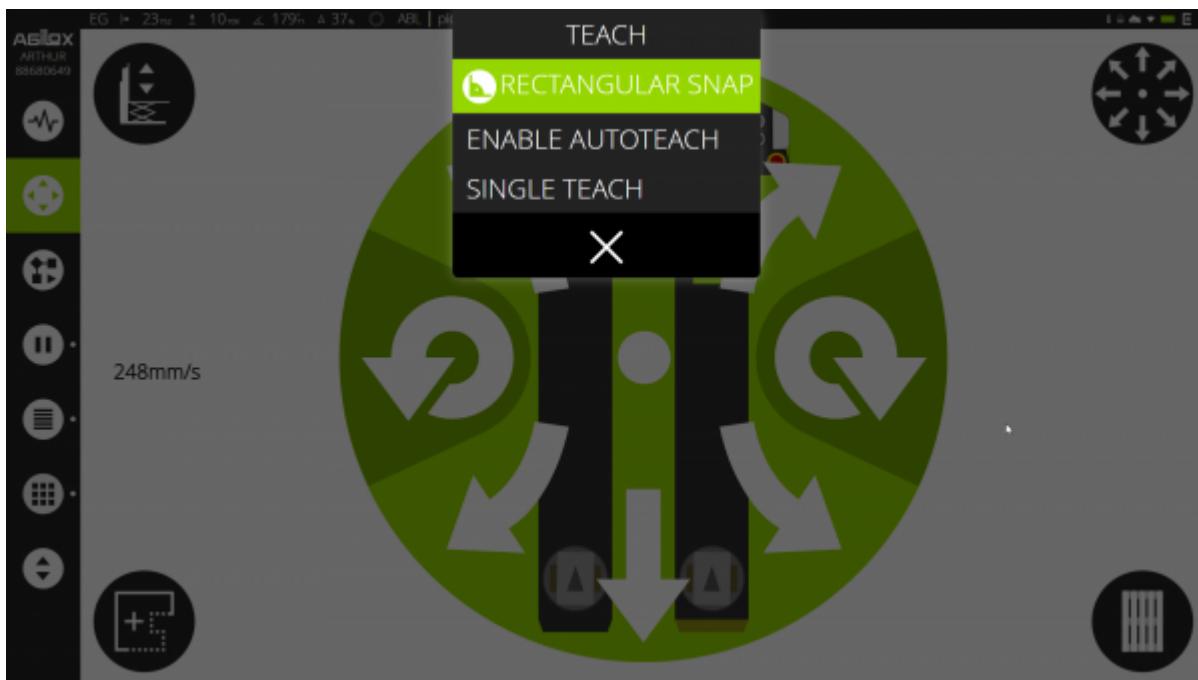
4.4.1.1 Map Settings

If there is no map existing yet, the AGILOX vehicle is in the center of the screen and assumes a map of 50m x 50m. If the actual room size differs a lot from this assumption, the map settings (room size) can be adjusted to the actual room size. Enter the dimensions (width / length) and confirm. Otherwise the room dimensions can be adjusted together with the Offsets after the initial mapping and acquisition.

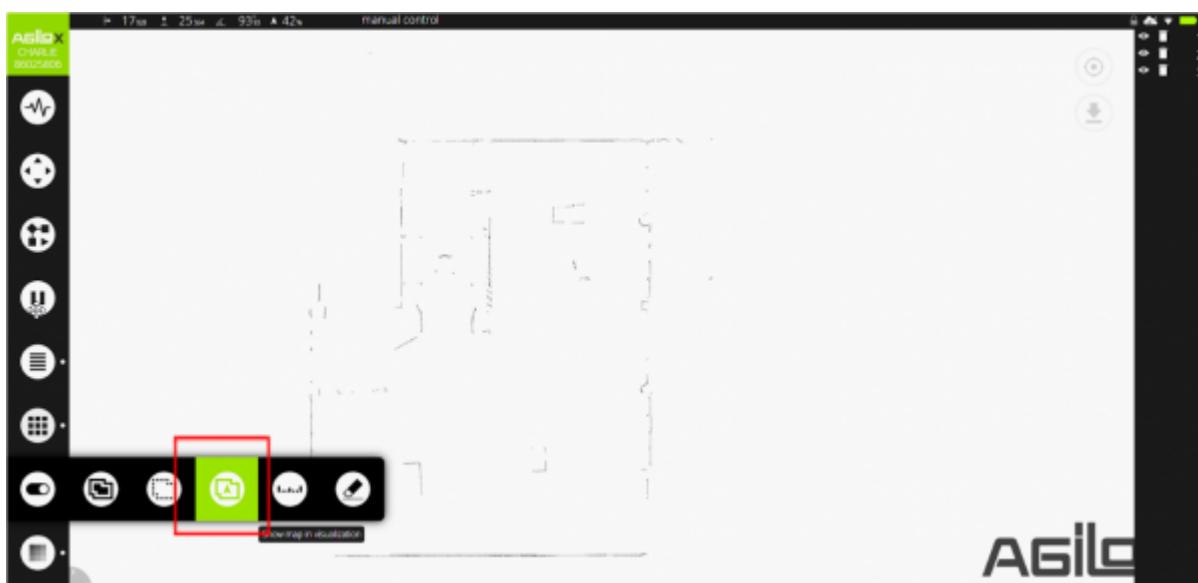


4.4.1.2 Autoteach of contours

1. Define travel paths for the AGILOX vehicle.
2. Define the later map orientation. Which side (wall) should later be up in visualization?
3. Place the AGILOX vehicle in manual mode approximately in the center of the area of movement, about 90°C to the wall that will later be the upper wall on the map. (maybe use an extension gap of the floor for proper orientation)
4. Open manual operation menu in visualization (HMI). Click on "Add new contour from navigation-scanner". Activate "Rectangular snap" (green when activated).



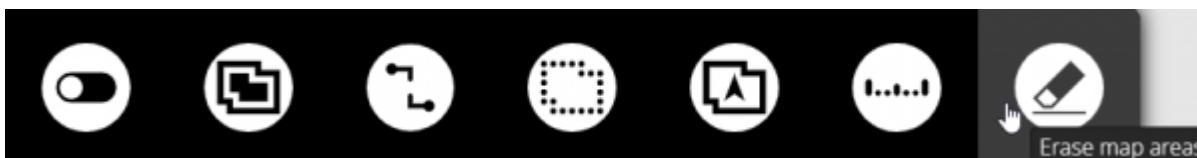
5. Click on "Single-Teach". The vehicle emits an audible signal when the teach is finished.
6. Now the detected contours can be shown in "Visualization Toggles" - "Show map in visualization".



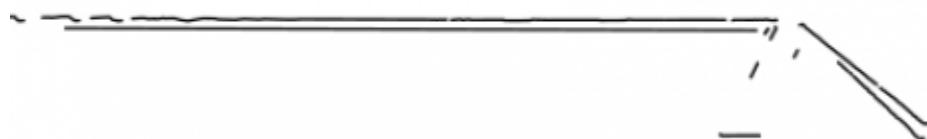
7. By pushing CTRL and SHIFT on the keyboard the mouse is replaced by crosshairs and the contours and their alignment can be checked. If necessary the contours can be deleted on the top right, the vehicle can be repositioned and a single teach can be repeated, to achieve a better alignment of the contours.
8. Once the alignment of the first contours is ok, leave "Rectangular snap" activated and click on "Enable Autoteach". The bottom left symbol turns blue when autoteach is enabled.



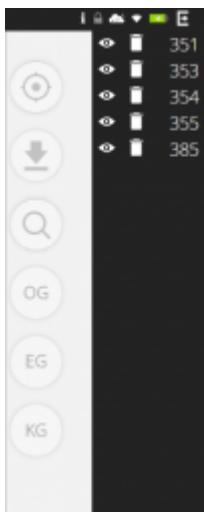
9. Drive along the future travel paths of the AGILOX vehicle in manual operation mode. The AGILOX vehicle adjusts its speed and stops automatically for another acquisition of contours. Avoid teaching in a circle and rather teach in a star shape to avoid offsets at the end of big circles.
10. The navigation scanner detects static and dynamic contours. Dynamic contours (such as movable or temporary items) can be deleted by using the delete function "Erase map area".



11. Parallel contours with a small distance to each other should be avoided as these might cause positioning problems later.



12. The created map can be composed of several single scans that can be shown or hidden on the top right. The overall map is composed of these scans.



13. Once the teaching is finished, deactivate "Enable Autoteach" and "Rectangular snap" in manual operation menu again.
14. If the contours are too less in some areas, a "single teach" can be made.
15. The room dimensions and offsets can now be adjusted to the contours. The ruler tool (Visualization Toggles - Ruler tool) in visualization can be useful to measure the correct offset value.
16. The finished navigation map can be downloaded by clicking on the download button and can be opened and edited using a vector-drawing program such as Inkscape (see [05-map](#)).

4.5 Single Teach

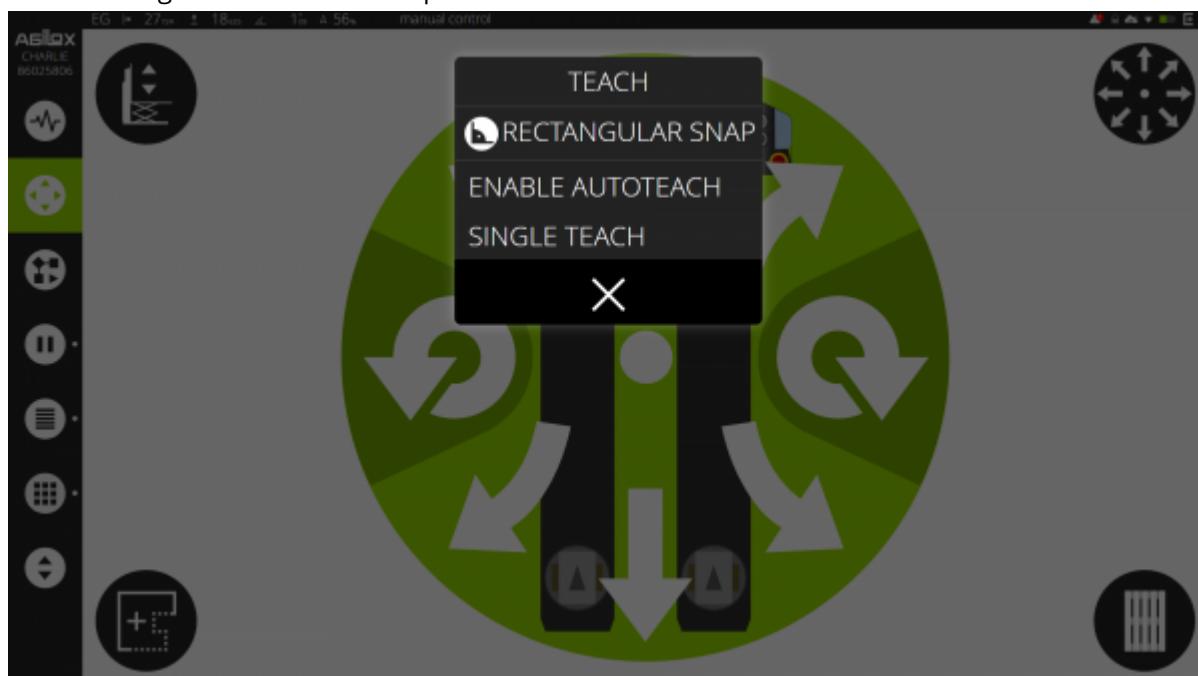


CAUTION

Acquisition of new contours may only be carried out when the AGILOX vehicle knows its current position and its position also matches the current physical position. Otherwise wrong contours are taught.

The single teach function, records the measurement data of the navigation laser scanner at the current position to create map contours. The navigation map can thus be selectively improved at individual positions. This makes sense especially around conveyor stations, in door areas or areas between two different rooms. Before a single teach is carried out, the positioning of the AGILOX vehicle must be checked. To do this, use the visualization toggles to show the scan data and the navigation map in visualization and check that the green lines match the black lines.

Start the single teach in manual operation mode.



The vehicle emits an audible signal once the teach is finished.

4.5 Map

To be able to find routes through the area of movement independently, AGILOX vehicles have to be provided with a map of the area during commissioning.

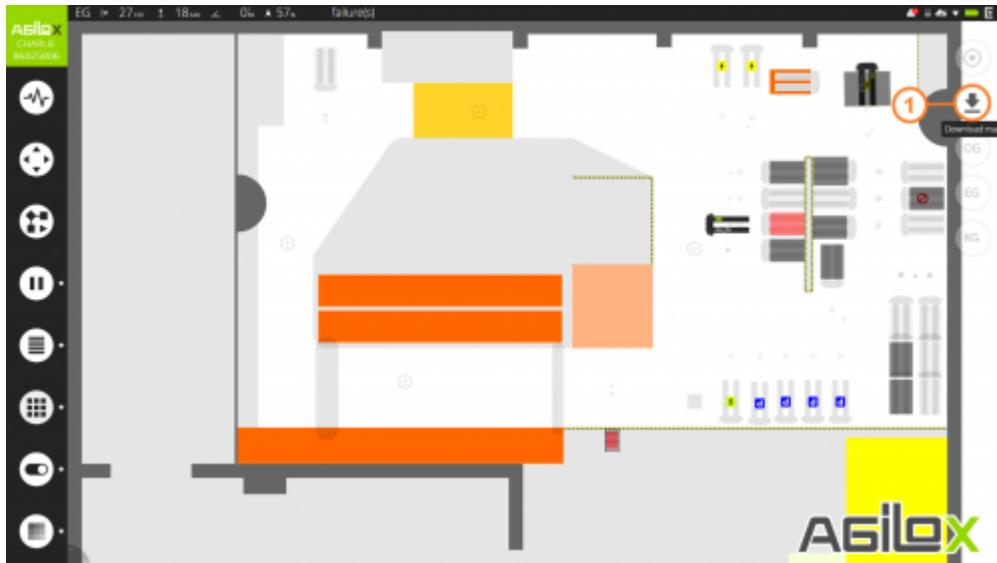
On the basis of this map, the system later creates the route map.



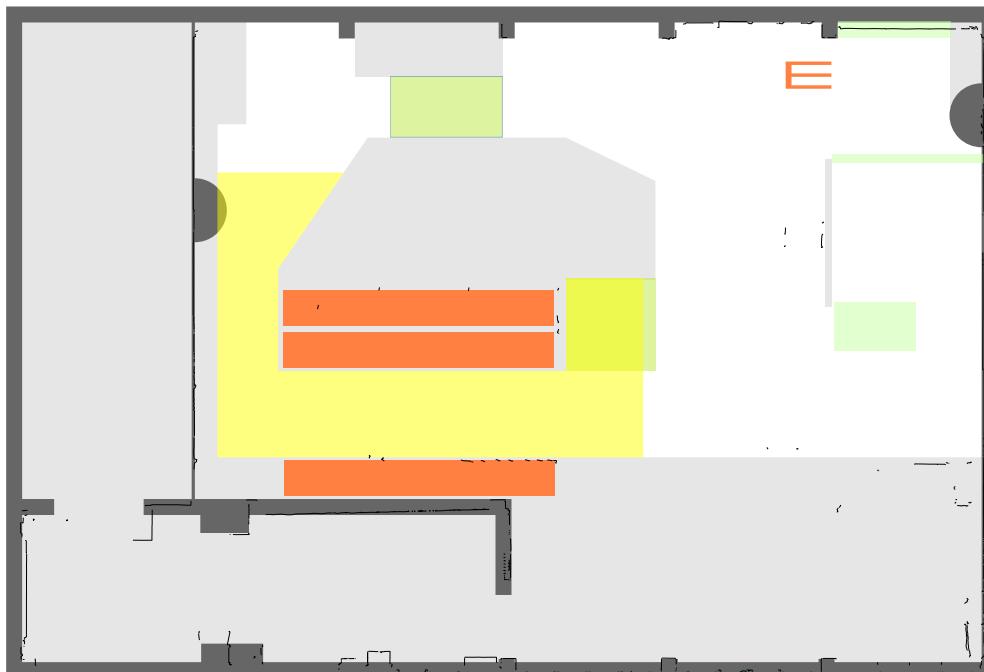
The reference point is always the centre of the vehicle.

Color	Description
white	Driving areas, on these areas routes are possible without restrictions.
green	Tracks (see 4.5.4.1 - Flags)
blue	Avoidance (see 4.5.4.1 - Flags)
yellow	Stations, this area may only be used for load handling, routes via stations are only possible if a track has been defined
orange	Fringe areas, are avoided on the route, if this is possible
red	Areas around prohibited areas, these areas are not driven over by the vehicle to avoid collisions.
black	Prohibited areas, no driving allowed (Walls, Columns, etc.)
other colors	See Tooltips for more information about those areas

To change general map settings or create a new SVG element, the map needs to be downloaded first. This can be done in the visualization window.



It is recommended to use Inkscape ([Inkscape Version 0.92.3](#)) to edit the map.



After editing, the new map can be uploaded by drag&drop the svg-file to the AGILOX visualization window.



NOTICE

Risk of damage to property

The entire UNION as well as the whole process can be permanently impaired.

- Only use this functionality after adequate training

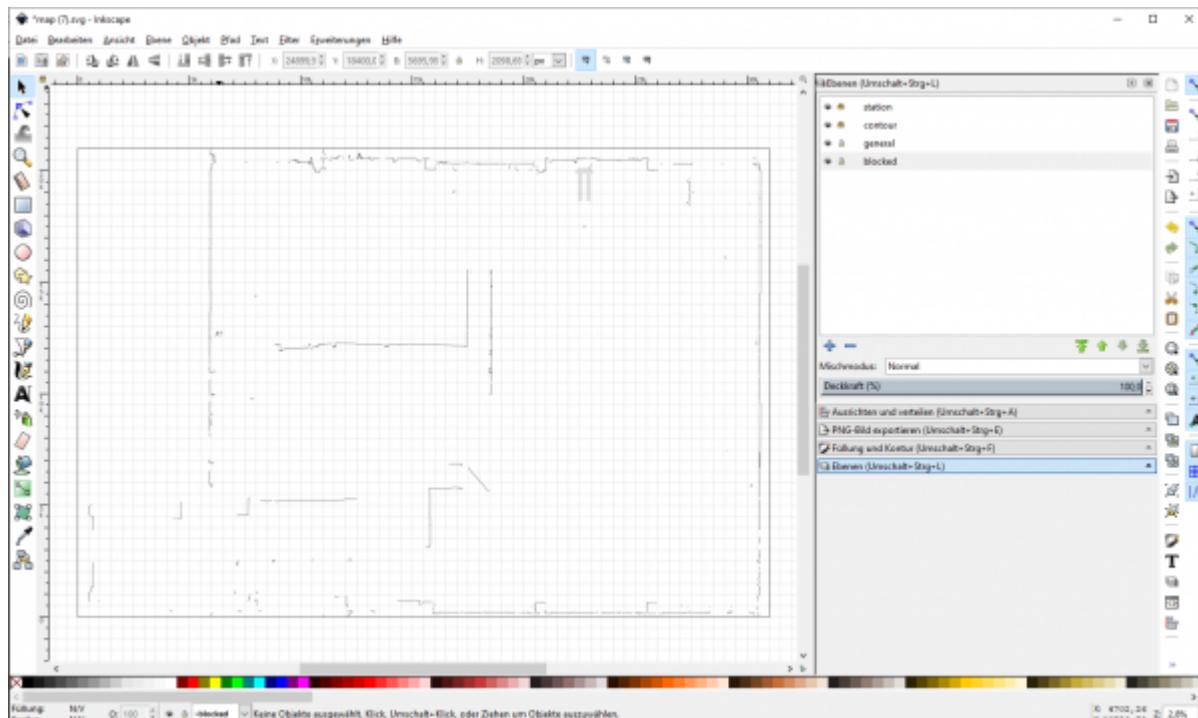


INFORMATION

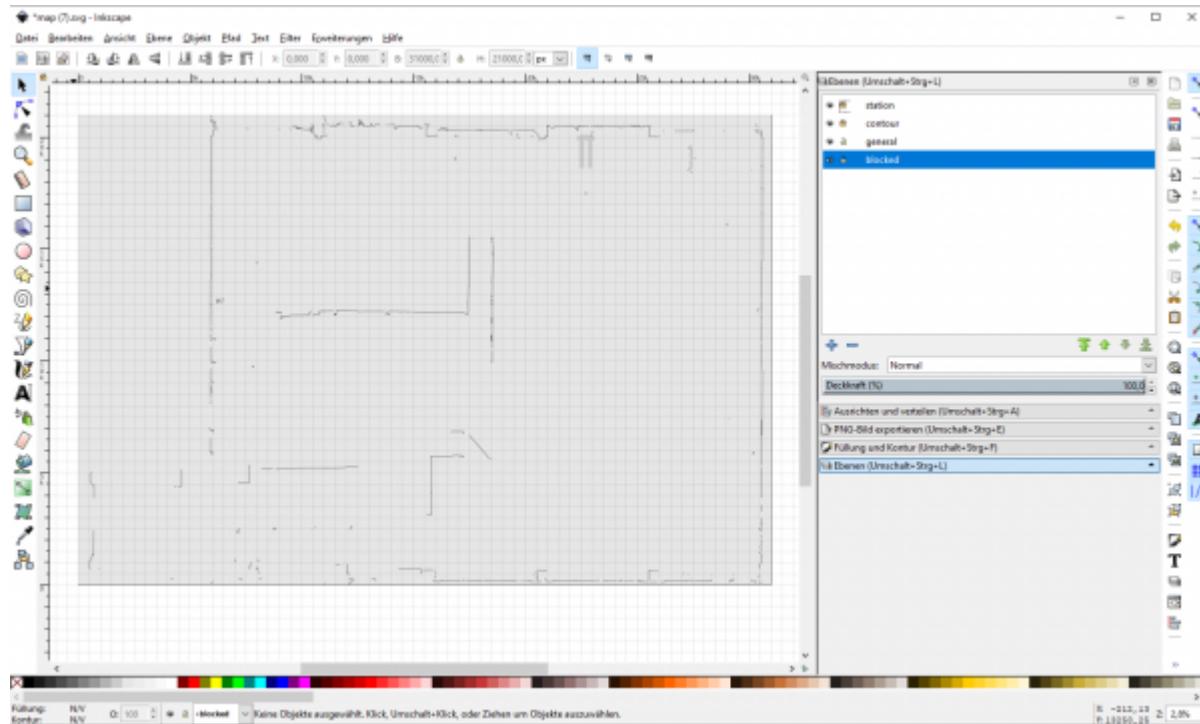
The manufacturer recommends performing a backup before modifying the map. See 'Backing up and Restoring' the AGILOX vehicle ([4.3.7.1 - System Settings](#)).

4.5.1 Initial Definition of Driving Paths

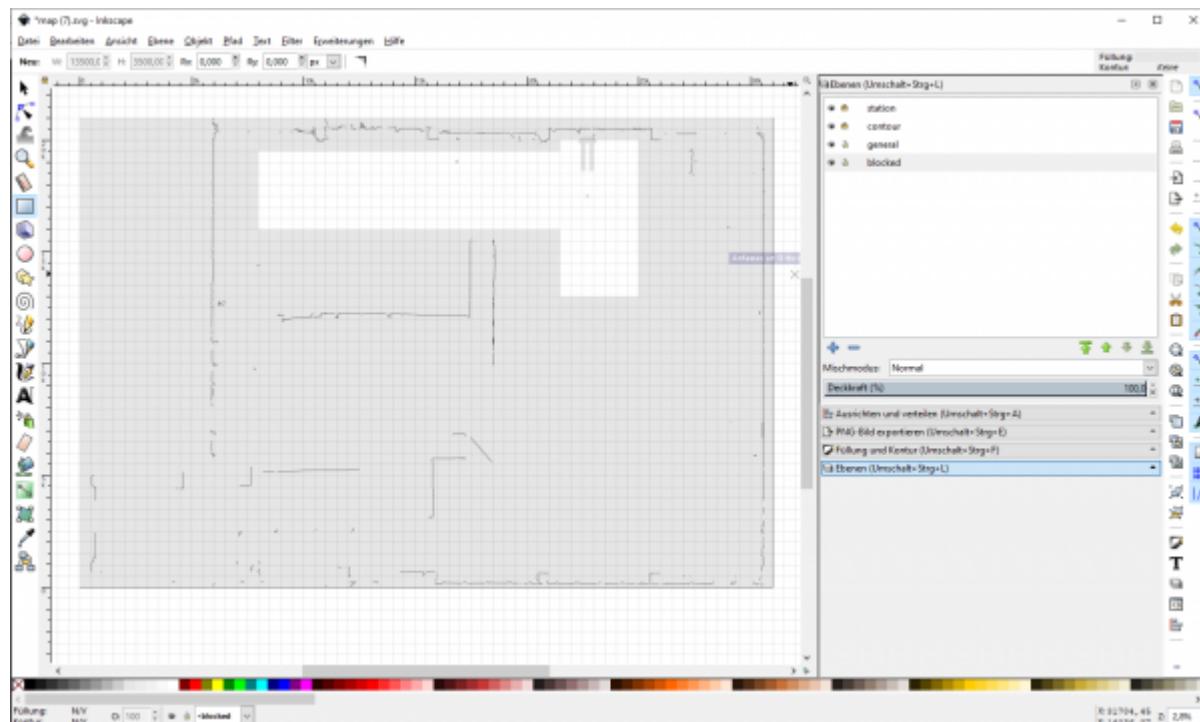
The downloaded map can be opened with the vector orientated drawing program "Inkscape".([Inkscape Version 0.92.3](#))



The next step is to draw a rectangle over the whole area in the layer "blocked" (color code e6e6e6ff).



Over this rectangle, the driving paths can be drawn. By using the Shift key both shapes can be marked and using "CTRL" + "-" the shape can be removed from the blocked area. Shapes can also be added to the blocked area by using "CTR" + "+".



4.5.2 Map layers

The maps consists of several layers, which can be shown and blocked individually.

The sheet size is already adjusted to the contours.

The grid-size is corresponding to 50cm x 50cm when zoomed out and 10cm x 10cm when zoomed in.

The suggested [Color schema](#) for room elements, tracks, etc. is documented separately.

4.5.2.1 Station

This layer contains all stations and waypoints.

Changes will have no effect, the layer is used as orientation guide in an empty map.

4.5.2.2 Contour

This layer shows all navigation points and merely serves as a guide for drawing the map.

Changes will have no effect, the layer will be ignored by AGILOX.

4.5.2.3 Blocked

This layer defines areas AGILOX is normally not allowed to drive on.

Room objects like walls or pillars should also be placed in this layer.

It's also a common practice to at first block the whole map (by drawing a gray area as big as the map itself) and then draw white areas for corridors/areas AGILOX is allowed to drive on.

4.5.2.4 General

All other SVG elements are within this layer. Any shape can be drawn and the attributes can than be defined via the HMI by double-clicking the SVG element, see information below.

If the element has no attributes, it is ignored and won't appear as an obstacle to the AGILOX vehicle. This function can be used for room labeling (e.g. texts on a driveway).

4.5.3 Color Schema

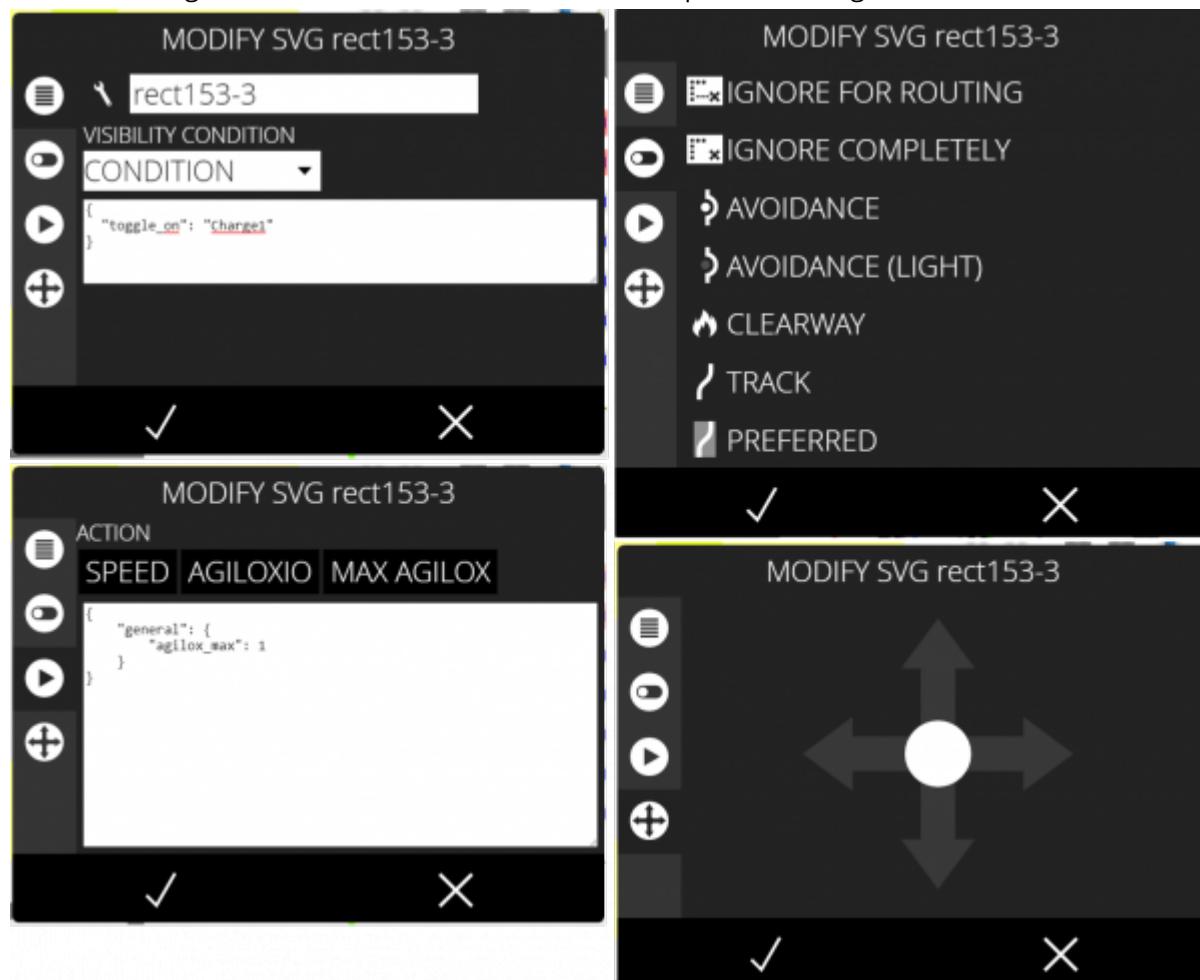
Objects	Color Code	
Room (<i>walls, pillars, room-elements</i>)	666666ff	
Blocked (<i>impassable areas, blocked areas</i>)	e6e6e6ff	
Track	99ff55ff	
Direction, Free	c7ffa180	
Avoidance, Clearway	bee84280	
Action	ffff0080	
Surveillance	ffce0d80	

4.5.4 SVG Elements

Elements can have different features, as described below. To create such elements, download the map and draw an element in the map layer "general" (e.g. by using "Inkscape" graphics program). Several elements that serve the same purpose or have the same content should be combined into one element (mark all areas and combine them with "CTRL "+""). In order not to impair performance, no more than 50 different elements should be present in one map.

After that upload the map to the hmi.

Double-clicking the SVG element in HMI visualization opens following window:



Description	Information
Modify SVG Element <Name>	Modify Action selected element The selected svg element is marked with a dashed line.
Element ID	Set element ID, has to be unique within the map The id can be used e.g. in conditions (see 4.7.5 - Workflow Condition Object) or in other action areas (see 4.5.5.2 - Conditional Actions)
Visibility Condition	A state of an AGILOX IO, a toggle or any condition can be defined. If set, the SVG Element is only visible, if the state/toggle/condition is true. AGILOX vehicle still performs the defined action of the element, but the element is invisible. e.g. for texts, traffic lights,
Flags	Set flags for action area, multiple choice possible. See description below.

Description	Information
Action	Action Settings according to 4.5.5 - Action Settings
Direction	Set Directions according to 4.5.4.2 - Direction . One click on an arrow makes the direction a hard direction (white), another click on it makes it a soft direction (gray).

4.5.4.1 Flags

IGNORE FOR ROUTING

The AGILOX vehicle will ignore all detected obstacles for navigation and routing. This is e.g. needed for high speed doors opening automatically when AGILOX is approaching them. Without a free area those doors would appear as barrier to AGILOX and it would not even try to find a route through.

IGNORE COMPLETELY

The AGILOX vehicle ignores all detected obstacles on this area. All areas with this flag will be completely freed from scan data.

The rope of a high speed door for example can be ignored and driven through with such an area. However this function does not work on station areas, as for station handling a completely different field monitoring takes place, which has no map information at all. Stations always must be completely free of any permanent obstacles.

Avoidance

The AGILOX vehicle will avoid driving over Avoidance Areas if other routes are possible and will only drive there if necessary. The AGILOX vehicle will leave the area again as soon as possible.

For example if there is an obstacle on the main route, it will stop and wait for a specific time first before driving on the avoidance area to bypass the obstacle.

Avoidance (light)

This is e.g. used for walkways. The AGILOX vehicle is basically allowed to drive there, but should avoid it as far as possible.

Clearway

In the event of fire, AGILOX will move away from areas defined as "clearway".

Track

By drawing a track you are forcing a passable area/line even in too narrow map regions or through blocked areas.

Additionally, tracks will always be preferred for routing. With this option desired routes can be created, which are used as far as possible.

Preferred

Preferred is to define preferred routes, but does not clear blocked areas or allows using red/orange areas.

Weight Limit 1500kg

AGILOX vehicles with a weight >1500kg (e.g. AGILOX OCF) must not drive on this area.

4.5.4.2 Direction



INFORMATION

From Software Version [20180608] the direction parameters have changed as follows and need to be adjusted accordingly, when the software gets updated.

It is possible to define allowed directions for this SVG element. If a direction is defined, it means that every direction is allowed, except the one against the defined direction.

Multiple directions can be defined. Each defined direction restricts the routing, as the AGILOX vehicle is not allowed to go against the defined directions.

There are hard directions, that must be followed in any case, and soft directions, which merely result in higher costs and can be ignored in exceptional cases.

Clicking a direction once makes it a hard direction (white arrow), clicking it again makes it a soft direction (gray arrow).

4.5.5 Action Settings

One or more actions can be defined, that are performed when the AGILOX vehicle is within the SVG element - to give a few examples: slow down on busy junctions, open a high-speed door, issue a warning sound, ask a web service for clearance, etc.

4.5.5.1 General Actions

Actions that should always be executed when the AGILOX vehicle is on that action area are defined in "general".

```
{
  "general": {
    "speed_max": 500,           //limit maximum speed to 500mm/s
    "agilox_max": 1,            // only 1 AGILOX vehicle at a time is allowed in this area
    "agilox_max_distance": 5000, // other AGILOX vehicles have to wait 5000 mm away from this area
    "surveillance_distance": 12000,
    "route_distance": 2000,
    "obstruction": 500,
    "no_rotate": true,
    "start_on_protective": true,
    "led": [
      [<red>, <green>, <blue>, <mode>], // left led
      [<red>, <green>, <blue>, <mode>] // right led
    ],
    "audio": 7,
    "volume": 80,
    "agiloxio": {
      "test": {
        "output": {             // set output 0 and 1
          "0": 1,
          "1": 1
        }
      }
    },
    "request": {
      "uri": "http://192.168.0.100/wms/controlDoor",
      "post": {
        "openDoor": true
      }
    }
  }
}
```

Attribut	Type	Description
speed_max	Number	Limit maximum speed to the defined value in mm/s.
agilox_max	Number	There is only the defined amount of AGILOX vehicles allowed on that area.
agilox_max_distance	Number	If the maximum AGILOX vehicle count defined for this area is reached already and another AGILOX vehicle wants to enter the area, "agilox_max_distance" defines where (in which distance to the area in mm) the AGILOX vehicle has to stop and wait for an AGILOX vehicle to leave the area.
surveillance_distance	Number	<p>Enlarge Monitoring Area: For very big "surveillance" or "enter" areas, the area monitored by the AGILOX vehicle can be enlarged, in order to be able to monitor the whole area (Default: 7m). Therefore an additional action area has to be defined next to the "surveillance" or "enter" area with the attribute "surveillance_distance". When the AGILOX vehicle is on this area, the monitoring radius is enlarged to the defined value in mm.</p> <p>ATTENTION: Depending on the floor properties, with big radii the floor could be detected as an obstacle!</p>

Attribut	Type	Description
route_distance	Number	This value defines the distance of the route from AGILOX vehicle's center to the area when the action is already executed before the AGILOX vehicle has even entered the area. This means actions of an area can be executed before the AGILOX vehicle has entered the area, if the area is located on the vehicle's route.
obstruction	Number	The default distance to obstacles is reduced to the defined value in mm (default 1500mm). This can be useful for gradients of the floor in the area of movement as the AGILOX vehicle would detect the floor as an obstacle. Obstruction can also be set to 0, to ignore obstacles on this action area completely. The defined obstruction distance is only valid as long as the AGILOX vehicle is on the action area with its center point. WARNING: Risk of collision!
no_rotate	bool	If this is set to true, any rotation during driving can be avoided on this area. If the route faces backwards, the AGILOX will drive backwards instead of turning first. Driving dynamic is the same as driving forwards, but slower. Attention: If the route goes 90° away from the vehicle, the vehicle will drive a curve and this might need a lot of space! This function only affects driving, it doesn't change anything for prepositions, narrow aisle or station handling.
start_on_protective	bool	If this is set to true, the start button (white) is ignited on defined areas, when protective field gets triggered. The AGILOX vehicle stops and waits for the start button to be pressed.
led	Array	The vehicle's LEDs can be controlled. See detailed information below.
audio	Number	The vehicle emits the audio file with the defined id. For available audio files see Operation Manual ANNEX I - Audio Signals
volume	Number	Audio volume in %, min. 50%, max 100%
agiloxio	Object	Set Output on AGILOX IO. See also 4.7.9 - AGILOX IO Request Definition
request	Object	Start a Webservice request. This request will be posted every 500ms as long as the AGILOX vehicle is on the action area. 4.7.8 - Webservice Request

LED

```
{
  "general": {
    "led": [
      [<red>, <green>, <blue>, <mode>], // led color intensity and mode of left led
      [<red>, <green>, <blue>, <mode>] // led color intensity and mode of right led
    ]
  }
}
```

led intensity:

Maximum light intensity	255
Half light intensity	128
Low light intensity	64

modes available:

LED_OFF	0
LED_ON	1
LED_BLINK - 30Hz	2
LED_FADE - 15Hz	3
LED_BLINK_FAST - 60Hz	4
LED_FADE_FAST - 30Hz	5
LED_BLINK_ULTRAFAST - 120Hz	6

example:

```
{  
    "general": {  
        "led": [  
            [128, 128, 0, 2], // left led light yellow blinking  
            [255, 0, 0, 5]   // right led intensive red fast fading  
        ]  
    }  
}
```

4.5.5.2 Conditional Actions

Actions defined in "conditional" are only executed if the defined condition is met.

```
{  
    "conditional": { <condition object>, <action> }  
}
```

Example: Stop or Limit Speed by Request

```
{  
    "general": {  
        "speed_max": 1000  
    },  
    "conditional": {  
        "agiloxio": {  
            "agi_io1": {  
                "input": {  
                    "0": 1,  
                    "1": 1  
                }  
            },  
            "speed_max": 500  
        }  
    }  
}
```

Example: Stop if no communication to AGILOX IO

```
{  
    "conditional": {  
        "agiloxio_invalid": ["agiloxio1", "agiloxio2"],  
        "stop": true  
    }  
}
```

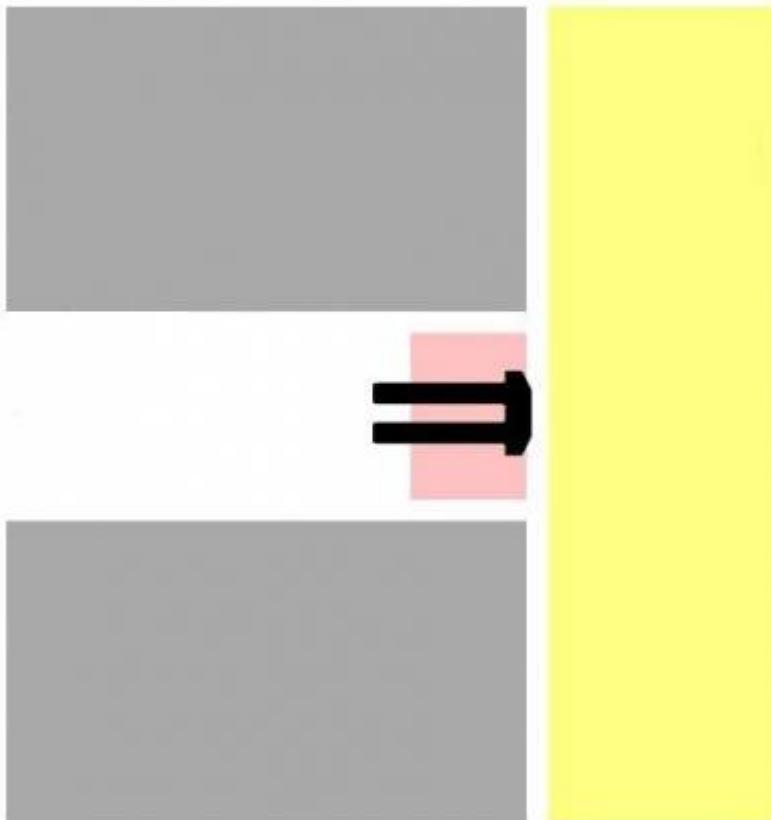
The condition is true, if the communication to one or more AGILOX IOs is not possible.

Example: Intersection with action areas

The condition "action_surveillance_count" can be used to respond to the number of scan points that hit an obstacle on another action-area.

For example, it is possible to define an area in front of an intersection (red in the picture below) on which the AGILOX vehicle will stop if an obstacle is detected on another, distant area (yellow in the picture).

Optionally, the condition "angle" can be used to limit the subsequent condition to a certain angle:

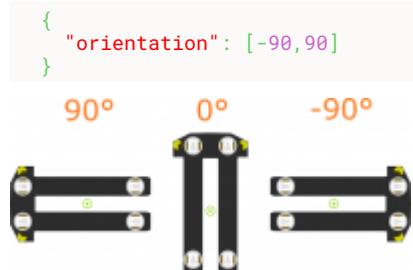


Red Action Area:

```
[  
  {  
    "general": {  
      "speed_max": 250  
    },  
    "conditional": {  
      "angle": {  
        "angle": -90000,  
        "tolerance": 2250  
      },  
      "action_surveillance_count": {  
        "actionid": ["entering1"], //Element ID of the action area that should be checked  
        "count": ">5"  
      },  
      "stop": true  
    }  
  }  
]
```

4.5.5.3 Orientation

An Action area can have the attribute "orientation" which defines an array of allowed orientations (in °) of the AGILOX vehicle when driving on this area.



The AGILOX vehicle drives up to 1m away from the area. If the angle difference is more than 5°, the vehicle stops and turns to the defined orientation. If the difference is less, the vehicle doesn't stop, but turns to the defined orientation while still driving. If the AGILOX vehicle is already on the area, it will also turn while still driving. It does not turn to an orientation more than 100° from its current orientation. The AGILOX vehicle will drive through the area with the defined orientation.

For safety reasons the AGILOX vehicle drives very slowly through such Orientation Areas.

4.5.5.4 Surveillance

Actions defined in "surveillance", are only executed if the AGILOX vehicle detects an obstacle within this SVG element. Foreign AGILOX vehicles detected on this area are ignored and do not count as obstacles.

Basically the same actions as in an action area can be defined.

Additionally, the action "no_station_leave" can be defined in this area, which prohibits the AGILOX vehicle to leave a station in the surveillance area, as long as it detects something there.

```
{
  "surveillance": {
    "no_station_leave": true
  }
}
```

4.5.5.5 Enter

Areas can be given a condition for the AGILOX vehicle to enter it. The AGILOX vehicle is then only allowed to enter the area, when the condition is met.

The condition is only checked when entering the area and has no influence to AGILOX vehicles that are already on the area.

Basically every condition can be defined, that can also be defined in workflows and make sense here.

```
{
  "enter": {
    "area_free": 3,
    "enter_distance": 500
  }
}
```

Additionally, the condition "area_free" can be used to define, how many scan points may be detected on the area maximum. In this case the condition is met, when the AGILOX vehicle detects less than three scan points on this area.

The condition of course only works within the monitoring radius of the AGILOX vehicle. (Default: 7m, see [4.5.5.1 - General Actions](#)). "enter_distance" defines a waiting position for the AGILOX vehicle (value in mm). The AGILOX waits there until the condition is met.

4.6 Collections

Collections can be viewed and changed with the system database editor ([4.3.7.2 - System Database Editor](#)).

4.6.1 agioloxio

See Operation Manual AGILOX IO for detailed information.

4.6.2 barrier

```
{
    "OB-Barrier": {
        "floor": 1,
        "role": 3,
        "auto": true,
        "barrier": [
            "Barrier 1": {
                "start": {
                    "x": 6110,
                    "y": 14000
                },
                "end": {
                    "x": 6120,
                    "y": 10000
                },
                "direction": 4
                "radius": 500
            },
            "Barrier 2": { <barrier object> },
            "Barrier 3": { <barrier object> },
            ...
        ],
        <condition 1>
        <condition 2>
        ...
    }
}
```

Attribute	Type	Required	Description
ID (here "OB-Barrier")	string	yes	unique identifier of the barrier/s
floor	number	no	The floor in which the barrier is located. (Default: 0) ATTENTION: This is not the "order" defined in collection "floor" but the order in which the floors are defined in the collection (starting with 0)!
auto	bool	no	This defines whether the barrier is switched on and off automatically or not. Automatic switching only works if a condition is defined and can be overridden by hmi's manual switching of barriers. Default: true.
role	number	no	This defines which users are allowed to manually switch the barrier on or off in hmi: 1... Worker + Leader + Admin 2... Leader + Admin 3... only Admin
barrier	object	yes	Barrier object. A barrier object can consist of several barriers.
Barrier ID (here "barrier 1")	string	yes	unique identifier of one specific barrier

Attribute	Type	Required	Description
start	object	yes	start point of the barrier with x and y coordinates. Start and Endpoint describe the line for the barrier.
end	object	yes	end point of the barrier with x and y coordinates. Start and Endpoint describe the line for the barrier.
direction	number	no	<p>The direction in which a barrier can be driven through can be defined.</p> <p>0 4 3 7 1 5 2 6</p> <p>0-3 are hard directions and 4-7 are soft directions, see 4.5.4.2 - Directions</p>
radius	number	no	Defines how hard a soft direction is. The bigger the radius, the worse is driving through it against its direction.
<condition>	object	no	<p>Conditions which must be met for the barrier to be "open". See examples below.</p> <p>Possible conditions are: "time", "agiloxio" and "toggle". (see also 4.7.5 - Workflow Condition Object)</p>

4.6.2.1 Examples

Simple barrier

```
{  
    "OB-Barrier": {  
        "floor": 1,  
        "role": 2,  
        "auto": false  
        "barrier": {  
            "Barrier 1": {  
                "start": {  
                    "x": 6110,  
                    "y": 14000  
                },  
                "end": {  
                    "x": 6120,  
                    "y": 10000  
                },  
                "direction": 3  
            },  
            "Barrier 2": {  
                "start": {  
                    "x": 16170,  
                    "y": 8000  
                },  
                "end": {  
                    "x": 16170,  
                    "y": 5600  
                },  
                "direction": 1  
            }  
        }  
    }  
}
```

barrier with AGILOX IO or time span

The response of AGILOX IO defines, whether the barrier should be set or not.
Barrier depending on input **1** of AGILOX IO **test** to be **1** or **true**

```
{  
    "barrier switch": {  
        "floor": 0,  
        "auto": false,  
        "barrier": {  
            "barrier 1": {  
                "start": {  
                    "x": 9000,  
                    "y": 6800  
                },  
                "end": {  
                    "x": 9000,  
                    "y": 9600  
                },  
                "direction": 5,  
                "radius": 400  
            },  
            "barrier 2": {  
                "start": {  
                    "x": 17000,  
                    "y": 6800  
                },  
                "end": {  
                    "x": 17000,  
                    "y": 9600  
                }  
            }  
        },  
        "agiloxio": {  
            "test": {  
                "input": {  
                    "1": 1  
                }  
            }  
        },  
        "operator": "or",  
        "time": {  
            "timespan": {  
                "0": {  
                    "start": "07:00",  
                    "end": "10:32"  
                },  
                "1": {  
                    "start": "13:00",  
                    "end": "22:00"  
                }  
            },  
            "dayofweek": {  
                "0": 1,  
                "1": 1,  
                "2": 0,  
                "3": 1,  
                "4": 1  
            }  
        }  
    }  
}
```

4.6.3 collective

```
{
  "coll1": [
    12345679,
    12345678
  ],
  "p3ufk": [
    12345678,
    10240990
  ],
  "sys3": [
    10240990
  ]
}
```

Attribute	Type	Required	Description
ID (here "coll1")	string	yes	Unique name for the collectives (free selectable).
Serial number	number	no	Serial number of the AGILOX vehicles, that are part of the collective. This can also be adjusted in system settings. 4.3.7.1 - System Settings

4.6.4 floor

The different floors of the area of movement can be defined in this collection.

```
[
  {
    "name": "EG",
    "order": 0
  },
  {
    "name": "OG",
    "order": 1
  },
  {
    "name": "KG",
    "order": -1
  }
]
```

Attribute	Type	Required	Description
name	string	yes	Name of the floor shown in hmi. Only short definitions are possible to be able to fit them in the symbol of the visualization:
order	number	yes	Order for the visualization of the floors in hmi: For the example above:

4.6.5 heartbeat

With this collection, a heartbeat can be defined. One, more or all AGILOX vehicle of the swarm can send a heartbeat to an AGILOX IO or a web service when they are online.

The heartbeat will be sent 4 times a second.

```
{
  "*": {
    "agiloxio": {
      "test": {
        "output": {
          "0": 1
        }
      }
    },
    "request": {
      "uri": "http://192.168.0.100/wms/hearbeat",
      "type": "webservice",
      "timeout": 2000,
      "json": true,
      "post": {
        "AGILOX Swarm": "online"
      }
    }
  }
}
```

Attribute	Type	Required	Description
*	string	yes	defines who should send a heartbeat: *: means all AGILOX vehicles of the swarm 12345678 : serial number on and AGILOX vehicles, which should send the heartbeat
agiloxio	object	no, if "request" is defined	The AGILOX IO object defines which output the heartbeat should be sent to. See 4.7.9 - AGILOX IO Request Definition
request	object	no, if "agiloxio" is defined	The request object defines the web service call. See 4.7.8 - Webservice Request

Example:

```
{
  "12345678": {
    "agiloxio": {
      "test2": {
        "output": {
          "0": 1
        }
      }
    }
  }
}
```

4.6.6 loadcarrier

The collection will change automatically if load carriers are changed in System Settings ([4.3.7.1 - System Settings - Load Carrier](#)) and the other way round.

Standard load carriers such as EPAL and EPAL6 as well as the AGILOX Box Carrier are not part of this collection.

```
{
  "lc1": {
    "name": "lc1",
    "width": "800",
    "length": "1300",
    "offset_feet": {
      "right": -120,
      "left": 50
    },
    "offset_entry": 2
  }
}
```

Attribute	Type	Required	Description
ID (here: "lc1")	string	yes	Unique ID for the load carrier. If a load carrier is defined in hmi's system settings, ID and "name" is identical.
name	string	yes	Name for the load carrier which is used in hmi.
width	string	yes	Width of the load carrier in mm. Can be defined in system settings as well.
length	string	yes	Length of the load carrier in mm. Can be defined in system settings as well.
offset_feet	object	no	For special load carriers that have feet that are not parallel (e.g. the right foot is 100mm further back than the left), the normal load carrier detection does not work, as it needs parallel feet. See Operation Manual 6.4.3 - Load Carrier Detection), the AGILOX vehicle would assume the load carrier to be positioned slanted and would not be able to pick it up correctly. For this case, "offset_feet" can be defined. The value can be positive or negative and can be defined for the right or the left foot "right" : the right foot has an offset to the left foot defined in mm (negative means further back) "left" : the left foot has an offset to the right foot defined in mm (negative means further back)
offset_entry	number	no	defines the offset in scan points while entering the load carrier, it is a numeric value and can be -5 to +5. It is recommended to set the value to 0 at first. If the AGILOX vehicle always touches the load carrier on the same side when entering, the value can be changed slightly. E.g. set it to 1 and see if it gets better. If it got worse change it to -1 and so on. Until the best value for offset_entry is found.



INFORMATION

Mixing up special load carries with offsets and normal load carries within one system is not possible. The AGILOX vehicle cannot detect which load carrier is on a station. If there are more special load carriers defined on one station, the AGILOX vehicle will always use the biggest offset defined for entering a load carrier.

4.6.6.1 Example

```
{  
    "epal3": {  
        "name": "EPAL_3",  
        "width": "800",  
        "length": "1300"  
    },  
    "special2": {  
        "name": "SPECIAL2",  
        "width": "1000",  
        "length": "1200",  
        "offset_feet": {  
            "right": -100  
        },  
        "offset_entry": -2  
    }  
}
```

4.6.7 mapping

For example, to assign a read barcode to a target, the collection /mapping can be used.

```
{  
    "target": {  
        "NTH123": "A1",  
        "SIY459": "\\"stationarea\\":["A"]}  
    }  
}
```

Attribute	Type	Required	Description
ID (here: "target")	string	yes	Name used in workflows. Defines which sub-area of the collection /mapping should be used. The raw key-value relationships are in there. See 4.7.7.3 - Targets from variables Multiple sub-areas are allowed.
key (here: "NTH123", "SIY459")	string	yes	This is the value which will be recognized in the bar code. The actual target (station, area) is assigned to that value.

4.6.8 supply_dispose

A supply/dispose order is generally speaking an automatically generated pickup-drop order. A load will be picked up from a station/area and dropped at another station/area.

```
[
  {
    "type": "supply",
    "auto": true,
    "priority": 0,
    "separate": true,
    "auto_condition": { <condition object> },
    "keep_siblings": true,
    "autoskip": false,
    "destination": { <target object> },
    "source": { <target object> },
    "surrogate": { <target object> },
    "workflow_start": {
      "id": 0,
      "variables": {
        "@VARIABLE1": "@SOURCE",
        "@VARIABLE2": "@DESTINATION"
      }
    },
    "barcode": { <barcode object> },
    "workflow": { <order object> }
  }
]
```

Attribute	Type	Required	Description
type	string	yes	supply: a station will be supplied if it is empty. dispose: a station will be emptied if it is occupied.
auto	bool	no	true means that the AGV supplies the station when it accidentally passes by the preposition within a meter without any workflow. Default: false
priority	number	no	If "auto":"true", a priority can be defined for the automatically generated order.
separate	bool	no	This can be used if a station area that is <u>separated</u> (and spread out all over the area of movement) is defined as source/target. If separate is set to true , each generated order has exactly one station as target/source instead of the whole station area. ATTENTION: This can cause problems if there are several vehicles combined with multiple-deep channels.
auto_condition	object	no	A condition can be defined, which needs to be met for the order to be generated automatically. Possible conditions see 4.7.5 - Workflow Condition Object
keep_siblings	bool	no	true means that sibling orders will not be canceled when the supply/dispose order is canceled (e.g. due to problems). Example Supply/Dispose for block storage: If the station in the very front is occupied, for each station of the channel an order will be created. If for one of them no target can be found and the order is canceled, all other orders will not be canceled if keep_siblings is set to true.

Attribute	Type	Required	Description
autoskip	bool	no	true: in the order created <i>autoskip</i> will be set to true for every action. Default: true (see 4.7.6 - Workflow Action Object). In case of an autoskip, the load will be carried back to the source, except a different station is defined in <i>surrogate</i> .
destination	object	yes	This target object defines the destination for the supply/dispose order. 4.7.7 - Workflow Target Object For supply orders this is the station that should be supplied.
source	object	yes	This target object defines the source for the supply/dispose order. 4.7.7 - Workflow Target Object For dispose orders this is the stations that should be emptied.
surrogate	object	no	If the load cannot be brought to the actual defined target or in case of an autoskip, an alternative target station can be defined here, for the load not to be brought back to the source.
workflow_start	object	no	An existing workflow with the corresponding ID can be started here. Variables can be used if they are defined in the workflow. @Variable1, in this case, is a variable of the existing workflow with id 0. @SOURCE and @DESTINATION are automatically filled with the target object defined above.
barcode	object	no	The destination can be determined by a bar code. See example below and 4.7.10 - Barcode Object
workflow	object	no	A complete workflow can be defined here. See example below and 4.7.3 - Workflow Order Object

4.6.8.1 Examples

Simple supply order

This example will supply station **A1** with load from stationarea **E**, in the defined time span.

```
[  
  {  
    "type": "supply",  
    "auto": true,  
    "auto_condition": {  
      "time": [  
        {  
          "timespan": [  
            {  
              "start": "08:00",  
              "end": "12:00"  
            },  
            {  
              "start": "12:30",  
              "end": "17:00"  
            }  
          ]  
        }  
      },  
      "keep_siblings": true,  
      "destination": {  
        "station": [  
          "A1"  
        ]  
      },  
      "source": {  
        "stationarea": [  
          "E"  
        ]  
      },  
      "surrogate": {  
        "stationarea": [  
          "X"  
        ]  
      }  
    }  
  }]
```

Simple dispose order

This example will dispose station **B5** to stationarea **L1C3**.

```
[  
  {  
    "type": "dispose",  
    "auto": false,  
    "source": {  
      "station": [  
        "B5"  
      ]  
    },  
    "destination": {  
      "stationarea": [  
        "L1C3"  
      ]  
    }  
  }]
```

Dispose order with existing workflow, source and target as variables

```
[  
  {  
    "type": "dispose",  
    "auto": false,  
    "source": {  
      "stationarea": [  
        "Fertigwaren"  
      ]  
    },  
    "destination": {  
      "stationarea": [  
        "Lager"  
      ]  
    },  
    "workflow_start": {  
      "id": 0,  
      "variables": {  
        "@VARIABLE1": "@SOURCE",  
        "@VARIABLE2": "@DESTINATION"  
      }  
    }  
  }  
]
```

Dispose order with destination from barcode

```
[  
  {  
    "type": "dispose",  
    "auto": false,  
    "source": {  
      "stationarea": [  
        "TESTLAGER"  
      ]  
    },  
    "destination": {  
      "stationarea": [  
        "@BARCODE"  
      ]  
    },  
    "barcode": {  
      "type": "code128",  
      "length": 2,  
      "surrogate": "B2",  
      "light": true,  
      "crop": {  
        "width": 100,  
        "height": 50,  
        "left": 0,  
        "top": 50  
      }  
    }  
  }  
]
```

Dispose order with complete workflow defined in collection

```
[
  {
    "type": "dispose",
    "auto": false,
    "source": {
      "stationarea": [
        "AREA-51"
      ]
    },
    "workflow": {
      "order": {
        "action": [
          {
            "action": "drive_over",
            "target": {
              "station": [
                "WAIT_AREA_51"
              ]
            },
            "action_skip": {
              "condition": {
                "distance": {
                  "station": "WAIT_AREA_51",
                  "radius": "<22000"
                }
              }
            }
          },
          {
            "action": "pickup",
            "target": "@SOURCE",
            "barcode": true,
            "autoskip": true
          },
          {
            "action": "drive_over",
            "target": {
              "station": [
                "WAIT_DROP"
              ]
            },
            "action_skip": {
              "occupied": false
            }
          },
          {
            "action": "drop",
            "target": {
              "stationarea": [
                "@BARCODE"
              ]
            },
            "barcode": {
              "type": "qr",
              "surrogate": "RESERVE",
              "regex_replace": {
                "pattern": "/(\d{3})(\D{1,3})(\d{2})(.*)/",
                "replacement": "${1}${3}"
              }
            },
            "autoskip": true
          },
          {
            "action": "drop",
            "target": {
              "stationarea": [
                "RESERVE"
              ]
            },
            "action_skip": {
              "occupied": false
            }
          }
        ]
      }
    }
]
```

```
    }  
}  
]
```

4.6.9 toggle

Toggles can be created in this collection to use them in conditions and workflow views.

```
{  
  "Toggle1": {  
    "value": false  
  },  
  "Toggle2": {  
    "value": true  
  },  
  "John": {  
    "name": "Johnny",  
    "value": false,  
    "image": "indiv/vis.svg"  
  }  
}
```

Attribute	Type	Required	Description
name	string	no	defines the name of the toggle shown in workflow views.
value	bool	no	shows the current value of the toggle
image	string	no	After consultation with AGILOX GesmbH a customized image for the toggle is possible.

4.7 Workflow Definition Language

Repeating tasks in an organization, can perfectly be depicted by workflows. A workflow is basically an order that may be enhanced by variables to make it more generic.

Workflows can be defined and adjusted in the Database Editor ([4.3.7.2 - System Database Editor](#)).

4.7.1 Workflow Variables

Variables can be used in workflows to make them more generic than a fully couched order.

Syntax

Variables must satisfy the following syntax:

```
@VAR_NAME;Var_Description;var_type[ ;default_value]
```

Note:

Var_Description and **var_type** are required to be able to start the workflow via HMI.
If a workflow is always started via API call by a host system, only @VAR_NAME is required.

@VAR_NAME

Starting with the @-sign, you may use any name for your variables (except those of *Special variables* below).

Var_Description

Description is free text, shown to the user on the HMI when choosing the value for this variable.

Var_Types

The variable type defines what stations or station areas should be shown to the user in the drop down list:

	Example	Description
station	@SRC;Pickup source;station	List of all stations
stationfiltered	@SRC;Pickup source;stationfiltered;/^Container/	List of stations, filtered by given regular expression (Stationname starts with "Container" in this case)
stationarea	@SRC;Select source area;stationarea	List of all stationareas
stationareafiltered	@SRC;Select source area;stationareafiltered;/^Water/	List of stationareas, filtered by given regular expression (Stationarea name starts with "Water" in this case)
stationareastation	@TARGET;Select target;stationareastation;Storage	List of all stations in the stationarea given. (All stations in the area "Storage" in this case)
list	@TARGET;Choose target;list;Store-1:str-1,Store-7:str-7,Backups	A list of targets to choose from separated by ", ". Note: ":" may be used to separate value from text being displayed (show "Store-7" on HMI, but use "strg-7" as variable value in workflow).

	Example	Description
input	@INFO;Add a note to this order;input;-none-	A text input box will be presented to the user on the HMI. Default value "-none-" is preset (or used if host system is starting this workflow omitting the variable)
serial number	@AGILOX;AGILOX;serialnumber	Serial number of an AGILOX vehicle
scanner	@DESTINATION;Destination;scanner;/emea\\KC{1}[0-9]{1}/	For hand scanners. The scanner input can be filtered by given regular expression. (in this case a text containing "emea" followed by "C" and one number. e.g. "emeaC1")

default_value

The default_value is used, if a host system omitted a variable.

Special variables

There are some special variables that can be used:

- **@SERIALNUMBER:**
will be replaced by the serial number of AGILOX that is executing the web service call.
Useful if a workflow is started from an AGILOX's HMI, as always that very AGILOX will do the job, even though the workflow syntax is generic.
- **@CURRENT_STATION:**
will be replaced by the current station the AGILOX vehicle is at.
- **@PREV_TARGET:**
will be replaced by the previous Target from action before.
- **@ORDERID:**
will be replaced by the current order id of the AGILOX vehicle.
- **@REQUEST:**
will be replaced by the response from a webservice request. See [4.7.8 - Webservice Request](#)
- **@BARCODE:**
represents the last determined barcode by the barcode processing object. See [4.7.10.2 - Barcode Processing Object](#)
- **userdata:**
a "non-variable" named "userdata" can be used to append its contents to the order object.
Be careful, case sensitive and without the preceding @
- **@MAPPING:**
is used for the collection "/mapping". see [4.8.9 - Mapping](#)

4.7.2 Workflow Object

```
{
  "17": {
    "name": "EXAMPLE",
    "status": "0",
    "maxinterval": "600",
    "canceltime": 3,
    "order": { <order object> }
  }
}
```

Attribute	Type	Required	Description
id ("17" in this case)	string	yes	unique identifier of a workflow, reserved keywords: • "purge" • "protectivestop" • "fire"
name	string	yes	friendly name for the workflow
status	number	yes	status: 0 = enabled, 99 = disabled and not shown in HMI
maxinterval	number	yes, if "auto": true	Time interval in seconds of starting a workflow automatically. Default: 0
canceltime	number	no	Activates a countdown with a cancel button on the workflow page if > 0. Handy to avoid accidentally starting a workflow. Default: 0
order	order object	yes	see 4.7.3 - Workflow Order Object

4.7.3 Workflow Order Object

```

"order": {
    "priority": 0,
    "repeat": 4,
    "workflow_max_pending": 5,
    "nonrelevant": true,
    "recipient": {
        "serialnumber": 12345678
    },
    "unique": true,
    "concurrent": {
        "class": "production",
        "count": 2
    },
    "interrupt": true,
    "concurrency": 100,
    "autoskip": true,
    "keep_siblings": true,
    "max_speed": 300,
    "deadline": 1543320000,
    "deadline_relative": 3600,
    "ignore_barrier": ["barrier1"],
    "active": { <agilox io object> / <webservice request> },
    "event": { <event object> },
    "start": { <condition object> },
    "resume": { <condition object> },
    "create": { <condition object> },
    "action": { <action object> }
}
    
```

Attribute	Type	Required	Description
priority	number	no	Defines the priority of an order. The numeric value is interpreted ascendingly - lower value, higher priority. The priority may be negative for higher priority. Default: 50
repeat	number	no	Defines the count of runs an order should be repeated. Default: 1
workflow_max_pending	number	no	Defines how many orders of this workflow can be pending at the same time. If the maximum number has been reached, the creation of another order with this workflow will be denied with an error message.
nonrelevant	bool	no	Nonrelevant orders are orders, which may be canceled while active. Cancellation is only possible, if not occupied and while driving from one station to another station. Default: false
recipient	object	no	Recipient can be a field of serial numbers (e.g. "serialnumber": [12345678, 87654321]) or collectivs ("collective": ["col1", "col2"]) or a mixture of both ("collective": ["col1"], "serialnumber": [12345678, 23456789]). If given, an order is addressed to the given group of AGILOX devices.

Attribute	Type	Required	Description
unique	bool / object	no	If set to true , this Workflow (with this ID) can only be done once at a time. Default: false Additionally, a class can be defined, which can be used in several workflows, to only perform one of these workflows at a time: "unique": { "class":"product1"} If an order with the same workflow ID or class is being created, while another workflow with this ID or class is still running, the system creates an error message and the order can not be created.
concurrent	object	no	A class can be defined here as well as the count of orders with this same class that may be startet at the same time.
interrupt	bool	no	In combination with repeat . If interrupt is set to true , after every run of the order, the order gets retained to possibly allow other orders with higher priority to be started in between. If no other order is intermediate, the interrupted order is started after a short time. Default: false
concurrency	number	no	If other AGILOX vehicles in the union are already processing orders in the same target area, e.g. the same channel of a block storage, as the first action of this order, this order will not be started, if there are still other orders to process. This avoids two AGILOX vehicles working in the same target area at the same time and potentially blocking each other. Value in %, default: 0.
autoskip	bool	no	If set to true AGILOX will skip pickup or drop actions, if it cannot fulfill them (unexpected occupied/empty). Default: false
keep_siblings	bool	no	If set to true, sibling orders will not be canceled when the order is canceled due to problems.
max_speed	number	no	Set maximum speed for this order. Value in mm/s. This option can only limit the maximum speed, but cannot make the AGILOX vehicle faster, if other (safety) reasons slow it down.
deadline	number	no	Defines a time (unix timestamp) when the order should be completed. Priority of the order is raised with the time deadline coming closer.
deadline_relative	number	no	
ignore_barrier	array of string	no	Defines barriers which will be ignored for this order.
active	agilox io object	no	An output of an AGILOX IO can be set or a webservice request can be started when the order is active.
event	event object	no	see 4.7.4 - Workflow Event Object

Attribute	Type	Required	Description
start	condition object	no	Consists of a condition, that needs to be met before the order can be started. See 4.7.5 - Workflow Condition Object
resume	condition object	no	Consists of a condition, that needs to be met for orders with the status "retained" to be continued. See 4.7.4.1 - Methods and 4.7.5 - Workflow Condition Object
create	condition object	no	Consists of a condition, that needs to be met, before the order can be created. See 4.7.5 - Workflow Condition Object
action	action object	yes	see 4.7.6 - Workflow Action Object

4.7.4 Workflow Event Object

Events can be defined within orders and actions. Events inside of actions are effecting only the specified actions. Events inside of orders are effecting all actions (be careful).

Events typically call methods, described below.

An event can be a simple defined action or up to a complex object structure.

```

"event": {
    "order_done": <method> / <object>,
    "order_started": <method> / <object>,
    "order_canceled": <method> / <object>,
    "action_done": <method> / <object>,
    "no_station_left": <method> / <object>,
    "carrier_empty": <method> / <object>,
    "max_retries": <method> / <object>,
    "condition_timeout": <method> / <object>,
    "obstruction_timeout": <method> / <object>,
    "order_created": <method> / <object>,
    "no_carrier": <method> / <object>,
    "no_target": <method> / <object>,
    "no_route": <method> / <object>,
    "no_response": <method> / <object>,
    "timeout": <method> / <object>,
    "target_reached": <method> / <object>,
    "loadcarrier_error": <method> / <object>,
    "floor_change": <method> / <object>,
    "no_barcode": <method> / <object>,
    "ambiguous_barcode": <method> / <object>,
    "overload": <method> / <object>,
    "loadcarrier_too_wide": <method> / <object>,
    "station_entered": <method> / <object>,
    "station_left": <method> / <object>,
    "conveyor_unready": <method> / <object>
}
    
```

Attribute	Type	Required	Description
order_done	string / object	no	Only in Orders, not in actions. Is triggered when the order is done.
order_started	string / object	no	Only in Orders, not in actions. Is triggered when the order is started.
order_canceled	string / object	no	Only in Orders, not in actions. Is triggered when the order is canceled.
action_done	string / object	no	Only in Actions. Is triggered when the action is done.
no_station_left	string / object	no	Is triggered, if no station is left in target list. See 4.7.7 - Workflow Target Object
carrier_empty	string / object	no	Is triggered, when carrier gets empty within a drop action.
max_retries	string / object	no	Is triggered, if max_retries of action is exceeded. The max_retries count has to be defined additionally within the action.
condition_timeout	string / object	no	Is triggered, if condition_station, condition_rotate or condition_lift in an action object is not met after the defined timeout. The timeout needs to be defined within the same action as well.
obstruction_timeout	string / object	no	Is triggered, in case of an obstruction within an action after the defined timeout has been reached. The time for obstruction_timeout has to be defined additionally within the action.

Attribute	Type	Required	Description
order_created	string / object	no	Is triggered, when the order is created.
no_carrier	string / object	no	For orders with Box Carrier: This is triggered when the action is not possible with the current configuration (e.g. box carrier empty at drop, box carrier full at pickup,...)
no_target	string / object	no	Is triggered, if there is no target determined for the current action
no_route	string / object	no	Is triggered, if no possible route to the target can be found.
no_response	string / object	no	This event is triggered while starting an action, if the action has a request defined in its root and the request failed. See 4.7.8 - Webservice Request
timeout	string / object	no	Is triggered, if a timeout is exceeded. The time for the timeout has to be defined additionally within the action.
target_reached	string / object	no	Is triggered, when drive-target is reached. Target can be a waypoint or a preposition of a regular station.
loadcarrier_error	string / object	no	Is triggered, if AGILOX has determined a station in a pickup action and cannot detect a valid load carrier during positioning, before entering.
floor_change	string / object	no	Is triggered, if the AGILOX vehicle uses a link station to change floors.
no_barcode	string / object	no	Is triggered, if there is no barcode information at the start of an action.
ambiguous_barcode	string / object	no	Is triggered, if more than one barcode left after filtering.
overload	string / object	no	Is triggered, if the weight of the load is too high.
loadcarrier_too_wide	string / object	no	Is triggered, when the AGILOX vehicle measures that the pallet during pick up and it is actually wider than the station would allow.
station_entered	string / object	no	Only in actions. Is triggered, when the AGILOX vehicle has entered the current station.
station_left	string / object	no	Only in actions. Is triggered, when the AGILOX vehicle has left the current station.
conveyor_unready	string / object	no	Only in actions. Is triggered, if box carrier target or source conveyor is not ready when approaching.

4.7.4.1 Methods

Example:

```
"event": {  
    "no_station_left": "action_skip"  
}
```

"action_skip"

Skip the current action.

"action_done"

Complete current action and continue with the following one.

"action_skip2"

Skip the current and following action.

"action_done2"

Complete current and following action and continue with the following one.

"order_cancel"

Cancel whole order.

"order_done"

Mark order as done.

"action_gotoXX"

Go to action with number XX.

"cancel_and_drop"

Only when the AGILOX vehicle is on a station, cancel current order and drop load here.

"order_retain"

The active order will be changed to the status "retained". The condition for the order to be resumed can be defined with the attribute "resume" in [Workflow Order Object](#).

"order_interrupt"

The active order will be interrupted and changed to status "inactive". The order is treated like any other inactive order. The "start" condition in [Workflow Order Object](#) is considered as usual.

4.7.4.2 Objects

Request

This is a full web service request definition. If called, the request is executed.

See [4.7.8 - Webservice Request](#)

Target

This is a full target definition overwriting the current target definition of an action.

See [4.7.7 - Workflow Target Object](#)

Agiloxio

This is a full agilox io request definition. If called, the agiloxio request is executed.

See [4.7.9 - AGILOXIO Request Definition](#)

Notification

```
"order_canceled": {  
    "notification": "Text"  
}
```

Defined Text is shown on Laser Scanner. AGILOX stops and lights the start button.

Occupation

```
"station_left": {  
    "occupation": {  
        "A1": false,  
        "A2": true  
    }  
}
```

Occupations of stations can be changed depending on events.

Workflow

Starts the workflow with the id defined.

```
"order_done": {  
    "workflow": {  
        "id": 0,  
        "variables": {  
            "@SOURCE": "@CURRENT_STATION",  
            "@DESTINATION": "B1"  
        }  
    }  
}
```

Attribute	Type	Required	Description
id	number	yes	ID of the Workflow, that should be started.
variables	object	depending on Workflow	Variables needed for the defined Workflow.

4.7.5 Workflow Condition Object

Conditions can be defined at different points in a workflows and the hmi.

In order to keep the workflow more generic it can be better to use the schedule function in [4.3.7.3 - Workflow Settings](#) as this makes it possible to define more rules for the same workflow.

```
{
    "operator": "and",
    "batterylevel": ">=50",
    "z": ">500",
    "nav_match": ">60",
    "secondary_match": ">30",
    "time": { <time> },
    "schedule": { <schedule> },
    "request": { <request> },
    "agiloxio": { <agiloxio request> },
    "distance": { <distance> },
    "idle": {
        "time": ">60"
    },
    "weight": ">100",
    "occupied": true,
    "stationdata": { <stationdata> },
    "target_occupied": { <target object> },
    "target_empty": { <target object> },
    "targetage": {
        "target": { <target object> },
        "maxage": 600
    },
    "toggle_on": ["Charge1", "Charge2", "operator": "or"],
    "toggle_off": "Charge3",
    "barrier_active": "Barrier1",
    "barrier_inactive": ["Barrier2", "Barrier3"],
    "serialnumber": [12345678, 87654321],
    "collective": "production",
    "action_agilox_count": {
        "actionid": "area23",
        "count": 0
    },
    "load_scan_count": ">10"
}
```

Attribute	Type	Required	Description
operator	string	no	" and " (all following conditions must be met) " or " (one of the following conditions must be met). Default: "and"
batterylevel	number with operator	no	checks current battery level in %
z	number with operator	no	checks current lift height in mm
nav_match	number with operator	no	checks the current navigation match in %
secondary_match	number with operator	no	checks the current secondary match in %
time	time object	no	see Time below
schedule	schedule object	no	see Schedule below
request	request object	no	see 4.7.8 - Webservice Request
agiloxio	agilox io object	no	see 4.7.9 - Agilox IO Request
distance	distance object	no	see Distance below
idle	object	no	true if AGILOX has been idle for defined time (in seconds)

Attribute	Type	Required	Description
weight	number with operator	no	checks weight, value in kilograms
occupied	bool	no	checks if AGILOX is occupied or not
stationdata	stationdata object	no	see Stationdata below
target_occupied	object	no	checks if the specified targets are occupied, true if at least one of the targets is valid and occupied.
target_empty	object	no	checks if the specified targets are empty, true if at least one of the targets is valid and empty.
targetage	object	no	checks when the occupation of the specified target was checked last. "maxage" means the occupation information must be older than the defined age in seconds for the condition to be true "minage" means the occupation information must not be older than the defined time in seconds for the condition to be true
toggle_on	string or array of strings	no	checks whether the specified toggle is currently on. A field of several toggles can also be specified. An optional "operator": "or" can be defined, which switches from the default "and" to "or" if multiple items are defined. See 4.6.9 - Collection toggle how to define toggles
toggle_off	string or array of strings	no	checks whether the specified toggle is currently off. A field of several toggles can also be specified. An optional "operator": "or" can be defined, which switches from the default "and" to "or" if multiple items are defined. See 4.6.9 - Collection toggle how to define toggles
barrier_active	string or array of strings	no	checks whether the specified barrier is currently active. A field of several barriers can also be specified. An optional "operator": "or" can be defined, which switches from the default "and" to "or" if multiple items are defined. See 4.6.2 - Collection barrier how to define barriers
barrier_inactive	string or array of strings	no	checks whether the defined barrier is currently inactive. A field of several barriers can also be specified. An optional "operator": "or" can be defined, which switches from the default "and" to "or" if multiple items are defined See 4.6.2 - Collection barrier how to define barriers
serialnumber	number or array of numbers	no	checks the serial number of the AGILOX vehicle
collective	string or array of strings	no	checks if the AGILOX vehicle is part of the given collective(s)

Attribute	Type	Required	Description
action_agilox-count	object	no	<p>checks how many AGILOX vehicles are on action areas with the defined id. In this example, the condition is fulfilled, if there is no AGILOX vehicle on an action area with the id "area23". "count" can be any number and can also have an operator (e.g. "count":"<4", "count":"!0", ...)</p>
load_scan_count	number with operator	no	checks how many scan rays of the navigation scanner are muted, because of the load protudes into the height of the nav scanner.

4.7.5.1 Negation

Each conditions can also be negated by adding a prefixed "!" :

Example:

```
"action_skip": {"!serialnumber": [12345678]}
```

The action will be skipped if serialnumber is not "12345678" .

4.7.5.2 Time

This is used to set a timespan and the days of the week in which the AGILOX should start the workflow.

```

"time": {
  "timespan": [
    {
      "start": "07:00",
      "end": "20:30"
    },
    {
      "start": "21:00",
      "end": "22:00"
    }
  ],
  "dayofweek": [
    1,
    2,
    3
  ]
}

```

Attribute	Type	Required	Description
timespan	array of time objects	no	Timespan consists of "start" and "end" time.
dayofweek	array	no	1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday, 7 = Sunday

4.7.5.3 Distance

```

"distance": {
  "radius": ">3000",
  "station": "Trash",
  "coordinate": {
    "x": 100000,
    "y": 100000
  }
}

```

Attribute	Type	Required	Description
radius	number with operator	yes	Radius in mm around the AGILOX
station	string	no, if coordinate is used	Defined Station
coordinate	object	no, if station is used	Defined Coordinates

4.7.5.4 Schedule

```
"schedule": {
  "1": [
    [
      "06:00",
      "12:00"
    ],
    [
      "13:00",
      "17:00"
    ]
  ],
  "2": [
    [
      "06:00",
      "12:00"
    ],
    [
      "13:00",
      "17:00"
    ]
  ],
  "3": [
    [
      "06:00",
      "12:00"
    ],
    [
      "13:00",
      "17:00"
    ]
  ],
  "4": [
    [
      "06:00",
      "12:00"
    ],
    [
      "13:00",
      "17:00"
    ]
  ],
  "5": [
    [
      "06:00",
      "12:00"
    ]
  ]
}
```

The first level "1" to "7" stands for the days of the week (Monday to Sunday). Every day has an array of timespans with start and ending time.

4.7.5.5 Stationdata

```
"stationdata": {
  "item": "charge",
  "value": true
}
```

Available stationdata items:

- **"charge"**
- "value": true or false

At this moment, "charge" is the only available stationdata.

4.7.6 Workflow Action Object

Every order consists of at least 1 action.

```
"action": {
  "0": {
    "tags": ["pickupC1", ...],
    "action": "pickup",
    "target": { <target structure> },
    "transfer_side": "left",
    "transfer": { <transfer object> },
    "ignore_barrier": ["barrier1"],
    "request": { <request> },
    "condition_station": { <condition object> },
    "condition_rotate": { <condition object> },
    "condition_lift": { <condition object> },
    "condition_timeout": 10,
    "action_skip": { <condition object> },
    "action_skip2": { <condition object> },
    "autoskip": true,
    "max_speed": 300,
    "acceleration": 100,
    "active": { <agilox io object> / <webservice request> },
    "retry": 180,
    "max_retries": 1,
    "obstruction_timeout": 30,
    "event": {
      "max_retries": <method/object>,
      "condition_timeout": <method/object>,
      "obstruction_timeout": <method/object>,
      <event object>
    },
    "action_remain": { <condition object> },
    "mintime": 60,
    "holdtime": 60,
    "barcode": true,
    "lock_group": true,
    "lock_immediate": true,
    "id": "007"
  },
  "1": {
    "action": "wait",
    "time": 60
    "signal": {
      "delay": 30
    }
  }
}
```

Attribute	Type	Required	Description
"0"			<p>action number: the actions are processed consecutively. It is also possible to leave gaps (e.g. 10, 20, 30, ...) for later changes of the workflow without having to renumber of all actions.</p>
tags	array of strings	no	The action can be tagged in preparation for a new feature in AGILOX Analytics, where you can search for such tags.
action	string	yes	<p>"pickup": Drive to a target and pickup load. Target is mandatory. The AGILOX vehicle recognizes whether a box carrier is needed for the target or not and automatically picks up or drops the box carrier if needed.</p> <p>"drop": Drive to a target and drop load. Target is mandatory. The AGILOX vehicle recognizes whether a box carrier is needed for the target or not and automatically picks up or drops the box carrier if needed.</p> <p>"drop_here": Drop load at current station.</p> <p>"drive": Drive to a target and stop there. Target is mandatory.</p> <p>"drive_over": The same as drive, but assume the action to be done 1000mm before reaching target. So a drive over near target without stopping is possible. Target is mandatory.</p> <p>"wait": Wait for defined time, trigger or for a condition (see 4.7.5 - Workflow Condition Object). Attribute "time", "trigger" or "condition" is needed, default is 60 seconds. "signal" can be defined to emit an audio signal once the time defined in "delay" has passed. The audio signal with ID 33 is used for this (ANNEX I - Audio Signals)</p> <p>"supply": Drive to a target, check if target can be supplied and automatically create pickup and drop-action within the order. A database-collection 4.6.8 - Collection supply_dispose is necessary for the assignment. Target is mandatory.</p> <p>"dispose": Drive to a target, check if target can be emptied and automatically create pickup and drop-action within the order. A database-collection 4.6.8 - Collection supply_dispose is necessary for the assignment. Target is mandatory.</p> <p>"charge": Drive to the nearest charging station.</p> <p>"park": Drive to the nearest suitable parking station.</p> <p>"fire": Internal action only. Do not use!</p>
time	number	no	Time in Seconds (e.g. for action "wait"). Default: 60

Attribute	Type	Required	Description
trigger	string	no	For action " wait ", values are: • "start" to light the start button and wait for it to be pushed • "v" to wait for a V sign. At least two fingers in plausible distance and plausible thickness need to be detected by the navigation laser scanner in the back of the vehicle. The detection is confirmed by an acoustic signal.
target	target object	yes, if the action needs it	Defined Target for the action above, see 4.7.7 - Workflow Target Object
transfer_side	string	no	For actions which require a box carrier, the side used to transfer boxes can be defined.
transfer	object	no	see 4.7.6.1 - Transfer Object
ignore_barrier	array of string	no	Define barriers which are ignored for this action. Multiple barriers can be given.
request	request object	no	see 4.7.8 - Webservice Request
event	event object	no	see 4.7.4 - Event Objects
condition_station (former: station_enter)	condition object	no	Checks a condition before entering the station. If condition is true → continue, if false → wait until condition becomes true or implemented timeout. See 4.7.5 - Condition Objects
condition_rotate (former: rotate_enter)	condition object	no	Checks a condition before rotating to enter a station (on Preposition). If true → continue, if false → wait until condition true or implemented timeout. See 4.7.5 - Condition Objects
condition_lift	condition object	no	Checks a condition before lifting during pickup action. If true → continue, if false → wait until condition true or implemented timeout. See 4.7.5 - Condition Objects
condition_timeout	number	no	Define timeout for conditions above. If condition is not met within the defined time in seconds, the event "condition_timeout" is triggered.
action_skip	condition object	no	Skips the action, if condition is true. See 4.7.5 - Condition Objects
action_skip2	condition object	no	Skips current and following action, if condition is true. See 4.7.5 - Condition Objects
action_remain	condition object	no	Checks a condition before going to the next action. If true → remain, if false → go to next action. See 4.7.5 - Condition Objects
max_speed	number	no	Set maximum speed for this action. Value in mm/s. This option can only limit the maximum speed, but cannot make the AGILOX vehicle faster, if other (safety) reasons slow it down.

Attribute	Type	Required	Description
acceleration	number	no	Set maximum acceleration for this action. Value in mm/s ² . This option applies only for driving and has no influence to stationhandling or positioning before entering a station.
active	agilox io object	no	An output of an AGILOX IO can be set or a webservice request can be started when the action is active.
autoskip	bool	no	If set to true AGILOX will skip that pickup or drop action, if it cannot fulfill it (unexpected occupied/empty). Default: false If autoskip is false and the first action of the workflow does not match the occupation status of the AGILOX vehicle, the order will not be started.
retry	number	no	Time in Seconds until retrying, if AGILOX cannot fulfill the action. Default: 10s
max_retries	number	no	Maximum number of retries. If exceeded, event "max_retries" is triggered.
obstruction_timeout	number	no	If an obstruction happens within the action, the event "obstruction_timeout" is triggered after the defined time in seconds.
mintime	number	no	Minimum time to stay in the action.
holdtime	number	no	Time to stand still and wait after an action is completed.
barcode	bool / barcode object	no	If set to true on a pickup action, AGILOX is trying to read a barcode and storing it for further use. Default: false For processing barcodes in a drop action, see 4.7.10 - Barcode Objects
lock_group	bool	no	"lock_group" locks the first channel of the action and also all channels, that have their prepositions in a radius of 3m around the first channel of the action. See 6.4.8 - Station/Channel Locking for details.
lock_immediate	bool	no	"lock_immediate" locks the first channel of the action immediately and the AGILOX vehicle does not start the action, if the lock is not granted. See 6.4.8 - Station/Channel Locking for details.
id	string	no	ID for the picked up or dropped item. Useful for AGILOX Box Carrier. The picked up item can be given an id, which can be used again to drop the correct item later. If no id was given, the foremost item is always dropped first.

4.7.6.1 Transfer Object

In the transfer object either a workflow is defined or a workflow with a workflow ID is called.

With this workflow, a pallet that is blocking the way to the actual destination can be moved out of the way. The current order is retained and a new transfer order is generated according to the transfer definition.

```
"transfer": {  
    "workflow": {  
        "order": {  
            "action": {  
                "0": {  
                    "action": "pickup",  
                    "target": "@SOURCE"  
                },  
                "1": {  
                    "action": "drop",  
                    "target": "@DESTINATION"  
                }  
            }  
        }  
    }  
}
```

"@SOURCE" is required and is filled automatically. "@DESTINATION" is optional and is filled with all stations of the current stationarea to move the pallet within the same stationarea. All stations of the source channel are excluded to avoid blocking the path again. Any destination can be defined, but make sure that stations of the source channel are excluded. This can be done with "@DESTINATION_NOT", which is automatically filled with all stations of the source channel.

If you do not want to define the entire workflow in the transfer object, you can also define the workflow separately and only call the workflow ID here. The variables @SOURCE, @DESTINATION and @DESTINATION_NOT are handed over automatically.

```
"transfer": {  
    "workflowid": 23  
}
```

4.7.7 Workflow Target Object

The defined target(s) are resolved to a list of stations in the first step of an action. There is no difference, if single stations, multiple station up to multiple stationareas are given, AGILOX always resolves to a list of stations.

Additionally a special target **coordinate** is possible, see examples.

```

"target": {
    "station": [ "C1" ],
    "stationarea": [ "A", "B" ],
    "!station": [ "A1", "B4" ],
    "not": {
        "station": [ "A4" ],
    },
    "stationorarea": [ "C1", "B", "D2" ],
    "sort": [ "sequential", "occupied" ],
    "coordinate": [
        {
            "x": 10000,
            "y": 10000,
        }
    ]
}
    
```

Attribute	Type	Required	Description
station	array of stations	no	Target Station(s)
stationarea	array of stationareas	no	Target Station Area(s)
!station (or !stationarea)	array of stations / stationareas	no	Individual stations/stationareas can be excluded from the target list. e.g. the whole stationarea A and B, except station A1 and B4
not	object	no	Individual stations or stationareas can be excluded from the target list. e.g. the whole stationarea A, except station A4 "not" itself is defined like a target structure and can interpret the same attributes.
stationorarea	array of stations or stationareas	no	The target can be a mixture of station(s) and area(s). In combination with "sort":"sequential" this can be used to get a mixed sort order with stations and areas.
coordinate	object	no	see coordinate target below
sort	string / array of strings	no	Can be one or more from the attributes below. If more than one, the sort order is in order of appearance in the array. Distance is always the last criteria for all target sorting methods. "sequential" : No sorting is done. The order given in stations and stationareas above is used. "station" : The sort order is defined in the station dialogue in the hmi. See 4.3.2 - Stations "occupied" : occupied stations first "empty" : empty stations first "concurrency" : targets that are already addressed by other AGILOX vehicles are sorted last "touched" : targets that have been touched (pickup, drop) by an AGILOX vehicle come first

4.7.7.1 Coordinate

A special coordinate-target is possible. No station is defined, instead the specification of a station.

```
"target": {  
    "coordinate": [  
        {  
            "x": 10000,  
            "y": 10000,  
            "angle": 9000  
        }  
    ]  
}
```

Angle value in 1/100 °.
x and y are mandatory.

4.7.7.2 Filter Stations/Stationareas

Target lists can also be filtered, using regex. See example below.

```
"target": {  
    "station":{  
        "0": "@STAT;Station;stationfiltered;/NTH/"  
    }  
}
```

This also works with "stationareafiltered" to filter station areas.

4.7.7.3 Targets from variables

Targes can also be defined via a webservice call or a barcode.

Webservice Call

```
"target": "@REQUEST"
```

The target is the response of the webservice call, that is defined within the same action. The response needs to have a valid target structure.

Barcode

```
"target": {  
    "station": "@BARCODE"  
}
```

The target is the result of a barcode processing object, defined within the same action. The Barcode needs to be processed in a way that it has a valid target structure ([4.7.10.2 - Barcode Processing Object](#)).

Mapping

If the barcode does not contain a valid target structure and no webservice is to be queried, a target structure can be assigned to the barcode via the mapping collection.

```
"target": "@MAPPING;target;@BARCODE"
```

@BARCODE is replaced by the result of a barcode processing object, defined within the same action. A valid target structure is assigned to the Barcode via the collection /mapping. See [4.8.9 - Mapping](#)).

4.7.8 Webservice Request

All webservice requests are entered in a list (queue) and the system tries to send them again and again in the background. Identical requests will not be processed more than once.

- Requests triggered from an event, are used to give information to a web service (e.g. order finished.). The response of this request is being ignored.
- Requests triggered by a variable (@REQUEST) in an action, are replacing the variable with the response from the request. If there is still no answer after 5 seconds, the request is being cancelled.

</WRAP>

Example:

```
"request": {  
    "uri": "http://192.168.0.100/wms/get_target",  
    "headers": {  
        "token": "Bearer 0123456789ABCDEF"  
    },  
    "post": {  
        "Material": "Aluminium"  
    },  
    "surrogate": {  
        "station": "A1"  
    },  
    "json": true  
}
```

Attribute	Type	Required	Description
uri	string	yes	uri of requested web service
headers	object	no	HTTP header field. Any headers and values can be defined. E.g. Login/Authentication of API interface can be done with this.
post	string	no	A text or a json command can be posted.
surrogate	object	no	Only for @REQUEST. If there is no response after 5 seconds, the variable is set to this value.
json	bool	no	Defines whether the response of the web service is formatted in json or not. Default: false

4.7.9 AGILOXIO Request Definition

```

"agiloxio": {
  "conveyor1": {
    "output": {
      "0": 1,
      "1": 1
    },
    "command": {
      "pickup_start": true,
      "warning_light": true
    },
    "input": {
      "operator": "or",
      "0": 1,
      "1": 1
    },
    "state": {
      "readyfordrop": true,
      "occupied": false
    }
  }
}
    
```

Attribute	Type	Required	Description
conveyor1			friendly name of the agiloxio as defined in database. See 4.3.7.2 - System Database Editor
output	object	one of these structures	set output according to list below or according to the numbers in 4.3.7.5 - AGILOX IO
command	object		set outputs according to command defined in database. See 4.3.7.2 - System Database Editor
input	object		check input signal according to number on agilox io.
state	object		check inputs according to state defined in database. See 4.3.7.2 - System Database Editor
operator	string	no	operator for input check can be "and" or "or". Default: "and"

output

- set or reset agiloxio outputs according to the table below.

SET OUTPUT	ACTION	REAL OUTPUT
0	set output with timeout	0
1	set output with timeout	1
2	set output with timeout	2
3	set output with timeout	3
4	set output with timeout	4
5	set output with timeout	5
6	set output with timeout	6
7	set output with timeout	7
8	set output with no timeout	0
9	set output with no timeout	1
10	set output with no timeout	2
11	set output with no timeout	3
12	set output with no timeout	4
13	set output with no timeout	5
14	set output with no timeout	6
15	set output with no timeout	7
16	reset output	0
17	reset output	1
18	reset output	2
19	reset output	3
20	reset output	4
21	reset output	5
22	reset output	6
23	reset output	7

4.7.10 Barcode Object

The barcode can be used with the variable @BARCODE. It can be used as target or can be sent as an information to a webservice.

See also [Commissioning Manual 5.10 - Workflow with Barcode](#).

4.7.10.1 Barcode Reading Object

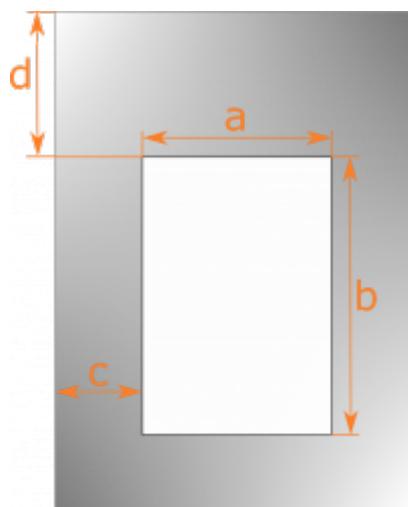
Read Barcode in pickup action, using default settings:

```
"barcode": true,
```

Or set options for barcode reading:

```
"barcode": {
    "light": false,
    "crop": {
        "width": 100,
        "height": 50,
        "left": 0,
        "top": 50
    },
    "datamatrix": false
}
```

Attribute	Type	Required	Description
light	bool	no	Turns camera light on or off. Default: false
crop	object	no	Crops barcode image.
width	number	no	crop width in %, see a in picture below
height	number	no	crop height in %, see b in picture below
left	number	no	crop position left in %, see c in picture below
top	number	no	crop position top in %, see d in picture below
datamatrix	bool	no	optional to disable datamatrix barcode detection (takes a lot of time)



4.7.10.2 Barcode Processing Object

Fill the variable @BARCODE with the result of the barcode processing object. The variable can be used in all following actions.

```
"barcode": {
    "type": "code128",
    "length": 2,
    "reverse": true,
    "regex": "/.*99.*/",
    "regex_replace": {
        "pattern": "/(\d{3})(\D{1,3})(\d{4})(.*)/",
        "replacement": "${1}${3}"
    },
    "surrogate": "NOREAD"
}
```

Attribute	Type	Required	Description
type	string	no	Filter for barcode type. Valid types are: <ul style="list-style-type: none"> • "qr" • "code39" • "code128" • "gs1-128" • "itf" • "datamatrix"
length	number	no	Filter for barcode length: Number of digits.
reverse	bool	no	If true, the last barcode of remaining barcodes is used.
regex	string	no	Filter to barcode content, see example below.
regex_replace	object	no	see example below
surrogate	string	no	If no barcode can be found, the response of the barcode request is set to this value.

Filter regex

```
"regex": "/.*99.*/"
```

Filter to barcodes containing 99.

Replace regex

```
"regex_replace": {
    "pattern": "/(\d{3})(\D{1,3})(\d{4})(.*)/",
    "replacement": "${1}${3}"
}
```

Processes barcode, example: **010MTB123456789** to **0101234**.

For further information look at [PHP function manual](#)

4.7.11 Special Workflows

There are some predefined workflow *IDs* for special tasks.
Those can be enabled or disabled on the HMI in workflow-settings.

4.7.11.1 PURGE

The purge workflow will be triggered if the vehicle is not processing an order but is occupied.
This happens e.g. if an order is canceled or action_skip events let the order finish without AGILOX dropping its load.

```
"purge": {  
  "name": "PURGE",  
  "status": 0,  
  "order": {  
    "action": {  
      "0": {  
        "action": "drop",  
        "target": {  
          "stationarea": "A"  
        }  
      }  
    }  
  }  
}
```

4.7.11.2 PROTECTIVE STOP

The protective stop workflow is triggered if someone steps into a station whilst AGILOX wants to drop load.
If no workflow is defined, AGILOX will stop and light the start button instead.

```
"protectivestop": {  
  "name": "PROTECTIVE STOP",  
  "status": 0,  
  "order": {  
    "action": {  
      "0": {  
        "action": "drive",  
        "target": {  
          "station": "standby"  
        },  
        "event": {  
          "action_done": {  
            "notification": "Press start button to continue"  
          }  
        }  
      }  
    }  
  }  
}
```

4.7.11.3 FIRE

If a fire alarm system is implemented, the AGILOX vehicle behaves as follows in case of fire:
The AGILOX vehicle stops immediately and is looking for the closest point on the map in the orange area of the route map, that is not on a clearway area. This point is the vehicle's target. Avoidance areas are ignored in this case. When the target is reached, the start button is lit and the AGILOX vehicle is waiting for acknowledgment.

Alternatively, a workflow with the ID "fire" can be defined, which will be startet in case of fire.

If the AGILOX vehicle is within a station when the fire alarm appears, it stops immediately, lights the start button and is waiting for acknowledgment.

If the AGILOX vehicle is on a charging station when the fire alarm appears, the start button also lights up and the vehicle is waiting for acknowledgment as well.

4.8 JSON API

Each AGILOX device (vehicle or gateway) provides a web service for communication with foreign systems. The AGILOX JSON API is used by partner systems in the company to access information from the AGILOX system or the creation of orders.

AGILOX can perform many complex actions, this document describes the most common applications. If you need further information, please feel free to contact AGILOX North America, Inc..

4.8.1 Basic usage

Every JSON-call expects and returns one multidimensional object. If only a subset of an object is wanted, a URL-subfolder can be used for selecting.

4.8.1.1 URL

The URL for POST and GET commands consists of hostname / IP address and port. The default port is 8100.

Example of a Valid API request using cURL:

```
curl http://ip-adress:8100/ci
```

4.8.1.2 Basic definitions

- timestamps are stored and used in unix time (seconds since 1970, see https://en.wikipedia.org/wiki/Unix_time).
- distances are defined in mm
- angles are defined in °/100 (whereas ° in HMI)

4.8.1.3 Flags

Flags can be added before API commands to change the response.

- **/gz/...:**
The response is in a gzip format. (<https://en.wikipedia.org/wiki/Gzip>)
- **/noversion/...:**
The response is without a **version**. Useful for database endpoints, that would always send a version in the end.
- **/compat/order/...:**
For large order objects, that are none JSON-conform listed. The orders are then separated by their order ID instead of "newline" separated.

4.8.1.4 Security

SSL

Inter-AGILOX communication uses SSL. For host or client communication SSL is also available and recommended. Therefore a SSL certificate and key file has to be uploaded in the AGILOX system settings HMI. Additionally, the DNS-infrastructure has to resolve the certificate to the AGILOX vehicle's wifi ip. This can only be done by the customer's IT department. The SSL-port for the AGILOX HMI is 443, the SSL web service port is 8443.

API-Key

An API-Key may be generated or defined in the system settings HMI. If a key is defined, all web service POSTs require this key additionally in their header field. GETs, however, are still possible without an API key.

For example:

```
curl\  
-d '{"@SOURCE":"A1", "@DESTINATION":"B1"}' \  
-H "apikey: SDFLIEJLE...DLKFJSLKDJF" \  
-H "Content-Type: application/json" \  
-X POST https://12345678.agilox.net:8443/workflow/0
```

If the webservice application does not support HTTP headers, the "apikey" can also be sent in the post body.

```
{"@SOURCE": "C1", "apikey": "jdfsj...fk1"}
```

4.8.2 Most Common Application

Basic Workflows are defined during commissioning by AGILOX North America, Inc..

Repeating tasks in an organization can perfectly be depicted by workflows. A workflow is basically an order that may be enhanced by variables to make it more generic.

Workflows can either be started from each vehicle's HMI, from a special workflow view or via a web service call.

The most common application for JSON API is to start such workflows via a web service call.

4.8.2.1 GET Workflows

```
curl http://ip-adress:8100/workflow/0
```

The returned object lists all available workflows. The first level of indexes are workflow-ids.

Note: calling the web service without "0" will also work, but as only one workflow object (containing all workflows) is existing, it's OK to directly access object "0" and thus avoid getting a useless layer in the response.

4.8.2.2 GET Single Workflow

```
curl http://ip-adress:8100/workflow/0/2
```

The returned object shows the workflow with id 2.

Note: You need to have the "0" in your call, as workflow number 2 is a subset of the [only] workflow object. See "Workflows" above.

Example

```
{
  "id": 2,
  "name": "AREA STATION",
  "status": 0,
  "order": [
    {
      "action": [
        {
          "action": "pickup",
          "target": {
            "stationarea": "@SOURCE;Source;stationarea"
          }
        },
        {
          "action": "drop",
          "target": {
            "station": "@DESTINATION;Destination;station"
          }
        }
      ],
      "recipient": [
        {
          "serialnumber": "@SERIALNUMBER"
        }
      ]
    }
  }
}
```

Attribute	Type	Description
id	number	The id has to be system-unique and numeric. It's being used to start the workflow.
name	string	Name is the display name used in HMI and workflow view.
status	number	Any value greater or equal 90 will disable the workflow. It won't be shown in HMI and cannot be called by web service.
order	object	See description of order below for details.

4.8.2.3 POST Workflow

To call a workflow from a host system, a JSON object containing all variables has to be created. The object can be posted to the web service in the post-body.

The web service recognizes the presence of post-data and interprets the incoming data.

The post body for the above sample could look like this:

```
curl \
-d '{"@SOURCE":"Store A", "@DESTINATION": "Station 3", "@SERIALNUMBER": "10000001"}' \
-H "Content-Type: application/json" \
-X POST http://ip-adress:8100/workflow/2
```

After posting this to the above URL, AGILOX 10000001 will pick up one EPAL from "Store A" block storage and drop it at "Station 3".

Workflow Variables (@VAR_NAME)

Variables are used in workflows, to make them more generic than a fully couched order. The web service call fills the variables with the required values.

See Commissioning Manual [4.7.1 - Workflow Variables](#) for more details.

POST to change order created by workflow

Orders created by workflows can be updated or changed by posting the workflow again with the given "orderid" in post body.

```
curl \
-d '{"@SOURCE":"Store A", "@DESTINATION": "Station 5", "@SERIALNUMBER": "10000001", "orderid": 844005161879770013}' \
-H "Content-Type: application/json" \
-X POST http://ip-adress:8100/workflow/2
```

4.8.3 General GET endpoints

API Name	API command	Description
System Overview	/ci	Get status object of all known AGILOX vehicles of the current union.
Order Pool	/order	The returned object shows the last 1.000 order-objects.
Swarm	/swarm	Shows all AGILOX systems of the swarm with serial numbers, name, status and IP address.
Database Endpoints		
Action	/action	The returned object shows all existing action areas of the map including their action area definition. See Commissioning Manual 4.5 - Map for detailed information.
AGILOX IO	/agiloxio_data	The returned objects shows all existing AGILOX IOs, their input states, states of the defined "states" and potential errors
Features	/feature	Shows the status of the modem (enabled/disabled) according to hmi system settings.
AGILOX Analytics	/fleetmanagement	Timestamps of successful communications with AGILOX Analytics.
Floor	/floor	Shows all available floors and their definition. See Commissioning Manual 4.6.4 - Collection floor
Translation	/localization	Shows every text of the hmi and corresponding translations. /0/1 for German and English /0/2 for Italian and English
Failures and Problems	/message	Decoding data for ci-failures and ci-problem. Failures are int.bit structured, problem is a single integer referring to the message/problem-list.
AGILOX Name	/name	Name of AGILOX vehicle.
Notification	/notification	Notification settings and E-Mail addresses according to hmi notification (Commissioning Manual 4.3.6.3 - Notification)
Occupation	/occupation	Occupation of all stations and load carrier data (if load carrier was dropped there by an AGILOX vehicle)
Order id	/orderid	ID of a current Order of the AGILOX connected to. Empty if the current AGILOX vehicle has no current order.
Station	/station	List of stations including their configurations.
Stationarea	/stationarea	List of stationareas.
Supply / Dispose	/supply_dispose	Collection of stations, which should be automatically supplied or disposed.
Telemetry	/telemetry	Collected cumulated drive-data of the current vehicle.
Union Name	/union	Union name, the current AGILOX vehicle is a member of.
User	/user	All available users for hmi login
Workflow	/workflow	The returned object lists all available workflows. See GET Workflows above.

Special endpoints		
API Collection	/collections	Lists all available database endpoints and their version count. See table above.
IP Address	/ip	Get IP address of current AGILOX vehicle.
Time	/time	Get current time of AGILOX vehicle as unix time.
Backup	/backup	Get encrypted backup of AGILOX union. This file contains all data for successful recovery in case of fundamental data loss.
Software Backup	/backup_software	Get encrypted software backup.
Map	/map.svg	Get the current visual map.
Failure	/failure	Get the current failure array, same as failure in ci.
Wi-Fi	/wifi	List of wifi networks in range of vehicle.
Serial Number	/snr	Get serial number of AGILOX vehicle currently connected to.
Cleanup	/cleanup	Trigger cleanup of unused stations, stationareas or orders. /cleanup/station /cleanup/stationarea /cleanup/order

4.8.4 System overview

Get status object of all known AGILOX vehicles of the current union.

```
curl http://ip-adress:8100/ci
```

The returned object encloses all known AGILOX vehicles. The first level of indexes are device-serial numbers. If only a subset of an object is wanted, a URL-subfolder can be used for selecting.

4.8.4.1 Device overview

Get status object of a specific AGILOX by using its serial number.

```
curl http://ip-adress:8100/ci/13096698
```

The returned object is an overview for the AGILOX "RALPH" with the serial number 13096698. Example:

```
{
  "filechange": 1538119132.9143,
  "serialnumber": 13096698,
  "status": 0,
  "release": 1,
  "union": "agilox",
  "name": "RALPH",
  "collective": false,
  "timestamp": 1538119150.3106,
  "position": [
    18640,
    26028,
    9176,
    -8609,
    46,
    48,
    1,
    0,
    496
  ],
  "feedback": [
    <feedback structure>
  ],
  "telemetry": {
    "distance": 100090296,
    "angle": 172199020,
    "busy": 300483.6161182,
    "idle": 1080393.539237,
    "operatingtime": 1380877.155355,
    "power": -52582.511682749
  },
  "route": [
    {
      "x": 26745,
      "y": 8860,
      "action": false,
      "area": 0
    },
    {
      "x": 26750,
      "y": 9150
    }
  ],
  "time_stopped": 0,
  "time_idle": 0,
  "wifilevel": 0,
  "batterylevel": 98,
  "weak_connection": 0,
  "occupied": false,
  "type": 1,
  "sw_version": 20180928,
  ...
  ...
  "obstruction": false,
  ...
  "failure": [
    0,
  ]
}
```

```
...
],
"ip": "10.10.55.91",
"ip_modem": "10.99.94.111",
"problem": false,
"hint": false,
"loadcarrier_data": [],
"battery_remain": 23982,
...
"bad": false,
...
"uptime": 4659,
"fork_overlength": 0,
...
"ssl": {
    "issuer": false,
    "valid": 1853058754,
    "cn": "13096698.agilox.net",
    "serialnumber": "85AC1953AD51D484"
},
"autoworkflow": {
    ...
},
"revision": {
    "action": 1822,
    ...
    "workflow_view": 93
},
"led": {
    "left": {
        "mode": 1,
        "red": 0,
        "green": 0,
        "blue": 0,
        "white": 128
    },
    "right": {
        "mode": 1,
        "red": 0,
        "green": 0,
        "blue": 0,
        "white": 128
    }
},
"station": false,
"station_approx": false,
"channelid": false,
"channel_group": 0,
"station_park": false,
"station_charge": false,
...
"ssdstatus": {
    "capacity": 68719476736,
    "written": 294440140800,
    "worn": 0.4,
    "used": {
        "\\"": 56559013888
    }
},
"ipc_image_version": 20180809,
"agilox_last": {
    "13096698": 1538119150.0104,
    "12345678": 1538119150.1068,
    "10240990": 1538119150.0857
},
"orderid": 130966981865130002,
"workflow": "11"
}
```

All relevant attributes are described in the table below, other attributes are for internal use only.

Attribute	Type	Description
filechange	number	Internal timestamp
serialnumber	number	Serial number of vehicle
status	number	Operation status of the vehicle: • 0 = "normal" • 90 = "shutting down"
release	number	Major software version. Default: 1
union	string	Union of vehicles it is dedicated to
name	string	Friendly name of the vehicle
collective	string / false	List of collectives the device is assigned to. Default: false
timestamp	number	Last refresh of data
position	array of numbers	Current position of device. Values are: • position sequence number • x [mm] • y [mm] • angle [°/100] • position match [%] • asd match [%] • asd run • actual floor • position speed
feedback	array of numbers	Data of plc communication for internal usage. See description below.
telemetry	object	Telemetry-data of device consisting of: • distance: overall driven distance of lifetime in mm • angle: overall rotation in °/100 • busy: busy time in seconds • idle: idle time in seconds • operatingtime: overall uptime in seconds • power: power consumption in Wh
route	array of objects	Current route to target with all calculated route points. False if no current route.
time_stopped	number	how long the vehicle has stopped (unexpectedly) [sec.]
time_idle	number	how long the vehicle has been idle [sec.]
wifilevel	number	Wifi signal level [%]
batterylevel	number	Battery level [%]
weak_connection	number	time the wifi connection has been bad for [sec.]
occupied	bool	current occupation of AGILOX vehicle (true / false)
type	number	1 for AGILOX vehicle 0 for AGILOX Host Gateway
sw_version	number	current software version (ipc)
obstruction	bool	current obstructions
failure	array of numbers	Ongoing failures. None if all are 0. Check http://host:8100/message for corresponding failure text

Attribute	Type	Description
ip	string	IP address of device
ip_modem	string	IP address of LTE modem
problem	number / false	Ongoing problem, if blocking problem exists. Check http://host:8100/message for corresponding problem text
hint	string / false	Human readable hint of vehicle's state
loadcarrier_data	array of objects	current load carrier data: <ul style="list-style-type: none"> • length (from source station definition) • width (measured 78mm from the ground) • offset • weight (roughly estimated)
battery_remain	number	battery remain time [sec.]
bad	bool	is "true" if <ul style="list-style-type: none"> • vehicle has failures • one of the push buttons (start or acknowledgement) is on • vehicle is not in automatic mode • vehicle is paused
uptime	number	uptime [sec]
fork_overlength	number	fork overlength [mm]
ssl	object	SSL information
issuer	string / false	name of issuer, false if issued by AGILOX vehicle
valid	number	valid until [unix time]
cn	string	common name
serialnumber	string	serial number of certificate
led	object	Information about left/right LED color and operation mode.
left	object	Information about left LED
mode	number	<ul style="list-style-type: none"> • 0 = off, • 1 = static color, • 2 = blinking, • 3 = blinking and fading, • 4 = fast blinking, • 5 = fast blinking and fading
red	number	red color component
green	number	green color component
blue	number	blue color component
white	number	white color component
right	object	Information about right LED
station	false / string	Station the AGILOX vehicle is currently standing in Default: false
station_approx	false / string	Approximate station the AGILOX vehicle is currently at. In contrast to "station", "station_approx" will also be filled with the station during entering or leaving a station and can be useful for foreign system, to check if the AGILOX vehicle is still busy with that station. Default: false

Attribute	Type	Description
channelid	false / number	Current channel id Default: false
channel_group	false / number	Current channel group Default: false
station_park	bool	Is the current station a parking station?
station_charge	bool	Is the current station a charging station?
ssdstatus	object	Health status of the internal SSD
capacity	number	size [byte]
written	number	how many bytes have been written
worn	number	wear level
used	number	memory in use [byte]
agilox_last	object	When have the other AGILOX vehicles of the union been seen last. [unix time]
orderid	number / false	Current order ID the device is processing
workflow	string / false	Current workflow ID the device is processing

feedback

```
{
    "feedback": [
        484,          // 0
        -2304,         // 1
        153,           // 2
        0,             // 3
        1,             // 4
        0,             // 5
        3,             // 6
        0,             // 7
        64,            // 8
        4406,          // 9
        -4312,         // 10
        -4408,         // 11
        4453,          // 12
        8,              // 13
        0,              // 14
        65,             // 15
        1048580,       // 16
        0,              // 17
        -926365496,    // 18
        -926365496,    // 19
        -926365496,    // 20
        51143,          // 21
        261,            // 22
        52494338,      // 23
        17235969,      // 24
        387784214,     // 25
        45693,          // 26
        0,              // 27
        16777236,      // 28
        895687759,     // 29
        302318328,     // 30
        1939,           // 31
        0,              // 32
        0,              // 33
        0,              // 34
        0,              // 35
        0,              // 36
        0,              // 37
        0,              // 38
        0               // 39
    ],
}
```

IDX	Name	Value Range	Unit	Values	Description
0	speed y	0 - 1000	mm/s		Velocity y
1	speed angle	0 - 30000	°/100s		Rotation speed
2	z lift height	0 - 1000	mm		Lift height
3	carrier	0 - 10		0 = EPAL 1 = BCO_GU_1x2_SWI 2 = BCO (Prototyp) 3 = BCO_RO_2x2_SER	currently installed load handling equipment
4	occupied			bit 0 = conveyor 1 occupied bit 1 = conveyor 2 occupied bit 2 = conveyor 3 occupied bit 3 = conveyor 4 occupied	The decimal value of the assignment information for Box Carrier. Each bit stands for the assignment of a conveyor on the box carrier
					<i>Examples:</i> 2 = 0010 ... conveyor 2 occupied 5 = 0101 ... conveyor 1 and conveyor 3 occupied 8 = 1000 ... conveyor 4 occupied

IDX	Name	Value Range	Unit	Values	Description
5	status	0 - 3		0 = normal operation 1 = failure 2 = pre-warning power interruption 3 = initializing	current status
6	mode	0 - 3		0 = off 1 = manual operation mode 2 = maintenance operation mode 3 = automatic operation mode	current operation mode
7	charge	0, 1		0 = vehicle not charging 1 = vehicle charging	charging information
8	soc	0 - 100			current state of charge
9	drive unit fl	-9000 - 90000	°/100		current position (angle) of drive unit front left
10	drive unit fr	-9000 - 90000	°/100		current position (angle) of drive unit front right
11	drive unit rl	-9000 - 90000	°/100		current position (angle) of drive unit rear left
12	drive unit rr	-9000 - 90000	°/100		current position (angle) of drive unit rear right
13	message bits			bit 0 = emergency stop bit 1 = maintenance bit 2 = manual bit 3 = automatic bit 4 = acknowledge lamp bit 5 = acknowledge button bit 6 = start lamp bit 7 = start button bit 8 = du fl recovery bit 9 = du fr recovery bit 10 = du rl recovery bit 11 = du rr recovery bit 12 = custom input bit 13 = box carrier input bit 14 = box carrier contactor	decimal value of message bits 13.0 emergency stop <input type="radio"/> 13.1 maintenance <input type="radio"/> 13.2 manual <input type="radio"/> 13.3 automatic <input checked="" type="radio"/> 13.4 acknowledge lamp <input type="radio"/> 13.5 acknowledge button <input type="radio"/> 13.6 start lamp <input type="radio"/> 13.7 start button <input type="radio"/> 13.8 du fl recovery <input type="radio"/> 13.9 du fr recovery <input type="radio"/> 13.10 du rl recovery <input type="radio"/> 13.11 du rr recovery <input type="radio"/> 13.12 custom input <input type="radio"/> 13.13 box carrier input <input type="radio"/> 13.14 box carrier contactor <input type="radio"/> <i>Example:</i> 8 = 000 0000 0000 1000 = Automatic Mode 4164 = 001 0000 0100 0100 = Manual Mode, Start lamp ON, Customer input ON
14	sensor			bit 1 = fork tip safety bit 2 = fork tip sensor short left bit 3 = fork tip sensor long left bit 4 = fork tip sensor short right bit 5 = fork tip sensor long right	decimal value of sensors fork tips 14.1 fork tip safety <input type="radio"/> 14.2 fork tip sl <input type="radio"/> 14.3 fork tip ll <input type="radio"/> 14.4 fork tip sr <input type="radio"/> 14.5 fork tip lr <input type="radio"/> <i>Example:</i> 8 = 0 1000 = fork tip sensor short right ON 24 = 1 1000 = fork tip sensor short right ON, fork tip sensor long right ON
15	distance_back	0 - 99999	mm		Distance sensor pallet detection

IDX	Name	Value Range	Unit	Values	Description																														
16	protective field			Byte 0: bit 0 = protective field left violated bit 1 = protective field right violated bit 2 = Scanner rotated bit 3 = MicroScan3 bit 4 = warning field 1 left bit 5 = warning field 1 right bit 6 = warning field 2 left bit 7 = warning field 2 right bit 8 = Stop request Byte 2 = current protective field set	decimal value protective field: <i>Example:</i> $1048816 = 1\ 0000\ 0000\ 0000\ 1111\ 0000 =$ <table style="margin-left: 20px; border-collapse: collapse;"> <tr><td>16.0</td><td>protective field left</td><td><input type="radio"/></td></tr> <tr><td>16.1</td><td>protective field right</td><td><input type="radio"/></td></tr> <tr><td>16.2</td><td>protective scanner rot</td><td><input type="radio"/> legacy</td></tr> <tr><td>16.3</td><td>protective scanner</td><td><input type="radio"/> S300</td></tr> <tr><td>16.4</td><td>warn 1 l</td><td><input checked="" type="radio"/></td></tr> <tr><td>16.5</td><td>warn 1 r</td><td><input checked="" type="radio"/></td></tr> <tr><td>16.6</td><td>warn 2 l</td><td><input checked="" type="radio"/></td></tr> <tr><td>16.7</td><td>warn 2 r</td><td><input checked="" type="radio"/></td></tr> <tr><td>16.8</td><td>stop request</td><td><input type="radio"/></td></tr> <tr><td>16.16</td><td>cur monitoring case</td><td>16 station_rear_mutling</td></tr> </table>	16.0	protective field left	<input type="radio"/>	16.1	protective field right	<input type="radio"/>	16.2	protective scanner rot	<input type="radio"/> legacy	16.3	protective scanner	<input type="radio"/> S300	16.4	warn 1 l	<input checked="" type="radio"/>	16.5	warn 1 r	<input checked="" type="radio"/>	16.6	warn 2 l	<input checked="" type="radio"/>	16.7	warn 2 r	<input checked="" type="radio"/>	16.8	stop request	<input type="radio"/>	16.16	cur monitoring case	16 station_rear_mutling
16.0	protective field left	<input type="radio"/>																																	
16.1	protective field right	<input type="radio"/>																																	
16.2	protective scanner rot	<input type="radio"/> legacy																																	
16.3	protective scanner	<input type="radio"/> S300																																	
16.4	warn 1 l	<input checked="" type="radio"/>																																	
16.5	warn 1 r	<input checked="" type="radio"/>																																	
16.6	warn 2 l	<input checked="" type="radio"/>																																	
16.7	warn 2 r	<input checked="" type="radio"/>																																	
16.8	stop request	<input type="radio"/>																																	
16.16	cur monitoring case	16 station_rear_mutling																																	
17	speed x				velocity x																														
18	cell voltage 1			0 - 255 = 2,00 V - 3,60 V	cell voltages 1 - 4																														
19	cell voltage 2			0 - 255 = 2,00 V - 3,60 V	cell voltages 5 - 8																														
20	cell voltage 3			0 - 255 = 2,00 V - 3,60 V	cell voltages 9 - 12																														
21	cell voltage 4			0 - 255 = 2,00 V - 3,60 V	cell voltages 13, 14																														
22	liftcurrent	-20000 - 20000	mA		motor current (lift motor controller) in mA																														
23	firmware				SPS Firmware Version																														
24	protective field version				Protective Field Version																														
25	temperatures	-128 - 127	°C	Byte 0 = battery temperature 1 Byte 1 = battery temperature 2 Byte 2 = battery temperature 3 Byte 3 = ambient temperature	battery and ambient temperature																														
26	battery voltage		mV																																
27	bco signals			bit 0 = gap control L bit 1 = gap control R bit 2 = busy bit 8 - 15 = occupation C0 - C7 bit 16 - 23 = sensor left C0 - C7 bit 24 - 31 = sensor right C0 - C7																															
28	battery type			Byte 0 = battery capacity [Ah] Byte 1 = bms type Byte 2 = cell type Byte 3 = bms firmware version																															
29	motor currents front	-127 bis +127	mA	Byte 0: FL1 Byte 1: FL2 Byte 2: FR1 Byte 3: FR2																															
30	motor currents rear	-127 bis +127	mA	Byte 0: RL1 Byte 1: RL2 Byte 2: RR1 Byte 3: RR2																															
31	drives firmware																																		
32	conveyor weight		dag	Byte 0 & 1: weight conveyor 0 Byte 2 & 3: weight conveyor 1																															

IDX	Name	Value Range	Unit	Values	Description
33	conveyor weight		dag	Byte 0 & 1: weight conveyor 2 Byte 2 & 3: weight conveyor 3	
34	conveyor weight		dag		Res.
35	conveyor weight		dag		Res.
36	weight value		kg	Byte 0 & 1: installed, valid Byte 2 & 3: weight value	optional weight monitoring system

4.8.5 Order Pool

```
curl http://ip-adress:8100/order
```

The returned object shows the last 1.000 order-objects. The first level of indexes are order-ids.

4.8.5.1 Single order

```
curl http://ip-adress:8100/order/130966981865120001
```

The returned object shows the order-object of the order with the id 130966981865120001. Example:

```
"130966981865120001": {
  "recipient": [
    {
      "serialnumber": 13096698
    }
  ],
  "action": [
    {
      "action": "pickup",
      "target": {
        "station": "Boden2"
      }
    },
    {
      "action": "drop",
      "target": {
        "station": "Boden1"
      }
    }
  ],
  "workflow": "0",
  "workflow_name": "STATION    STATION",
  "carrier": "epal",
  "id": 130966981865120001,
  "status": 99,
  "priority": 50,
  "timestamp": 1538117776,
  "created": 1538117654,
  "telemetry": {
    "start": {
      "distance": 100036846,
      "angle": 172022820,
      "operatingtime": 1379391.9916191,
      "carrier": 0,
      "charge": 0,
      "batterylevel": 72,
      "position": [
        29776,
        26572,
        8832,
        16545,
        51,
        0,
        0,
        0,
        0
      ],
      "time": 1538117669,
      "power": -52535.061
    },
    "end": {
      "distance": 100041328,
      "angle": 172060477,
      "operatingtime": 1379498.5431879,
      "carrier": 0,
      "charge": 0,
      "batterylevel": 72,
      "position": [
        32304,
        27204,
        10457,
```

```
17961,
49,
62,
1,
0,
5
],
"time": 1538117776,
"power": -52540.524
},
"delta": {
"distance": 4482,
"angle": 37657,
"operatingtime": 106.55156879988,
"batterylevel": 0,
"power": -5.4629999999961
}
},
"repeat": 1,
"run": 1,
"username": "admin",
"sw_version": "20180928",
"checksum": "f154d8dfd9e4cd1025ebb5421322b263",
"serialnumber": 13096698,
"failure": false,
"stats": {
"pickups": 1,
"stations_checked": 2,
"action": [],
"log": [
{
"start": 1538117669,
"width": 850,
"length": 695,
"actionid": 0,
"action": "pickup",
"station": "Boden2",
"skip": false,
"end": 1538117701
},
{
"start": 1538117701,
"actionid": 1,
"action": "drop",
"station": "Boden1",
"skip": false,
"end": 1538117776
}
],
"actions": {
"pickup": 1,
"drop": 1
}
},
"current_station": "Boden1",
"pickup_drop_done": true
}
```

Attribute	Type	Description
recipient	array of objects	Every item identifies a "serialnumber" or a "collective". If given, an order is addressed to the given group of AGILOX devices.
serialnumber	number	serial number of recipient
collective	array of strings	recipient collective
action	array of objects	Action is a list of actions to be done. Every action-item consists of at least one action (charge, drive, pickup or drop) and additional (sometimes optional) data. Actions are processed sequential in order of presence in the object.
action	string	See Commissioning Manual 4.7.6 - Workflow Action Object
target	object	A station identified by its name or a coordinate with all necessary values can be defined. See Commissioning Manual 4.7.7 - Workflow Target Object
workflow	string	Workflow ID of the origin / If order was started from a workflow
workflow_name	string	Workflow name of the origin
carrier	string	Current carrier type of AGILOX vehicle: <ul style="list-style-type: none"> • "epal": standard, (no load bearing device) • "box1": for box carrier device
id	number	The ID is system-unique and numeric. It is created by a host system or by an AGILOX device. IDs created by AGILOX devices are prefixed with their 8-digit serial number. The following 11 digits are raised for every order. IDs between 1 and 99999999999999999 are available for host-systems.
status	number	Order status: <ul style="list-style-type: none"> • 0 = created (pending) • 1 = active (AGILOX has started order) • 98 = canceled • 99 = finished
priority	number	Orders are being done by priority if queing up. The priority specifies the importance of an order. AGILOX rates every order for the decision, which order to do next. Decision formula: <code>priority * 100 - order_age_in_seconds + distance_to_first_action_in_mm / 100</code> The result is used ascendingly. Default: 50
timestamp	number	Timestamp of last change of order [unix time]
created	number	Timestamp of creation of order [unix time]
telemetry	object	Life time telemetry data of AGILOX vehicle
start	object	Telemetry data at start of order
<i>distance</i>	number	Overall driven distance of lifetime [mm]
<i>angle</i>	number	Overall rotation [°/100]
<i>operatingtime</i>	number	Overall powered on time [sec.]
<i>carrier</i>	number	current load carrier: <ul style="list-style-type: none"> • 0: epal • 1: box1

Attribute	Type	Description
<i>charge</i>	number	Currently on a charging station? • 0: false • 1: true
<i>batterylevel</i>	number	Current battery level
<i>position</i>	array of numbers	Current position of device. Values are: • position sequence number • x [mm] • y [mm] • angle [°/100] • position match [%] • asd match [%] • asd run • actual floor • position speed
<i>time</i>	number	Current time [unix time]
<i>power</i>	number	Current power consumption [Wh]
<i>end</i>	object	Telemetry at end of order
<i>delta</i>	object	Telemetry difference between start and end
repeat	number	How often to repeat the order
run	number	Current run of order
username	string	Username the order was created by
sw_version	number	AGILOX software version to detect API differences
checksum	string	MD5 Checksum
serialnumber	number	Serial number of executing AGILOX vehicle
failure	bool	Its set to true, if a failure occurs, which has to be acknowledged.
stats	object	Statistics:
<i>pickups</i>	number	Count of pick ups
<i>stations_checked</i>	number	Count of stations checked (e.g. for block storage): how many stations checked before finding a suitable one
<i>log</i>	array of objects	While processing the order, AGILOX is logging the following information for each action:
<i>start</i>	number	Start time of action [unix time]
<i>width</i>	number	Measured width of the load carrier [mm] (measured 78mm from the ground)
<i>length</i>	number	Length definition of station [mm]
<i>weight</i>	number	Roughly estimated weight of load [kg]
<i>actionid</i>	number	Action ID
<i>action</i>	string	Action type. See Commissioning Manual 4.7.6 - Workflow Action Object
<i>station</i>	string	Station driven to for this action.
<i>skip</i>	bool	Indicates whether the action got skipped or not.
<i>end</i>	number	End time of action [unix time]
<i>actions</i>	object	Summary of all actions of this order

Attribute	Type	Description
current_station	string	current station <u>or</u> next station if AGILOX is still driving <u>or</u> last station at end of order
pickup_drop_done	bool	Is set to true when executing the first action. If not set, the AGILOX vehicle has not done anything yet and the order can still be canceled without problems.

4.8.5.2 Special order GET endpoints

/order/last	get the last changed 100 orders
/order/open	get all open orders (status 0 or 1), maximum 100
/order/done	get the done orders (status 98 or 99), maximum 100
/order/since/1503643084	get the orders changed since ... unix time
/order/full	get all orders, maximum 10.000
/order/current	get current order of current AGILOX,
/order/cancel/123	cancel order with the id 123, if no id, the current id is used
/order/skip/123	skip current action of order with id 123, if no id, the current id is used

In all cases, if multiple orders and their complete order structure are returned, they are none JSON-conform listed. Instead of a huge object for all orders, the orders are "newline" separated for processing. To show order IDs as well, the **/compat** flag can be used. The reason for this is a limited memory for parsing JSON objects. If objects get too large, processing wouldn't be possible. It is recommended to use the **/gz - flag** to compress the response to a gzip format.

4.8.6 POST Orders - Create/Edit orders

4.8.6.1 Create order



INFORMATION

For most applications a Workflow is posted instead of an order. See [Most Common Application](#) above.

To create an order, a JSON object of the order-definition according to the order-object-description above has to be created. The object can be posted to the web service in the post-body. The web service checks the incoming data for validity and creates the new order, if no error occurs. Example of an order (drive to station) with CURL:

```
curl\\
-d '{"id": 100000010000000000, "action": [{"action": "drive", "target": {"station": "A1"}}]}' \\
-H "Content-Type: application/json" \\
-X POST http://ip-adress:8100/order
```

Examples

In the following examples, only the data structure of the POST command is shown.

Minimal order (drive to station)

```
{
  "id": 100000010000000000,
  "action": [
    {
      "action": "drive",
      "target": {
        "station": "A1"
      }
    }
  ]
}
```

Minimal order (drive to coordinate)

```
{
  "id": 100000010000000000,
  "action": [
    {
      "action": "drive",
      "target": {
        "coordinate": [
          {
            "x": 1000,
            "y": 1000,
            "z": 100, //z and angle are optional.
            "angle": 0
          }
        ]
      }
    }
  ]
}
```

Address one very device to do order

```
{  
  "id": 1000001000000000,  
  "action": [{  
    "action": "drive",  
    "target": {  
      "station": "A1"  
    }  
  }],  
  "recipient": {  
    "serialnumber": 1000001  
  }  
}
```

Address two collectives to do order

```
{  
  "id": 1000001000000000,  
  "action": [{  
    "action": "drive",  
    "target": {  
      "station": "A1"  
    }  
  }],  
  "recipient": {  
    "collective": ["A", "B"]  
  }  
}
```

Complete order for pick up and drop of an EPAL

```
{  
  "id": 1000001000000000,  
  "action": [{  
    "action": "pickup",  
    "target": {  
      "station": "A1"  
    }  
  },  
  {  
    "action": "drop",  
    "target": {  
      "station": "B1"  
    }  
  }]  
}
```

Complete order for pick up and drop of an EPAL with free usable host-data

```
{  
    "id": 100000010000000000,  
    "userdata": {  
        "customernr": 123456,  
        "customername": "AGILOX North America, Inc.",  
        "ordernr": 123456  
    },  
    "action": [{  
        "action": "pickup",  
        "target": {  
            "station": "A1"  
        }  
    },  
    {  
        "action": "drop",  
        "target": {  
            "station": "B1"  
        }  
    }]  
}
```

Complete order with special waiting time

```
{  
    "id": 100000010000000000,  
    "action": [{  
        "action": "pickup",  
        "target": {  
            "station": "A1"  
        }  
    },  
    {  
        "action": "wait",  
        "time": 60  
    },  
    {  
        "action": "drop",  
        "target": {  
            "station": "B1"  
        }  
    }]  
}
```

In this case, the vehicle waits 60 seconds before starting drop-action. Instead of a numeric value, "start" can be given. In case of "start" as wait-value, the vehicle will wait for the start-button being pressed.

4.8.6.2 POST to Update Existing Order

For updating, the order object can be posted again, with the changed data and given id. Be careful, if no id is given, the order will be interpreted as a new order.

The checks are the same as the routine creating orders.

4.8.6.3 POST to Cancel Order

To delete an order, a JSON object with `orderid` and attribute `cancel` has to be posted to the web service. The web service checks the incoming data for validity and cancels the order, if no error occurs.

```
curl\  
  -d '{"id": 100000010000000000, "cancel": "true"}'\  
  -H "Content-Type: application/json"\\  
  -X POST http://ip-adress:8100/order
```

4.8.6.4 POST Results

Return values of post-requests are JSON objects. Every return-object consists at least of the concerned id, and a status. If everything went well, the status is **success** and the response contains an order number. Details (progress, status, telemetry, ...) of the newly created order can be queried via web service call (see [Single Order](#) above).

```
{"id":10000001000010017,"status":"success"}
```

If the status is **error**, the body also contains a list of errors. For example:

```
{"id":false,"status":"error","error":[{"invalid":"id"}]}
```

Possible Errors

limit exceeded	"error": [{"limit_exceeded": "too many pending orders, try again later"}] The maximum amount of orders has been exceeded. The maximum amount of orders is defined as "Amount of AGILOX vehicles in the swarm" x 100.
invalid: recipient	"error": [{"invalid": "recipient"}] A CHARGE or PARK order has been posted, but no serial number or an unknown serial number has been specified. CHARGE and PARK orders need one specified vehicle and are no pool orders. Or an order has been addressed to a nonexistent serial number or collective.
invalid: id	"error": [{"invalid": "id"}] An ID has been posted by a host system, but this id already exists. The same id was assigned twice.
invalid: <variable>	"error": [{"invalid": "station @DESTINATION;Destination;station"}] A variable is missing or wrong.
missing: action	"error": [{"missing": "action"}] There is no action defined in this order.
invalid:action	"error": [{"invalid": "action [xxx]"}] An unknown action has been defined. Possible actions are: pickup, drop, drive, charge, park, drive_over, fire, supply, dispose, drop_here, wait, carrier_pickup, carrier_drop
invalid:repeat	"error": [{"invalid": "repeat"}] A "repeat" has been defined in the order and the value is >1000.
missing:action_target	"error": [{"missing": "action_target", "action": "<action_id>", "action_type": "<action-type>"}] An action needs a target, but no target has been defined. Actions which need a target are: pickup, drop, drive, drive_over, supply, dispose
no valid target found	"error": [{"error": "no valid target found", "action": "<action_id>", "action_type": "<action-type>"}] An invalid target has been defined.
inconsistent station carrier	"error": [{"error": "inconsistent station carrier"}] The target list (e.g. a storage) requires different load handling equipment (Box Carrier, forks).

update_denied:status	"error": [{"update_denied": "status"}] An attempt has been made to update an order id which is already active (status <> 0)
unique order no more cancelable	"error": [{"error": "unique order no more cancelable"}] This "unique" order cannot be updated anymore. If a "unique" order should be updated, the system tries to replace the old one with the new one. This works, even if the order is already active, but ONLY until the first preposition has been reached.
unique order still active	"error": [{"error": "unique order [<order-id>] still active, status [<order-status>]"}] An attempt has been made to create another "unique" order, but a "unique" order with this ID/class already exists.
workflow max pending exceeded	"error": [{"error": "workflow_max_pending [<max_count>] exceeded"}] The amount of "workflow_max_pending" defined in the order object of this workflow has been exceeded.

4.8.7 POST Occupation - Change occupation of stations

To change occupation of stations an occupation object can be posted.

Example:

```
curl\
-d '{"A": true, "B": true, "C": false}'\
-H "Content-Type: application/json"\
-X POST http://ip-adress:8100/occupation
```

4.8.8 Post AGILOX vehicle

A POST to **http://ip-adresse:8100** or **http://ip-adresse:8443** with a POST-Body can be used as an interface to control an AGILOX vehicle or the AGILOX system.

4.8.8.1 Stop and Resume AGILOX vehicles

To stop one or more AGILOX vehicles, a JSON object with the corresponding serial number(s) and the command "stop" can be posted.

The POST Body is designed as an array, to be able to hand over any number of commands within one POST. Which makes it possible to stop several serial numbers at the same time by posting the same command for every serial number.

```
curl\
-d '[{"agiloid": 12345678, "command": "stop"}, {"agiloid": 87654321, "command": "stop"}, ...]'\
-H "Content-Type: application/json"\
-X POST http://ip-adresse:8100
```

To start the AGILOX vehicles again, the command "resume" with the corresponding serial number needs to be posted. The functionality is the same as above.

```
curl\
-d '[{"agiloid": 12345678, "command": "resume"}]'\
-H "Content-Type: application/json"\
-X POST http://ip-adresse:8100
```

4.8.8.2 Set/Reset Toggles

To set one or more toggles via the webservice interface, the following commands can be posted for toggles:

```
curl\  
-d '[{"toggle": "Charge1", "command": "on"}, {"toggle": "Charge2", "command": "off"}, ...]'\\  
-H "Content-Type: application/json"\\  
-X POST http://ip-adresse:8100
```

4.8.8.3 Set/Reset Barriers

Same as for toggles also works with barriers:

```
curl\  
-d '[{"barrier": "Test1", "command": "on"}, {"barrier": "Test2", "command": "off"}, ...]'\\  
-H "Content-Type: application/json"\\  
-X POST http://ip-adresse:8100
```



INFORMATION

This only works properly if the barriers are not automatically activated or deactivated by other conditions as well.

4.8.8.4 Change Occupation of Stations

The occupation of stations can also be changed by a webservice as follows:

```
curl\  
-d '[{"station": "A1", "command": "occupy"}, {"station": "A2", "command": "empty"}, ...]'\\  
-H "Content-Type: application/json"\\  
-X POST http://ip-adresse:8100
```

4.8.8.5 Force Pending Orders to be Next

To force one or more pending orders to be done next regardless any priorities the following command can be posted:

```
curl\  
-d '[{"orderid": "102409901971120001", "command": "force_next"}, {"orderid": "254896532150021548", "command": "force_next"}\,...]'\\  
-H "Content-Type: application/json"\\  
-X POST http://ip-adresse:8100
```

4.8.9 Mapping

For example, to assign a read barcode to a target, the collection /mapping can be used.

```
curl http://ip-address:8100/mapping
```

Example:

```
{
  "target": {
    "NTH123": "A1",
    "SIY459": "{\"stationarea\": [\"A\"]}"
  },
  "x": {
    "NTH123": "C2"
  },
  "version": 7
}
```

"Target" is the first structure level and defines which sub-area of the collection /mapping should be used. The raw key-value relationships are in there.

Any number of sub-areas are allowed.

This can be used in workflows. See Commissioning Manual [4.7.7.3 - Targets from variables](#).

The collection can be changed via a webservice post. Either the complete mapping structure:

```
curl\
-d '{<Complete mapping structure>}'\
-H "Content-Type: application\json"\
-X POST http://ip-address:8100/mapping
```

or a substructure

```
curl\
-d '{"NTH123":"A1", "SIY459": {"\\"stationarea\\": [\"A\"]}}'\
-H "Content-Type: application\json"\
-X POST http://ip-address:8100/mapping/target
```

or just a value

```
curl\
-d '{"A1"}'\
-H "Content-Type: application\json"\
-X POST http://ip-address:8100/mapping/target/NTH123
```

Two hosts should not simultaneously describe the mapping structure with superimposed structures.

5. User Guide



WARNING

Risk of crushing

Carelessness while moving the lifting equipment and picking up transport structures (pallets, etc.) may lead to crushed hands and feet.

- No person is permitted to remain in danger areas while the lifting equipment is moving and the vehicle is backing up
- Be sure to be mindful of acoustic and visual warning signals whenever the lifting drive is active
- Persons are only allowed on the driving path or in the area of movement of AGILOX vehicles, if they are instructed in the vehicle's behavior
- Use proper personal protective equipment when working within the area of movement of AGILOX vehicles

5.1 Create Special Stationareas

5.1.1 Narrow Aisle

For narrow aisle, the AGILOX vehicle can already turn on a defined prepre-position waypoint and parallelly drive to and from the stations of the narrow aisle. Therefore a waypoint outside the narrow aisle is created, which will be used as prepre-position waypoint.

The preposition Y of the station, the prepre-position waypoint and the station must be perpendicular to each other.

The waypoint must be set as prepre-position waypoint in the station menu.

Minimum aisle dimensions, see Operation Manual [ANNEX-Operating Areas](#)



5.1.2 Block Storage

5.1.2.1 General

Block storages can be operated manually or from AGILOX vehicles, from one side only or from both sides. However, there is no manual operation allowed on the side, which is operated by AGILOX vehicles.



WARNING

Risk of being trapped

- If the block storage should be operated manually from both sides, or if the block storage area is regularly visited by persons, the system operator has to plan escape routes to avoid persons being trapped by load carriers.
- For maintenance work in the area of the block storage, the affected channels must be blocked in the HMI, to avoid persons being trapped by load carriers.
To avoid unintentional unblocking of the area before maintenance work is finished, the front stations of the affected channels should also be blocked mechanically (e.g. by safety pylons). AGILOX vehicles detect the occupation and cannot drive into the area.

5.1.2.2 Create Block Storages

All stations of the same channel must have the same preposition Y. For proper operation, the minimum distance between channels must be 200mm.

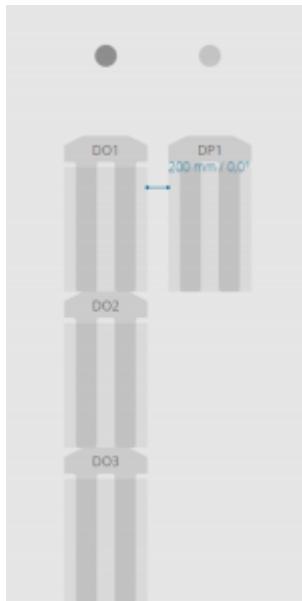
Minimum distance also depends on the load being stored. Higher load might need more space than small compact loads.

Additional requirements, see Operation Manual [ANNEX-Operating Areas](#)

To create a block storage, create the first station, set all configurations and assign it to a stationarea. This station can now be copied and moved by pressing the "Ctrl." key and move it with the cursor. The newly created stations must be placed right behind the first station to create a channel of the block storage. The preposition Y is automatically set the same as the station before.

To create another channel, copy/move the first station to the side. A new preposition Y is created and the channel can be filled as before.

By pressing the "Shift" key, stations can be moved. Be aware that the preposition Y also moves then.



5.2 Control High-Speed Door

As soon as the AGILOX vehicle is near the high-speed door, the signal "**Door Open Request**" is set. With this signal, the high-speed door is opened and as long as the signal is set, the door is kept open.

When the door is completely open, the signal "**Door Open Feedback**" is set by the high-speed door control. Only after this feedback is set, the AGILOX vehicle drives through the door.

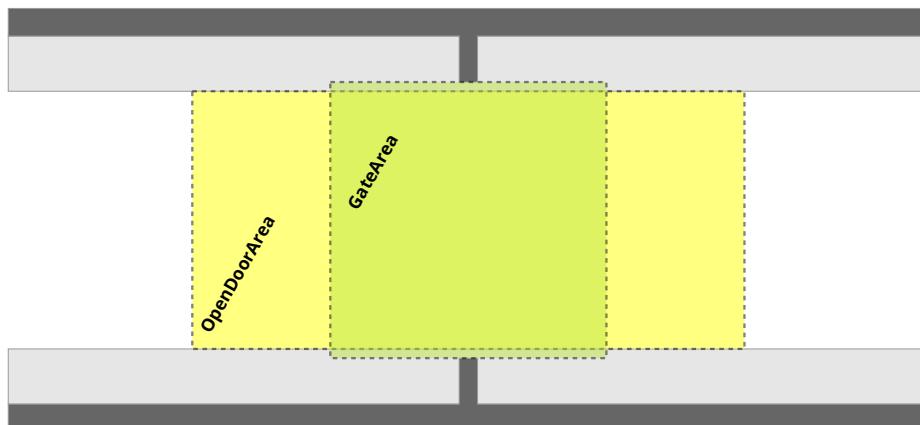
After the AGILOX vehicle has passed the high-speed door and left the area, the signal "**Door Open Request**" is reset after a defined time and the door can be closed automatically again.

5.2.1 Electrical Signals

Address	Name	Communication Direction	Description
OUT0	Door Open Request	AGILOX IO → High-Speed Door Control	0 ... High-speed door control can close the door 1 ... High-speed door control opens the door and keeps it open
INO	Door Open Feedback	AGILOX IO ← High-Speed Door Control	0 ... Door is NOT fully open 1 ... Door is fully open

5.2.2 Configuration AGILOX

AGILOX Map



Configuration SVG Element "OpenDoorArea"

ACTION

```
{  
    "general": {  
        "agiloxio": {  
            "aio1": {  
                "output": {  
                    "0": 1      // OUT0 is set as long as the AGILOX vehicle is on this area.  
                }  
            }  
        }  
    }  
}
```

Configuration SVG Element "GateArea"

ATTRIBUTE

- CLEARWAY (optional, if it is a fire gate)

ACTION

```
{  
    "enter": {  
        "agiloxio": {  
            "aio1": {  
                "input": {  
                    "0": 1      // Only if IN0 is set, the vehicle may enter the area.  
                }  
            }  
        },  
        "enter_distance": 500 // Waiting position 500mm away from the area.  
    }  
}
```

5.3 Crossing with Entry Control

The AGILOX vehicle approaches the crossing area and checks if it is allowed to enter. The approval for the entrance is controlled by the signal "**Entrance allowed**". Once the entrance is allowed, the AGILOX vehicle enters the crossing area and sets the signal "**AGILOX vehicle in the crossing area**". This signal can be used by other systems to check if the entrance is allowed. When the AGILOX vehicle has left the crossing area, the signal "**AGILOX vehicle in the crossing area**" is reset after a defined time.



INFORMATION

If the preposition of a station is in the crossing area, the exit from the station must be locked with the condition `<leave_wait>` in the collection station.

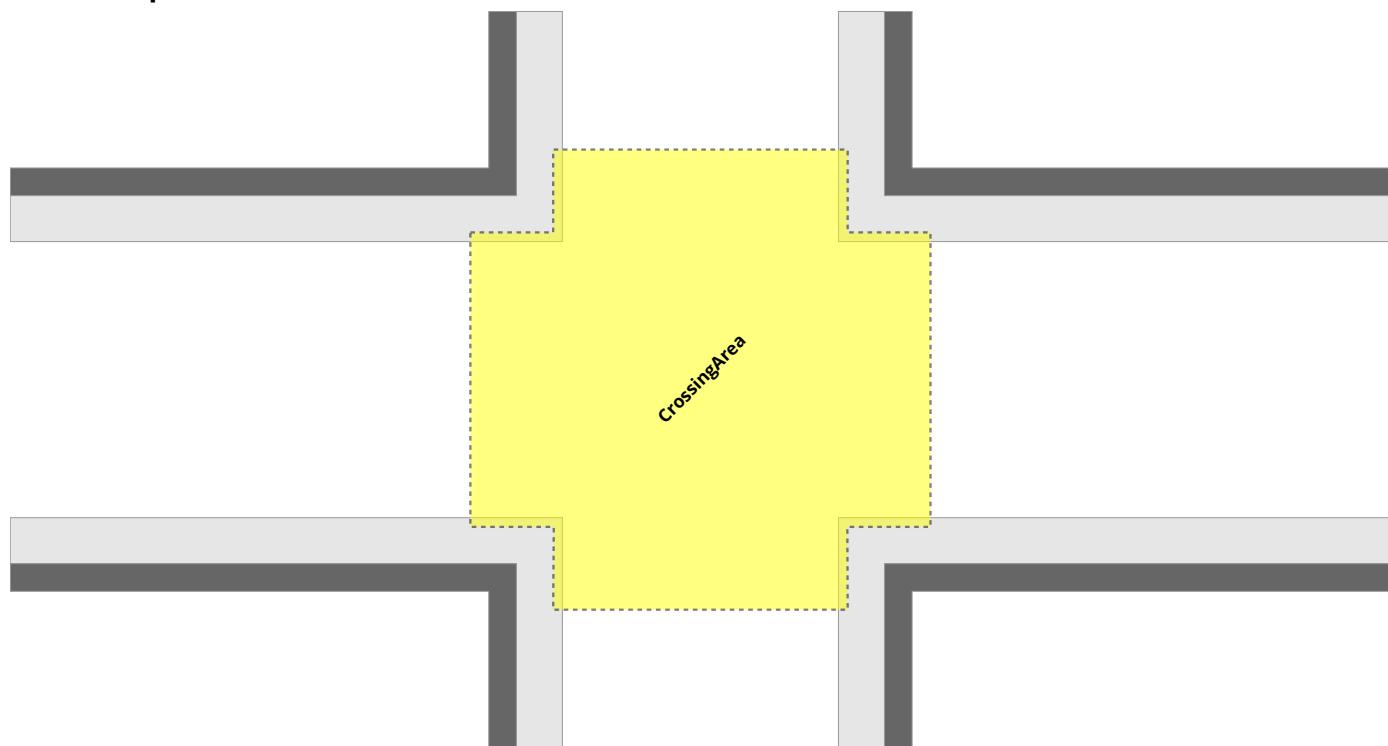
This must be done by AGILOX North America, Inc. or a System Partner.

5.3.1 Electrical Signals

Address	Name	Communication Direction	Description
OUT0	AGILOX vehicle in the crossing area	AGILOX IO → External Control Unit	0 ... no AGILOX vehicle in the crossing area 1 ... at least one AGILOX vehicle in the crossing area
INO	Entrance allowed	AGILOX IO ← External Control Unit	0 ... Entrance prohibited 1 ... Entrance allowed

5.3.2 Configuration AGILOX

AGILOX Map



Configuration SVG Element "CrossingArea"

ACTION

```
{  
    "general": {  
        "agiloxio": {  
            "aio1": {  
                "output": {  
                    "0": 1      // set signal 'AGILOX vehicle in the crossing area'.  
                }  
            }  
        },  
        "enter": {  
            "agiloxio": {  
                "aio1": {  
                    "input": {  
                        "0": 1      // Check, if entrance is allowed.  
                    }  
                }  
            },  
            "enter_distance": 1000 // If entrance is forbidden, the AGILOX vehicle waits 1000mm away from the area.  
        }  
    }  
}
```

5.4 Conveyor station

The following signals are required to drop/pick up pallets at a conveyor station:

- **Station Occupied**

Indicates the occupancy status of the station and is necessary for a correct process flow when dropping or picking up a pallet.

- **Station Ready**

This signal controls the entrance to the conveyor station. This is to prevent the AGILOX vehicle from entering a station which is for example: in trouble, pallet not yet in correct position, ...

- **AGILOX vehicle in Station**

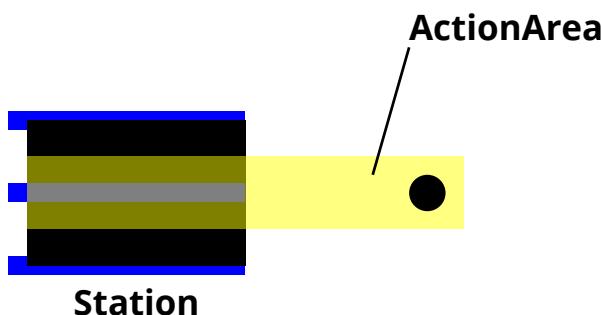
This signal is set as long as the AGILOX vehicle is in the station. Automatic movements (for example: lift table) are blocked.

5.4.1 Electrical Signals

Address	Name	Communication Direction	Description
OUT0	AGILOX vehicle in Station	AGILOX IO → conveyor control	0 ... AGILOX vehicle not in station 1 ... AGILOX vehicle in station
IN0	Station Occupied	AGILOX IO ← conveyor control	0 ... station empty 1 ... station occupied
IN1	Station Ready	AGILOX IO ← conveyor control	0 ... Entrance prohibited 1 ... station ready for dropping or picking up a pallet

5.4.2 Configuration AGILOX

AGILOX Map



Configuration Collection AGILOXIO

```
{  
    "aio1": {  
        "uri": "tcp://xxx.xxx.xxx.xxx:8001/",  
        "input_count": 12,  
        "output_count": 6,  
        "state": {  
            "station_occupied": {  
                "0": 1  
            },  
            "station_ready": {  
                "1": 1  
            }  
        },  
        "command": {  
            "in_station": {  
                "0": 1  
            }  
        },  
        "text": {  
            "input": {  
                "0": "Station occupied",  
                "1": "Station ready"  
            },  
            "output": {  
                "0": "AGILOX vehicle in station"  
            }  
        }  
    }  
}
```

Configuration STATION

- Pallet Detection
deactivate (necessary for conveyor station)
- Occupied Condition
select condition "aio1 - station_occupied".
- StationType
select a stationtype from the dropdown-list (usually station type 2, depending on the structure of the conveyor system).
This must be done by AGILOX North America, Inc. or a System Partner.

Configuration Collection STATION



INFORMATION

This must be done by AGILOX North America, Inc. or a System Partner.

Search for the data object for the conveyor station and add the parameter "condition_station".

```
{  
    ...  
    "Station": {  
        "x": ...,  
        "y": ...  
        ...  
        "condition_station": {  
            "agiloxio": {  
                "aio1": {  
                    "state": {  
                        "station_ready": true // Condition for entering the station.  
                    }  
                }  
            },  
            ...  
        },  
        ...  
    }  
}
```



INFORMATION

As long as the condition is not fulfilled, the AGILOX vehicle stays in the current action. In the workflow a "condition_timeout" can be defined, which triggers an event for this situation.

Configuration SVG Element "ActionArea"

ACTION

```
{  
    "general": {  
        "agiloxio": {  
            "aio1": {  
                "output": {  
                    "0": 1 // set signal 'AGILOX vehicle in station'.  
                }  
            }  
        }  
    }  
}
```

5.5 Pallet Dispensers

A pallet dispenser is parameterized similar to a conveyor station.

In addition, in the case of a pallet dispenser **destacking** or **stacking** must also be initiated. These signals can be set via an action area or through an action in the workflow.

For better error handling, the fill level of the pallet dispenser is also needed (pallet dispenser full, pallet dispenser empty).

5.5.1 Electrical Signals

Address	Name	Communication Direction	Description
OUT0	AGILOX vehicle in Station	AGILOX IO → Pallet Dispenser Control	0 ... AGILOX vehicle not in station 1 ... AGILOX vehicle in station
OUT1	pallet destacking	AGILOX IO → Pallet Dispenser Control	0 → 1 ... initiate destacking pallet
OUT2	pallet stacking	AGILOX IO → Pallet Dispenser Control	0 → 1 ... initiate stacking pallet
IN0	station occupied	AGILOX IO ← Pallet Dispenser Control	0 ... station empty 1 ... station occupied
IN1	station ready	AGILOX IO ← Pallet Dispenser Control	0 ... Entrance prohibited 1 ... station ready for dropping or picking up a pallet
IN2	pallet dispenser empty	AGILOX IO ← Pallet Dispenser Control	0 ... at least 1 pallet in the pallet dispenser 1 ... pallet dispenser empty
IN3	pallet dispenser full	AGILOX IO ← Pallet Dispenser Control	0 ... pallet dispenser not completely full 1 ... pallet dispenser full
IN4	pallet dispenser failure	AGILOX IO ← Pallet Dispenser Control	0 ... pallet dispenser OK 1 ... pallet dispenser failure

5.6 Lift System

5.6.1 Mechanical Requirements

The gap between elevator and hall floor must be less than 10mm.

If it is not possible to reduce the gap to this distance for the total entry width due to constructional reasons, this must be ensured at least in the area of the drive units (wheels) + 50 mm tolerance on both sides.

Not permissible



Permissible

Constructional measures in the area of the drive units



The offset in height to the elevator cabin must be less than +/- 2mm.

During entering the cabin with the customer's maximum transport weight, a height fluctuation of +1mm to -5mm is allowed.



5.6.2 Electrical Requirements

The lift system is included to the AGILOX System via an AGILOX IO.

To AGILOX IO needs a network connection and 24V power supply. As an alternative to the 24VDC supply, an AGILOX IO Box can be used.

The electrical connection (signal exchange) between AGILOX IO and lift control will not be established by AGILOX!

AGILOX only supplies the AGILOX IO or AGILOXIO Box. The integration is carried out by the customer.

Due to the fact that the signal exchange between AGILOX IO and AGILOX vehicle uses WLAN, WLAN connection for the vehicle must be available at any time.

In order to prevent the WLAN connection being interrupted in the cabin or during floor change, it is recommended to mount an access point on or in the lift cabin.

5.6.3 Signal Exchange

For the connection of a lift system, one output signal and one input signal are required per floor. The output signal is the request to drive to a certain floor or call the lift (**lift floor xx - requested**) and the input signal is the feedback to the vehicle that the lift is ready (**lift floor xx - ready**). The process looks like this:

- The AGILOX vehicle drives to the waiting position of the lift
- AGILOX IO sends **lift floor 01 - requested** to lift control.
- Lift moves to floor 01.
- When the lift is in position and the doors are fully opened, the lift control sets the signal **lift floor 01 - ready** and the AGILOX vehicle is allowed to enter.
- As soon as the AGILOX vehicle has entered the lift, the signal **Lift floor 02 - requested** is used to start moving to the destination floor.
- Lift closes the doors and moves on floor 02.
- When the lift arrives in floor 02 and the doors are fully opened, the signal **lift floor 02 - ready** is set by the lift control and the AGILOX vehicle is allowed to exit.

5.6.4 Electrical Signals

Example with 3 floors

Address	Name	Communication Direction	Description
OUT0	Lift floor 01 - requested	AGILOX IO → Lift control	1 ... Lift requested for floor 01
OUT1	Lift floor 02 - requested	AGILOX IO → Lift control	1 ... Lift requested for floor 02
OUT2	Lift floor 03 - requested	AGILOX IO → Lift control	1 ... Lift requested for floor 03
IN0	Lift floor 01 - ready	AGILOX IO ← Lift control	1 ... Lift in floor 01 and ready to enter or leave
IN0	Lift floor 02 - ready	AGILOX IO ← Lift control	1 ... Lift in floor 02 and ready to enter or leave
IN1	Lift floor 03 - ready	AGILOX IO ← Lift control	1 ... Lift in floor 03 and ready to enter or leave

5.6.5 Configuration AGILOX

Configuration LINK

- AGILOX IO settings
 - Select AGILOXIO "aio1"
 - set LEAVE REQUEST and LOCK REQUEST to **Lift floor xx - requested**
 - set LEAVE WAIT CONDITION and LOCK WAIT CONDITION to **Lift floor xx - ready**
- LINK AGILOX IO
 - set LINK station.

5.7 Workflow

5.7.1 Workflow STATION -> STATION

```
{  
    "0": {  
        "name": "STATION    STATION",  
        "status": 0,  
        "order": {  
            "action": {  
                "0": {  
                    "action": "pickup",  
                    "target": {  
                        "station": "@SOURCE;Source;station"  
                    },  
                    "event": {  
                        "no_station_left": "order_cancel"  
                    }  
                },  
                "1": {  
                    "action": "drop",  
                    "target": {  
                        "station": "@DESTINATION;Destination;station"  
                    },  
                    "event": {  
                        "carrier_empty": "order_cancel"  
                    }  
                }  
            }  
        }  
    }  
}
```

5.7.2 Workflow STATION -> AREA

```
{  
    "1": {  
        "name": "STATION    AREA",  
        "status": 0,  
        "order": {  
            "action": {  
                "0": {  
                    "action": "pickup",  
                    "target": {  
                        "station": "@SOURCE;Source;station"  
                    },  
                    "event": {  
                        "no_station_left": "order_cancel"  
                    }  
                },  
                "1": {  
                    "action": "drop",  
                    "target": {  
                        "stationarea": "@DESTINATION;Destination;stationarea"  
                    },  
                    "event": {  
                        "carrier_empty": "order_cancel"  
                    }  
                }  
            }  
        }  
    }  
}
```

5.7.3 Workflow with description

```

{
    "17": {
        "name": "EXAMPLE", // Name: friendly name for the workflow
        "status": "0", // status: 0 = enabled, 99 = disabled
        "canceltime": 3, // time in seconds to cancel the order when using a workflow view
        "order": {
            "priority": 0,
            "repeat": 4, // repeats the order 4 times
            "event": {
                "order_done": {
                    "workflow": {
                        "id": 0,
                        "variables": {
                            "@SOURCE": "A1",
                            "@DESTINATION": "B1"
                        }
                    }
                }
            }
        },
        "start": { // optional start event
            "condition": { // a condition for the start event (so before starting the workflow)
                "operator": "and", // all conditions (machinedata, time, request, distance, ...) must be met
                "machinedata": { // checks the machinedata
                    "item": "batterylevel", // in this case the battery level
                    "value": ">80" // to be above 80%
                },
                "time": { // also checks the time
                    "timespan": [ // in this case two timespans
                        {
                            "start": "07:00", // from 07:00 to 20:30
                            "end": "20:30"
                        },
                        {
                            "start": "21:00", // and from 21:00 - 22:00
                            "end": "22:00"
                        }
                    ],
                    "dayofweek": [ // and also the day of week to be between
                        1, // 1 = Monday and...
                        2,
                        3,
                        4,
                        5 // ...5 = Friday
                    ]
                },
                "agiloxio": { // we're still in the start event, now asking an AGILOX IO
                    "test": {
                        "input": {
                            "2": 1
                        }
                    }
                }
            },
            "action": {
                "0": { // an order needs to do an action
                    "action": "pickup", // e.g. a pick up...
                    "target": {
                        "stationarea": "WEP0" // ...from a stationarea (so it tries to find an epal in that area
                    }
                },
                "event": { // // as we dont know if we'll find an epal there, we define an event for...
                    "no_station_left": "action_skip" // ..."no_station_left", which means we checked all
                }
            }
        },
        "1": {
            "action": "drop", // We wan't to drop that epal at...
            "target": {
                "station": "@SOURCE;Source;stationareastation;AREA51" // Only stations from AREA51 are
            }
        }
    }
}

```

on its own

stations

available to choose from

```

        "condition_station": { // before entering the source station
            "agiloxio": { // we check an input of an AGILOX IO.
                "test": {
                    "input": {
                        "3": 1
                    }
                }
            },
            "condition_timeout": 30, // we set a timeout for "condition_station".
            "event": {
                "carrier_empty": "action_skip2", // skip 2 actions, if the carrier is empty because we
skipped the action before.
                "condition_timeout": "action_done" // we define an event "condition_timeout", which will
be performed if the timeout for condition_station has been reached.
            }
        },
        "2": {
            "action": "drop", // We'll come to here in any case. So we're trying to drop our epal to...
            "target": {
                "station": "Altek-Reserve1" // ... "Altek-Reserve1" when "Altek's" web service did not
allow us to drop (see action above)
            },
            "event": {
                "carrier_empty": "action_skip" // but if we skipped the pickup before, we need to skip
the drop also
            }
        },
        "3": {
            "action": "pickup", // We're trying to pick up an epal at...
            "target": {
                "station": "@TRASH;Station;stationfiltered;/Trash/" // ... "Trash"-station
            },
            "action_skip": { // // but we skip this pickup if...
                "distance": { // ... the distance to ...
                    "station": "Trash", // the station "Trash"
                    "radius": ">3000" // is larger than 3m in radius.
                }
            },
            "event": {
                "no_station_left": "action_skip" // Also skip it, if there's no epal at this station
            }
        },
        "4": {
            "action": "drop", // as we might have picked up a pallet from "Trash" before, we also need
to drop it...
            "target": {
                "station": "Trashdump" // at station Trashdump
            },
            "retry": 180, // if no station found for drop or the only station is occupied,
// retry after 180 seconds. AGILOX will wait 180s.
            "max_retries": 1, // trigger event after 1st retry
            "event": {
                "carrier_empty": "action_skip", // if we skipped the pickup before, we need to skip the
drop as well
                "max_retries": "order_cancel", // cancel order after 1st retry
                "no_station_left": {
                    "agiloxio": { // request to AGILOX IO if no station left
                        "test": {
                            "output": {
                                "2": 1
                            }
                        }
                    }
                }
            }
        },
        "5": {
            "action": "park", // at the end of this workflow, we go to a parking station
            "remain": { // and stay there
                "condition": {
                    "idle": { // until we've reached an idle time of 60 seconds.
                        "time": "<60" // So in other words: we're parking for at least one minute.
                    }
                }
            },
            "mintime": 60, // stay in this action for at least 60 seconds
        }
    }
}

```

```
        "holdtime": 60 // stay in action until the AGILOX vehicle has been standing still for at least 60 seconds
    },
    "6": {
        "action": "drive",
        "target": {
            "coordinate": {
                "x": 10000,
                "y": 10000
            }
        }
    }
}
```

5.7.4 Workflow with weight check

Pick up from source station, if weight >25kg drop to current station, show text and ignite start lamp. If weight is smaller or equal 25kg, drop to target station.

```
{  
  "100": {  
    "name": "EMPTY CHECK",  
    "status": 0,  
    "order": {  
      "action": {  
        "0": {  
          "action": "pickup",  
          "target": {  
            "station": "Sourcestation"  
          }  
        },  
        "1": {  
          "action": "drop_here",  
          "action_skip": {  
            "weight": "<=25"  
          },  
          "event": {  
            "action_done": {  
              "notification": "Empty loadcarrier too heavy"  
            }  
          }  
        },  
        "2": {  
          "action": "drop",  
          "target": {  
            "station": "Targetstation"  
          },  
          "action_skip": {  
            "occupied": false  
          }  
        }  
      }  
    }  
  }  
}
```

5.7.5 Workflow Loop between two Stations

A load is transported back and forth between two stations A and B in a loop.

```
"1000": {
    "name": "LOOP A <-> B",
    "status": 0,
    "order": [
        {
            "repeat": 9999,
            "autoskip": true,
            "recipient": {
                "serialnumber": "@SERIALNUMBER"
            },
            "action": [
                {
                    "0": {
                        "action": "pickup",
                        "target": {
                            "station": "A"
                        }
                    },
                    "1": {
                        "action": "drop",
                        "target": {
                            "station": "B"
                        }
                    },
                    "2": {
                        "action": "drop",
                        "target": {
                            "station": "A"
                        }
                    },
                    "3": {
                        "action": "pickup",
                        "target": {
                            "station": "B"
                        }
                    },
                    "4": {
                        "action": "drop",
                        "target": {
                            "station": "A"
                        }
                    },
                    "5": {
                        "action": "drop",
                        "target": {
                            "station": "B"
                        }
                    }
                }
            ]
        }
    ]
}
```

5.8 Workflow Drop at Conveyor Station

Use Case

Workflow will be started manually. The source station must be selected in a dialoge. After that the pallet will be droped at the conveyor station **ALTEK1**.

Configuration Collection agiloxio

For the conveyor station handling a signal "station occupied" and "station ready" is required.

```
{
  "aio1": {
    "uri": "tcp://10.10.55.201:8001/",
    "input_count": 12,
    "output_count": 6,
    "text": {
      "input": {
        "0": "Fire alarm",
        "1": "ALTEK1 occupied",
        "2": "ALTEK1 ready for load handling"
      }
    },
    "state": {
      "fire": {
        "0": "0"
      },
      "conveyor-ALTEK1-occupied": {
        "1": 0
      },
      "conveyor-ALTEK1-ready": {
        "2": 0
      }
    }
  }
}
```

Configuration Collection station

The occupied signal can be assigned to the station via HMI. The station ready signal needs to be added to the collection station manually.



INFORMATION

This must be done by AGILOX North America, Inc. or a System Partner.

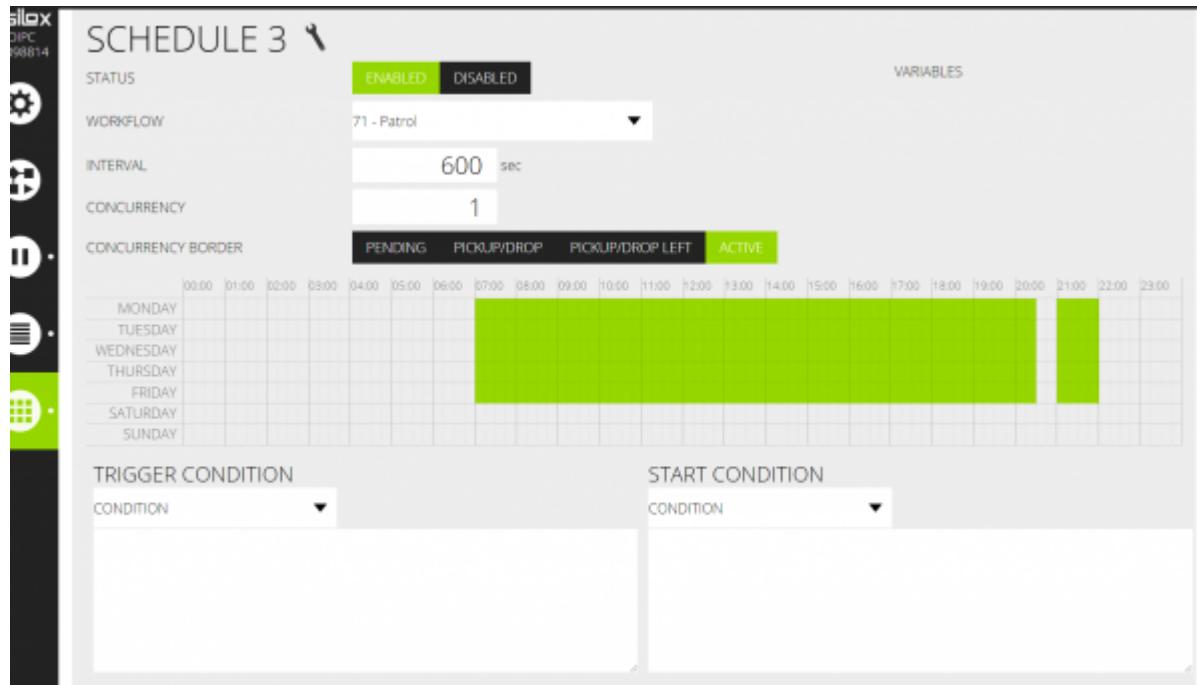
```
"ALTEK1": {
    "x": 55750,
    "y": 23100,
    "z": 440,
    ...
    "occupied": {
        "agiloxio": {
            "aio1": {
                "state": {
                    "conveyor-ALTEK1-occupied": true
                }
            }
        }
    },
    ...
    "condition_station": {
        "agiloxio": {
            "aio1": {
                "state": {
                    "conveyor-ALTEK1-ready": true
                }
            }
        }
    }
}
```

Workflow

```
{  
    "70": {  
        "name": "Drop to ALTEK1",  
        "status": 0,  
        "canceltime": 10, // time in seconds to cancel the order when using a workflow view  
        "order": {  
            "priority": 0,  
            "action": {  
                "0": {  
                    "action": "pickup",  
                    "target": {  
                        "station": "@SOURCE;Quelle;stationareastation;AREA51" // Only stations from AREA51 are  
available  
                    },  
                    "event": {  
                        "no_station_left": "order_cancel"  
                    }  
                },  
                "1": {  
                    "action": "drop",  
                    "target": {  
                        "station": "ALTEK1"  
                    },  
                    "condition_timeout": 50, // time to fulfill the station enter condition (condition defined  
in collection station)  
                    "event": {  
                        "carrier_empty": "order_cancel",  
                        "condition_timeout": "action_skip"  
                    }  
                },  
                "2": {  
                    "action": "drop",  
                    "target": {  
                        "stationarea": "RESERVE"  
                    },  
                    "event": {  
                        "carrier_empty": "order_cancel"  
                    }  
                }  
            }  
        }  
    }  
}
```

5.9 Workflow with Schedule

Schedule



Workflow

```
{
  "71": {
    "name": "Patrol", // friendly name of the workflow
    "status": "0", // status: 0 = enabled, 99 = disabled
    "order": [
      {
        "repeat": 4, // repeat the order 4 times
        "action": [
          {
            "0": { // an order needs to do an action
              "action": "pickup", // e.g. a pickup...
              "target": {
                "stationarea": "WEP0" // ...// ...from a stationarea (so it tries to find an epal in
that area on its own)
              }
            },
            "event": { // as we don't know if we'll find an epal there, we define an event for...
              "no_station_left": "order_cancel" // ..."no_station_left", which means we checked all
stations
              // within that area and could not find an epal. So
we cancel the order.
            }
          }
        ],
        "1": [
          {
            "action": "drop", // we wan't to drop that epal at...
            "target": [
              {
                "station": "Altek" // ...a station called "Altek".
              }
            ],
            "event": [
              {
                "carrier_empty": "order_cancel" // cancel order if carrier empty after previous action
              }
            ]
          }
        ]
      }
    ]
  }
}
```

5.10 Workflow with Barcode

5.10.1 Basic Usage

Enable barcode reading in first action and process barcode as target in second action. It is recommended to switch off the light for normal conditions.

The following example also crops the barcode image to the bottom half. This could be necessary to read only barcodes in bottom 50% of the image. The width and height are the crop size in %, left and top are the position of the crop area.

```
{
  "name": "BARCODE TEST",
  "status": 0,
  "order": [
    {
      "action": {
        "0": {
          "action": "pickup",
          "target": {
            "station": "B1"
          }
        },
        "barcode": {
          "crop": {
            "width": 100,
            "height": 50,
            "left": 0,
            "top": 50
          }
        }
      },
      "1": {
        "action": "drop",
        "target": {
          "station": "@BARCODE"
        },
        "barcode": {
          "type": "code128",
          "length": 2,
          "surrogate": "B2"
        }
      }
    }
  }
}
```

5.10.2 Request mit Barcode

Read barcode and process barcode in a request and use response as target. The response needs to have a valid structure ([4.7.7 - Workflow Target Object](#)). If no barcode was read, the request is done with "NOREAD".

```
{  
    "name": "BARCODE TEST",  
    "status": 0,  
    "order": {  
        "action": {  
            "0": {  
                "action": "pickup",  
                "target": {  
                    "station": "B1"  
                },  
                "barcode": true  
            },  
            "1": {  
                "action": "drop",  
                "target": "@REQUEST",  
                "barcode": {  
                    "type": "code128",  
                    "length": 2,  
                    "surrogate": "NOREAD"  
                },  
                "request": {  
                    "uri": "http://10.10.55.67:8100/test/0",  
                    "post": {  
                        "barcode": "@BARCODE"  
                    },  
                    "surrogate": {"station": ["B2"]},  
                    "json": true  
                }  
            }  
        }  
    }  
}
```

5.11 Workflow with AGILOX JSON-API

5.11.1 Workflow with Target from Host System

A complete model case for host communication with barcodes.

5.11.1.1 Workflow call by host

The host must call an existing Workflow ID, **101** in our example.

The URI for the call is an AGILOX Webservice.

The Workflow has a variable **@SOURCE** defined, which defines the source area for the pick up of a pallet. **A** in this example.

Priority is used to sort orders in the order pool.

userdata is a structure of arbitrary amount of data, which the host service wants to hand over for reference.

http://10.10.55.16:8100/workflow/101

POST-Body

```
{
  "@SOURCE": "A",
  "@PRIORITY": 50,
  "userdata": {
    "host_orderid": 3541387
  }
}
```

This must be done by the host system.

5.11.1.2 Workflow Definition

The Workflow is defined in AGILOX user interface and looks like following JSON structure. The host system does not need know this stucture.

```
{  
    "101": {  
        "name": "Example AGILOX JSON-API",  
        "status": 0,  
        "order": {  
            "priority": 50,  
            "action": {  
                "0": {  
                    "action": "pickup",  
                    "target": {  
                        "stationarea": {  
                            "0": "@SOURCE"  
                        }  
                    },  
                    "barcode": true  
                },  
                "1": {  
                    "action": "drop",  
                    "target": {  
                        "stationarea": ["@REQUEST"]  
                    },  
                    "request": {  
                        "uri": "http://10.10.55.100/getTarget",  
                        "post": {  
                            "host_orderid": "@ORDER.userdata.host_orderid",  
                            "barcodes": "@BARCODE_RAW"  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

- **@SOURCE** comes from the host system.
- **@REQUEST** will be replaced by the respond from the host system after the request.
- **@BARCODE_RAW** will be replaced by the complete barcode structure (barcode types + contents)
- **@ORDER.userdata.host_orderid** will be replaced by the userdata-host_orderid by the host system.

5.11.1.3 AGILOX Pickup (Action 0)

AGILOX does the pickup of the first action. The source is defined by the host system. After the pickup is done and the barcode has been read, the second action will be prepared.

5.11.1.4 Request before drop (Action 1)

AGILOX will perform the request to the host system, which is defined in the Workflow. In this example the request looks as follows:

http://10.10.55.100/getTarget

POST-Body

```
{  
    "host_orderid": 3541387,  
    "barcodes": [  
        {"type": "qr", "data": "135468354"},  
        {"type": "code39", "data": "324136543"},  
        {"type": "code128", "data": "1362456673"},  
        {"type": "itf", "data": "12345"},  
        {"type": "code128", "data": "34523451324"},  
        {"type": "code128", "data": "2345345"},  
        {"type": "code128", "data": "2456456"},  
        {"type": "code128", "data": "24562456"},  
        {"type": "code128", "data": "2345345"}  
    ]  
}
```

This must be done by the Agilox vehicle.

Respond from host system (target)

AGILOX will get following response from the host system as direct answer to the request:

B

B will be used as replacement for **@REQUEST**. Now the order is completely defined and the drop action can be executed. **This must be done by the host system.**

5.11.1.5 Drop (Action 1)

The AGILOX vehicle will drive to area **B** and drop the pallet.

Internal representation of the order

Once the response has been processed, the order looks as follows. The host system does not need to know this structure.

```
{  
    "action": {  
        "0": {  
            "action": "pickup",  
            "target": {  
                "stationarea": {  
                    "0": "A"  
                }  
            },  
            "barcode": true  
        },  
        "1": {  
            "action": "drop",  
            "target": {  
                "stationarea": ["B"]  
            },  
            "request": {  
                "uri": "http://10.10.55.100/getTarget",  
                "post": {  
                    "host_orderid": 3541387,  
                    "barcodes": {  
                        {"type": "qr", "data": "135468354"},  
                        {"type": "code39", "data": "324136543"},  
                        {"type": "code128", "data": "1362456673"},  
                        {"type": "itf", "data": "12345"},  
                        {"type": "code128", "data": "34523451324"},  
                        {"type": "code128", "data": "2345345"},  
                        {"type": "code128", "data": "2456456"},  
                        {"type": "code128", "data": "24562456"},  
                        {"type": "code128", "data": "2345345"}  
                    }  
                }  
            }  
        }  
    }  
}
```

5.11.2 Workflow with Event Based Webservice POST

The AGILOX vehicle should post a feedback to a webservice when a defined action of the workflow is done. In the following example, the AGILOX vehicle picks up a load from a station, drives to a waypoint, posts a feedback to the webservice after it has passed the waypoint and drops the load to another station.

Basically the post to the webservice is possible in any action and can also be triggered by other events (e.g. "target_reached", "station_entered", "station_left").

```
[  
  {  
    "103": {  
      "name": "Test154",  
      "status": 0,  
      "order": {  
        "action": {  
          "0": {  
            "action": "pickup",  
            "target": {  
              "station": "@SOURCE;Source;station"  
            },  
            "event": {  
              "no_station_left": "order_cancel"  
            }  
          },  
          "1": {  
            "action": "drive_over",  
            "target": {  
              "station": "04"  
            },  
            "event": {  
              "action_done": {  
                "request": {  
                  "uri": "http://10.10.55.77:9999/test.php",  
                  "post": {  
                    "Registration": "Entry",  
                    "Order-ID": "@ORDERID"  
                  },  
                  "json": true  
                }  
              }  
            },  
            "2": {  
              "action": "drop",  
              "target": {  
                "station": "@DESTINATION;Destination;station"  
              },  
              "event": {  
                "carrier_empty": "order_cancel",  
                "station_left": {  
                  "request": {  
                    "uri": "http://10.10.55.77:9999/test.php",  
                    "post": {  
                      "Registration": "Exit",  
                      "Order-ID": "@ORDERID"  
                    },  
                    "json": true  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }]
```

5.12 Workflow with supply/dispose

5.12.1 Dispose Workflow with target determination via Barcode

Use Case

A storage should be checked for load carriers by a dispose order. If load carriers have been detected, they should be picked up. The target is written in a barcode. If a failure occurs, e.g. the barcode couldn't be read or the target is not existing, the load carrier should be dropped immediately and the order should be canceled.

Workflow to check SOURCEAREA

```
{  
  "100": {  
    "name": "Check SOURCEAREA",  
    "status": 0,  
    "order": {  
      "priority": 50,  
      "action": {  
        "0": { // Check SOURCEAREA for load carriers ready to be disposed.  
          "action": "dispose",  
          "target": {  
            "stationarea": "SOURCEAREA"  
          }  
        }  
      }  
    }  
  }  
}
```

Dispose Workflow in Collection supply_dispose

```

[

{
    "type": "dispose",
    "auto": true,
    "keep_siblings": false,
    "seperate": true,
    "source": {
        "stationarea": [
            "SOURCEAREA"
        ]
    },
    "workflow": {
        "order": {
            "action": [
                { // Pick up load carrier and read barcode.
                    "action": "pickup",
                    "target": "@SOURCE",
                    "barcode": {
                        "light": false
                    },
                    "event": {
                        "no_station_left": "order_cancel"
                    }
                },
                { // process barcode, determine target and drop at target.
                    "action": "drop",
                    "target": {
                        "stationarea": [
                            "@BARCODE"
                        ],
                        "sort": "station"
                    },
                    "barcode": {
                        "type": "qr",
                        "reverse": true
                    },
                    "event": {
                        "no_barcode": "action_skip", // no barcode found -> skip ACTION.
                        "no_target": "action_skip", // no target found -> skip ACTION.
                        "no_station_left": "action_skip2", // target stations are occupied -> skip ACTION and skip next
ACTION.
                        "action_done": "order_done" // ACTION successfully done -> order done.
                    }
                },
                { // load carrier will be droped at current station and order will be canceled.
                    "action": "drop",
                    "immediate": true,
                    "target": "@SOURCE",
                    "event": {
                        "action_done": "order_cancel"
                    }
                },
                { // Load carrier will be taken back to source.
                    "action": "drop",
                    "target": "@SOURCE",
                    "event": {
                        "carrier_empty": "action_done",
                        "action_done": "order_cancel"
                    }
                }
            ]
        }
    }
]

```

5.13 Dashboard Overview



ATTENTION

Due to recent changes of the web browsers, cookies from embedded unencrypted pages (HTTP) are blocked.

Therefore this dashboard only works if the AGILOX systems have a valid SSL certificate and are connected via HTTPS.

Use Case

For a swarm of AGILOX vehicle a overview of all vehicles in the browser view can be helpful.

Such an overview page can be created using framesets.

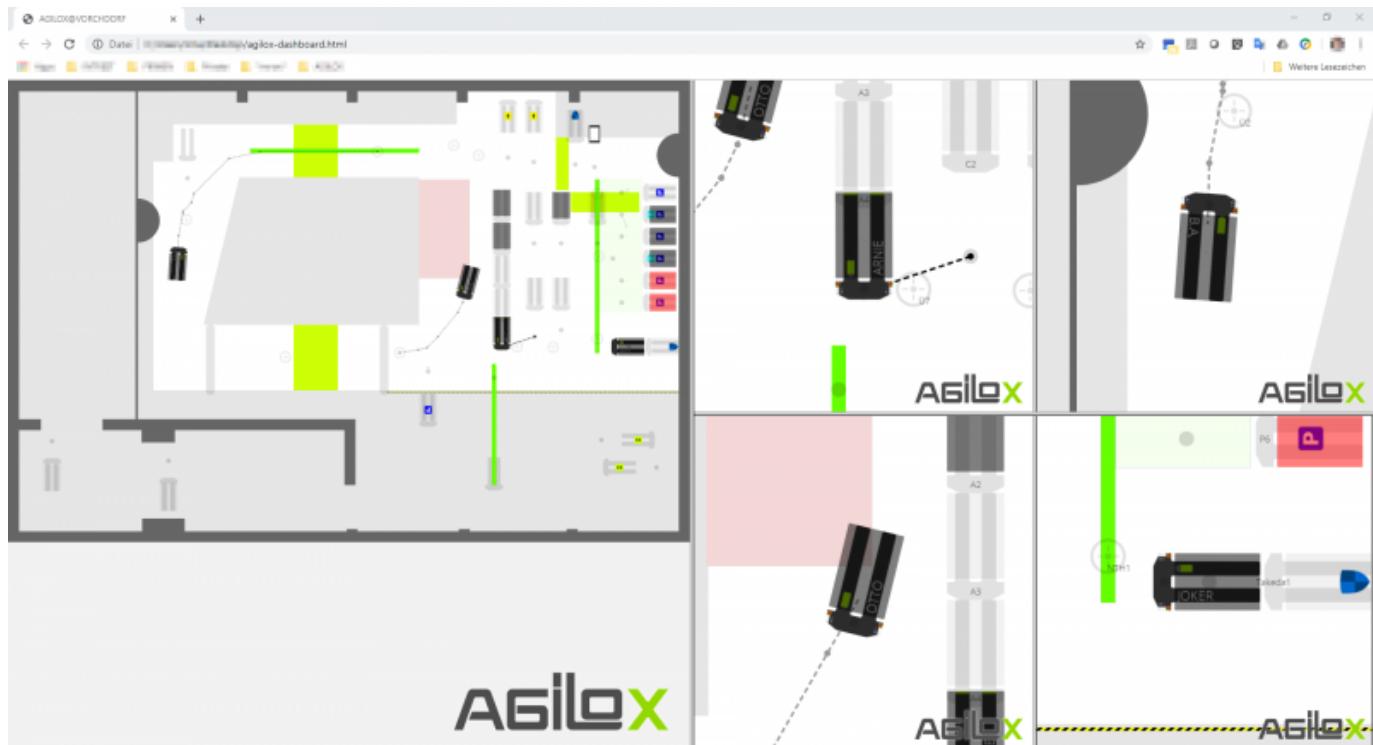
Therefore the windows will be shown within defined frames of the page.

Example for a Dashboard

Following Code needs to be saved in a html file (e.g.: agilox-dashboard.html)

```
<!DOCTYPE html>
<html>
<head>
    <title>AGILOX@VORCHDORF</title>
</head>
<frameset cols="50%, 50%">
    <frame src="https://agipc.production.local/?kiosk"></frame>
    <frameset rows="50%, 50%">
        <frameset cols="50%, 50%">
            <frame src="https://agipc.production.local/?follow=89552943&kiosk"></frame>
            <frame src="https://agipc.production.local/?follow=87584338&kiosk"></frame>
        </frameset>
        <frameset cols="50%, 50%">
            <frame src="https://agipc.production.local/?follow=36703284&kiosk"></frame>
            <frame src="https://agipc.production.local/?follow=13685004&kiosk"></frame>
        </frameset>
    </frameset>
</frameset>
</html>
```

Screenshot of example above



Additional Function

If there are several floors, you can use **?floor** to select the floor for the overview.

Example

```
<frame src="https://agipc.production.local/?floor=2"></frame>
```

5.14 Workflow Dispose Area

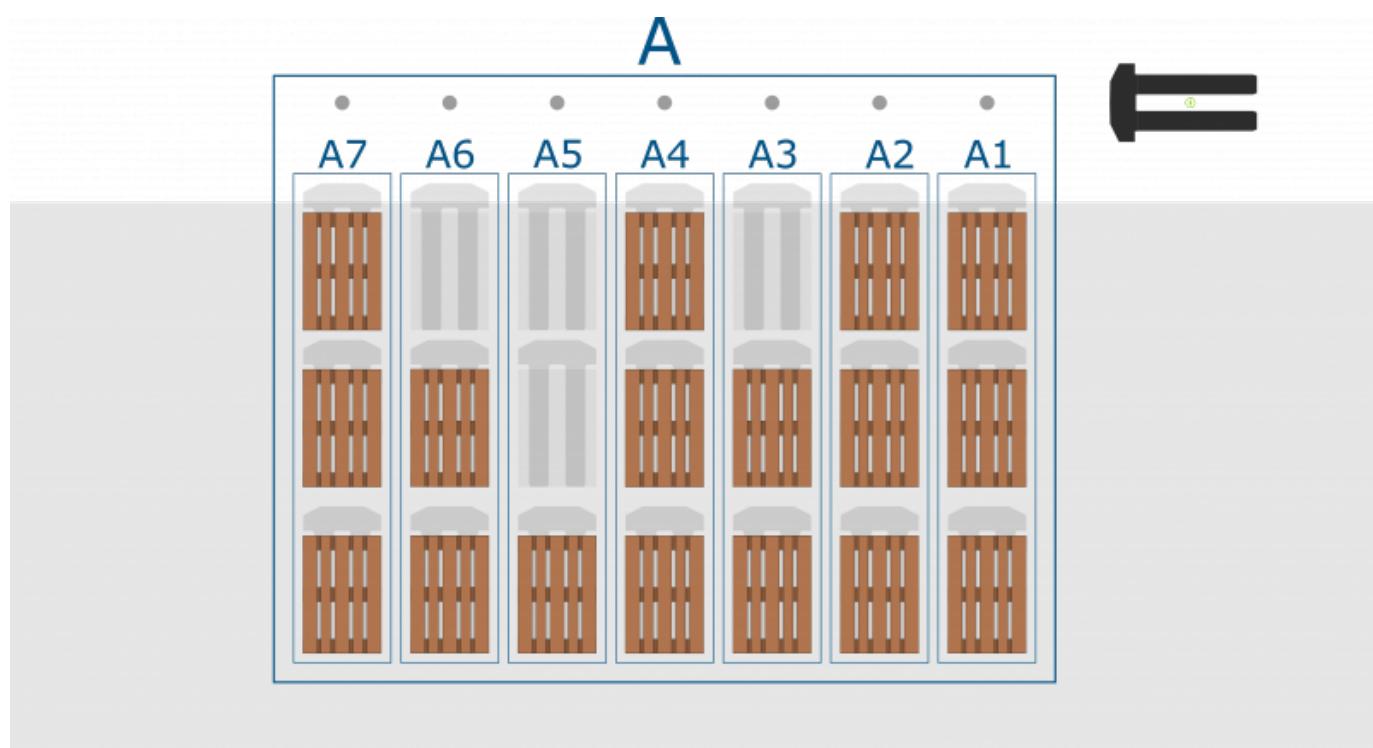
Use Case

An AGILOX swarm should empty a complete block storage with just one command. The block storage is multiple deep and has several channels.

If an error occurs during dropping a load, the carrier should be returned to its channel and the remaining transports from this channel should be stopped.

Block Storage

The block storage in this example consists of 7 channels and each channel is triple deep. Each channel is assigned to stationareas. On the one hand a stationarea that defines the respective channel (e.g.: A1, A2, ...) and on the other hand a stationarea that defines the entire block storage (in our case **A**).



This distribution is necessary so that if an error occurs during transport, only the orders of one channel are cancelled.

Dispose Workflow

The workflow for disposing the area consists of 3 parts. **PART 1:** The workflow for the transport.

PART 2: The supply/dispose definition that calls the transportation workflow.

PART 3: A workflow that checks the occupancy status of the block storage and thereby starts the transportation workflows based on the supply/dispose definition.

PART 1 Transportation Workflow

This workflow consists of a PICKUP and a DROP action. The source object is consigned directly from the supply/dispose definition. The target object in this example is a conveyor station with the name HRL.

```

"1000": {
    "name": "BLOCK STORAGE    HRL",
    "status": 0,
    "order": [
        {
            "action": {
                "0": {
                    "action": "pickup",
                    "target": "@SRC",
                    "event": {
                        "no_station_left": "order_cancel"
                    }
                },
                "1": {
                    "action": "drop",
                    "target": {
                        "station": "HRL"
                    },
                    "event": {
                        "no_station_left": "action_skip",
                        "action_done": "order_done"
                    }
                },
                "2": {
                    "action": "drop",
                    "target": "@SRC",
                    "event": {
                        "action_done": "order_cancel"
                    }
                }
            }
        }
    ]
}
    
```

PART 2 Supply/Dispose Definition

For each channel a dispose definition needs to be entered in the collection supply_dispose. The example for the first channel is as follows.

```

[ {
    {
        "type": "dispose",
        "auto": false,
        "source": {
            "stationarea": [
                "A1"
            ]
        },
        "workflow_start": {
            "id": 1000,
            "variables": {
                "@SRC": "@SOURCE"
            }
        }
    }
]
    
```

PART 3 Checking the block storage area and generating transfer orders

For the transfer orders to be generated, the vehicle must check the block storage area. This is done using a workflow with a DISPOSE action. This workflow can then be started manually or time-/event-controlled.

```
"1001": {
    "name": "Dispose BLOCK STORAGE",
    "status": 0,
    "order": {
        "interrupt": true,
        "autoskip": true,
        "action": {
            "0": {
                "action": "dispose",
                "target": {
                    "stationarea": "A"
                }
            }
        }
    }
}
```

ANNEX - AGILOX Feature Feed

Commissioning Manual Version 2021Q1

20201002

- **Workflow Definition Language**

new attribut "!station" and "!stationarea" in Workflow Target Object
[4.7.5 - Workflow Target Object](#)

20200925

- **Workflow Definition Language**

the variable @BARCODE is now the static last filtered barcode

[4.7.1 - Workflow Variables](#)

new attribut "datamatrix" in barcode object

[4.7.10 - Barcode Object](#)

- **Map**

New attribut "start_on_protective" for action areas

[4.5.5.1 - General Actions](#)

20200911

- **HMI System Settings**

New feature "Greeter".

[4.3.7.1 - System Settings](#)

20200908

- **Map**

New action area attribut: "Weight Limit 1500kg" for AGILOX OCF

[4.5.4.1 - Flags](#)

20200904

- **Workflow Definition Language**

new condition "load_scan_count"

[4.7.5 - Workflow Condition Object](#)

202009xx

- **HMI System Settings**

Order tuning for floor changes

[4.3.7.1 - System Settings](#)

Archive

20200902

- **Workflow Definition Language**

Every condition can be prefixed with a "!" sign to negate the result.

[4.7.5 - Workflow Condition Object](#)

20200818

- **Workflow Definition Language**

The conditions "toggle_on", "toggle_off", "barrier_active" and "barrier_inactive" now allow single values or arrays of items with optional operator.

[4.7.5 - Workflow Condition Object](#)

20200806

- **Map**

Element ID for SVG Elements instead of Element name. Element ID replaces Action ID and must be unique.

Identical elements are no longer automatically combined. This must be done manually in Inkscape.

[4.5.4 SVG Elements](#)

20200702

- **Workflow Definition Language**

Individual stations can be excluded from the target list.

[4.7.7 - Workflow Target Object](#)

Transfer Object to move pallets out of the way, blocking the way to the actually wanted pallet

[4.7.6.1 - Transfer Object](#)

20200618

- **Map**

New conditions for action areas.

[4.5.5.2 - Conditional Actions](#)

20200609

- **HMI System Settings**

A fire alarm condition can be defined.

[4.3.7.1 - System Settings](#)

20200525

- **Workflow Definition Language**

New optional attribut "tags" in Workflow Action Object.

[4.7.6 - Workflow Action Object](#)

20200519

- **Workflow Definition Language**

New Order attribute "deadline_relative".

[4.7.3 Workflow Order Object](#)

20200513

- **HMI Station**

The available-condition of stations has extended functionality. If the condition is not fulfilled, no event is defined and no agiloxio-occupation is defined, the station is emptied. The occupation is no more checked and it is almost handled like a deactivated station.

[4.3.2.1 - Station](#)

20200508

- **HMI Station**

New Option Safe Drop and Safe Drop + .

[4.3.2 - Station](#)

20200429

- **HMI Workflow settings**

New type of workflow view button: "Workflow List"

[4.3.7.3 - Workflow Settings](#)

20200414

- **HMI System Settings**

New language **Danish** now available.

[4.3.7.1 - System Settings](#)

20200409

- **HMI System Settings**

New languages **Chinese** and **French** now available.

[4.3.7.1 - System Settings](#)

- **Workflow Definition Language**

New event "ambiguous_barcode", if more than one barcode left after filtering.

[4.7.4 - Workflow Event Object](#)

New condition "action_agilox_count"

[4.7.5 - Workflow Condition Object](#)

- **Map**

Action areas can now have an id.

[4.5.5.6 - Action ID](#)

20200323

- **JSON API**

New flag "**failure**" in order object, set to true, if a failure occurs, which has to be acknowledged.

[4.8.5.1 - Single Order](#)

- **Workflow Definition Language**

New attribute in Workflow Order Object "**concurrent**".

[4.7.3 - Workflow Order Object](#)

20200316

- **HMI Misc. Tools and Settings**

New button for shutdown/reboot a Host Gateway.

[4.3.7. - Misc. Tools and Settings](#)

- **HMI System Settings**

New Union configurations "Invalid Channel Entering Denied" and "Hardware Button Cancel Time".

[4.3.7.1 - System Settings](#)

20200306

- **Workflow Definition Language**

New conditions **serialnumber** and **collective**

[4.7.5 - Workflow Condition Object](#)

20200304

- **Barcode Typen**

New filter for barcode type **GS1 (former EAN-128/RSS)**

[4.7.10.2 - Barcode Processing Object](#)

20200211

- **HMI Stations**

New option for stations: SAFE DROP

[4.3.2.1 - Station](#)

20200122

- **Workflow Definition Language**

New event **no_carrier**

[4.7.4 - Workflow Event Object](#)

20200115

- **JSON API**

webservice now supports /swarm endpoint

[4.8.3 - General GET endpoints](#)

20200113

- **HMI System Settings**

Changed maximum barcode capture distance from 1000mm to 1200mm

[4.3.7.1 - System Settings](#)

20191212

- **Workflow Definition Language**

New order attribute „create“ to define an order-create condition, where an order gets dismissed with an error, if the condition is not fulfilled.

“condition” no longer necessary for “start” and “create” conditions.

[4.7.3 - Workflow Order Object](#)

20191106

- **HMI System Settings**

Default value for acceleration changed to 350mm/s².

[4.3.7.1 - System Settings](#)

20191030

- **JSON API**

A pending order can be forced to be done next via a webservice command.

[4.8.8.5 - Force Pending Orders to be Next](#)

- **Workflow Definition Language**

Do all Webservice requests asynchronous in background

[4.7.8 - Webservice Request](#)

20191029

- **HMI System Settings**

“MUTE BACKLOCALISATION WHEN OCCUPIED” has been removed, now the AGILOX vehicle mutes dynamically on its own.

Allow inverted custom input check

[4.3.7.1 - System Settings](#)

- **HMI Network tester**

New network interface statistics to check throughput

New webservice / request test interface

[4.3.6.4 - Network Tester](#)

- **JSON API**

New webservice command interface for toggles, barriers and stations

[4.8.8 - POST AGILOX Vehicle](#)

- **Workflow Definition Language**

New handling of following conditions: `condition_station` (former `station_enter`), `condition_rotate` (former `rotate_enter`) and `condition_lift`. After the condition a `condition_timeout` (time in seconds) can be defined, which applies for all of those conditions. The event `condition_timeout` is triggered if this timeout is reached. The conditions “`condition_station`” and `condition_rotate` can also be defined for a station in collection station and the workflow can react to such a condition.

[4.7.6 - Workflow Action Object](#)

[4.7.4 - Workflow Event Object](#)

20190924

- **HMI System Settings**

System setting item for disabling automatic carrier swap.

- [4.3.7.1 - System Settings](#)

- **Workflow Settings**

cancetime/countdown for workflow view buttons may now be defined on button level.

Allow setting a station empty or occupied by just one click on a workflow view button

- [4.3.7.3 - Workflow Settings](#)

- **Workflow Definition Language**

action attribute "transfer_side" to force a box carrier side used to transfer boxes

New attribute "acceleration" to define maximum acceleration for an action.

- [4.7.4 - Workflow Action Object](#)

Webservice Request now processed differently.

- [4.7.8 - Webservice Request](#)

New Events: "station_entered", "station_left", "condition_timeout" and "conveyor_unready". New object for events: "occupation"

- [4.7.4 - Workflow Event Object](#)

- **Map**

New attribute for action areas "no_rotation"

- [4.5.5.1 - Gernerall Actions](#)

New stop attribute for action area general structure: "stop":true

- [4.5.5.2 - Conditional Actions](#)

- **Collections**

New Collection "collective" for assigning vehicles to collectives

- [4.6.3 - collective](#)

- **JSON API**

New CI value "station_approx"

- [4.8.4.1 - Device Overview](#)

Use Webservice to control AGILOX vehicles

- [4.8.8 - POST AGILOX vehicle](#)

- **Vehicle Informationen**

let admin users check barcode camera functionality in maintenance mode

- [4.3.3 - Vehicle Informationen](#)

20190806

- **Workflow Definition Language**

New action event obstruction_timeout

- [4.7.4 - Workflow Event Object](#)

Condition "machinedata" is no longer supported, new conditions "batterylevel" and "z" instead.

- [4.7.5 - Workflow Condition Object](#)

clean order/workflow recipient concept and collective grouping reimplemented

- [4.7.3 - Workflow Order Object](#)

- **supply/dispose**

optional supply_dispose attribute to disable autoskip

- [4.6.8 - Supply/Dispose](#)

- **Map**

action area actions now are able to accept "route_distance" to do actions before entering

- [4.5.5.1 - General Actions](#)

- **HMI System Settings**

new settings "INTERLEAVE ADDITIONAL HULL"

new language: Czech

Collision avoidance fall check can be disabled

- [4.3.7.1 - System Settings](#)

20190715

- **Notification**

e-mail notifications now include information about pending issues

[4.3.6.3 - Notification](#)

20190626

- **Workflow Definition Language**

new order/action event "no_route"

[4.7.4 - Workflow Event Object](#)

20190613

- **Workflow Settings**

new workflow view (easybuttons) concept and automatic migration

[4.3.7.3 - Workflow Settings](#)

- **Map**

new "agilox_invalid" condition

[4.5.5.2 - Conditional Actions](#)

- **Workflow Definition Language**

Optional wait action delayed audio signal.

[4.7.6 - Workflow Action Object](#)

20190529

- **Collections**

Optional attribute "auto_condition" for supply/dispose collection.

[4.6.8 - Supply/Dispose](#)

- **Workflow Definition Language**

New function "workflow_max_pending" in workflow order object.

[4.7.3 - Workflow Order Object](#)

"auto": true in workflow object is no longer supported, use Workflow Schedule instead.

[4.7.2 - Workflow Object](#)

[4.3.7.3 - Workflow Settings](#)

- **System Settings**

Adjustable CI timeout for forgetting „suddenly vanished“ vehicles

New adjustable password policy "MEDIUM"

Adjustable battery plan charge level

[4.3.7.1 - System Settings](#)

New system setting „custom input“ where a custom action can be defined if custom input is set (plc input)

[4.3.7.1 - System Settings](#)

- **User Management**

new special rights feature concept and „local positioning“ as first special right

[4.6.5 - User Management](#)

- **Init Setup**

new internal software distribution concept, if vehicle with newer software version is entering union

[4.2.1 - Union](#)

20190517

- **Map**

new area attribute "IGNORE COMPLETELY" for not using scan data within this areas at all, rename attribute "FREE" to "IGNORE FOR ROUTING"

[4.5.4.1 - Flags](#)

20190506

- **Collections**

New Collection "loadcarrier" analogous to Load Carrier in System Settings.

[4.6.6 - loadcarrier](#)

New Collection "mapping"

[4.6.7 - mapping](#)

- **Special Load Carriers**

Special load carriers with non-parallel feet can be defined.

[4.6.6 - Collection loadcarrier](#)

- **Workflow Definition Language**

Action IDs also allowed with gaps.

[4.7.6 - Workflow Action Object](#)

20190412

- **Workflow Definition Language**

"fire" workflow as optional replacement for default behaviour implemented

[4.7.2 - Workflow Object](#)

20190408

- **Workflow Definition Language**

New condition "targetage" depending on target check age

[4.7.5 - Workflow Condition Object](#)

- **Map**

New "orientation" action area attribute to force agilox orientation while driving through special areas

[4.5.5.3 - Orientation](#)

- **Context Menus**

New option in context menu on vehicle, if vehicle is charging: "Allow Charge Break"

[4.3.1.4 - Context Menu on AGILOX vehicle](#)

20190322

- **Map**

action attribute "obstruction" can now be set to 0 to ignore obstacles completely

[4.5.5.1 - General Actions](#)

20190319

- **Workflow Definition Language**

New optional target sort algorithm for sorting by last pickup/drop on station.

[4.7.7 - Workflow Target Object](#)

- **JSON API and Workflow Definition Language**

new value mapping feature for converting for example a read barcode to a target station, writable by webservice.

[4.8.9 - Mapping](#)

[4.7.7.3 - Targets from variables](#)

20190308

- **HMI System Settings**

System setting for forced login to enter visualization or any other menu items.

System setting for disabling port 80 (http) to force user to use https.

System setting for maximum acceleration

System setting for minimum lift driving height to guarantee to be seen by other third party vehicles

[4.3.7.1 - System Settings](#)

- **Workflow Definition Language**

New Event: "loadcarrier_too_wide"

[4.7.4 - Workflow Event Object](#)

20190215

- **HMI System Settings**

Removed manual clean up buttons for old order removal. Orders are now automatically deleted 60 days after creation to keep database sleek.

[4.3.7.1 System Settings](#)

- **HMI Stations**

Conveyor station creation now only available, if carrier is picked up.

[4.3.2.2 - Conveyor](#)

- **Workflow Definition Language**

Order→active and action→active implemented for custom agiloxio or request executions.

[4.7.3 - Workflow Order Object](#)

[4.7.6 - Workflow Action Object](#)

- **AGILOX IO**

New agilox io trace hmi interface. Expanded agiloxio input and output count to 32.

[4.3.7.5 - AGILOX IO](#)

- **JSON API**

New **/compat** - flag for large order objects.

[4.8.1.3 - Flags](#)

20190201

- **HMI System Settings**

optional station entering current monitoring to detect pushes of unexpected obstacles.
new optional "orange" turn signal color

[4.3.7.1 - System Settings](#)

- **HMI System Settings**

Specific load carriers can be defined, adjusted or deleted in System Settings.

[4.3.7.1 - System Settings](#)

- **Workflow Definition Language**

"target_resume" no longer supported.

[4.7.6 - Workflow Action Object](#)

- **Map**

new action area property „surveillance_distance“ for overriding 7000mm limit

[4.5.5.1 - General Actions](#)

- **Map**

new area "enter" condition and new condition "area_free"

[4.5.5.5 - Enter](#)

- **Workflow Definition Language**

new events "no_barcode" and "overload"

new method "cancel_and_drop"

[4.7.4 - Workflow Event Object](#)

- **Collections**

new collection for heartbeat

[4.6.5 - Collection heartbeat](#)

20190118

- **HMI Workflow Settings**

Workflow Schedule: new concurrency border "pickup/drop left" is triggering after first pickup/drop station has been left

[4.3.7.3 - Workflow Settings](#)

20190117

- **Workflow Definition Language**

new target-type "stationorarea". Maybe be used if lazy or mixture of station/area/station with correct sorting is needed.

[4.7.7 - Workflow Target Object](#)

- **Map**

new area attribute "preferred". Inverse of avoidance light area. (May be used like a track, but without clearing red/orange area!)

[4.5.4.1 - Flags](#)

20190110

- **Workflow Definition Language**

new workflow-variable type "scanner", which allows filtering hand scanner input w/ regex.

[4.7.1 - Workflow Variables](#)

20181211

- **HMI Visualization**

Rooler tool also shows the relative angle now.

20181210

- **Workflow Definition Language**

new variable type "scanner" may be used to ask for a specific AGILOX to do the order.

[4.7.1 - Workflow Variables](#)

20181123

- **HMI Visualization**

rearrangement of Misc. Tools and Settings Menu

[4.3.7 - Misc. Tools and Settings](#)

- **HMI System Settings**

new order_start property "concurrent" to define maximum concurrent orders within one union to be done

[4.3.7.1 - System Settings](#)

- **HMI System Settings**

slider for carrier change for order tuning

[4.3.7.1 - System Settings](#)

- **HMI Visualization**

Complete Conveyor Assignment in Station dialog

station property "announce" implemented to announce target, that there will arrive an AGILOX soon

[4.3.2.2 - Conveyor](#)

- **Map Actions**

ignore foreign vehicles detected on surveillance areas

[4.5.5.4 - Surveillance](#)

- **Workflow Definition Language**

order deadline definition implemented to raise priority of orders with time deadline coming closer

[4.7.3 - Workflow Order Object](#)

- **Workflow Definition Language**

order and action property to ignore specific barriers

[6.5.3 - Workflow Order Object](#)

[4.7.6 - Workflow Action Object](#)

- **Workflow Definition Language**

action "condition_lift" implemented to conditionally wait before lifting for condition to get true
changed "station_enter" and "rotate_enter" to "condition_station" and "condition_rotate"

[4.7.6 - Workflow Action Object](#)

- **JSON API**

added possibility to do order updates of orders created by workflows by calling workflows again with "orderid" in POST body

[4.8.2.3 - POST Workflow](#)

20181102

- **Load Carrier Detection**

for stations exclusively used for load carrier type EPAL tolerant load carrier detection is always used (even if strict load carrier detection in system settings is set to ON)

[4.3.7.1 System Settings](#)

- **Workflow Definition Language**

- Collections Supply/Dispose**

"keep_siblings" option for supply/dispose definitions and orders for being able to not cancel sibling orders on canceling one of these orders automatically (e.g. due to problems).

[4.7.3 - Workflow Order Object](#)

[4.6.8 - Collection supply_dispose](#)

- **Map Actions**

action option "agilox_max_distance" for defining distance to agilox_max area

[4.5.5.1 - General Actions](#)

- **HMI Visualization**

context menu click on station now provides "dispatch station" for creating pickup and drop order easily

[4.3.1.3 - Context Menu on Station](#)

- **HMI System Settings**

system settings option to change led color for blinking before and while bending

[4.3.7.1 - System Settings](#)

- **HMI System Settings**

new central order start condition

[4.3.7.1 System Settings](#)

- **Map**

reduction of map layer count and new automatic "same action" svg element detection for internal consolidation

moved all svg xml tasks to visualization (context menu) to make things easier

[4.5.2 - Map Layers](#)

20181018

- **Collections**

Multi direction barriers implemented

[4.6.2 - Collection barrier](#)

- **Workflow Definition Language**

New functionality of action "wait" for V sign, clean time, start button and conditions

[4.7.6 - Workflow Action Object](#)

- **HMI Stations**

"Available Condition" now also works for charge or park stations

[4.3.2 - Stations](#)

20181017

- **HMI Workflow Settings**

Schedules can now have a friendly name

[4.3.7.3 - Workflow Settings](#)

- **HMI Workflow Settings**

Schedule start-condition (not only trigger condition) can be set via HMI now

[4.3.7.3 - Workflow Settings](#)

20181011

- **HMI Stations**

Link stations now allow specifying a via- and preposition-waypoint

[4.3.2.4 - Link Stations](#)

20181009

- **HMI Visualization**

Actions, directions, visibility and name of SVG areas now adjustable in HMI

[4.3.1.2 - Context Menu on SVG Elements](#)

- **Workflow Settings**

Toggles [for easybuttons] can now be configured in HMI / workflow views

[4.3.7.3 - Workflow Settings](#)

20180928

- **HMI Stations**

New station dialogue for better usability and extended possibilities

[4.3.2 - Stations](#)

- **HMI Workflow Settings**

New workflow scheduler for centralized configuration and extended possibilities

[4.3.7.3 - Workflow Settings](#)

- **Workflow Definition Language**

New target sorting with multi item feature. "Reverse" sort order removed.

[4.7.7 - Workflow Target Object](#)

- **Workflow Definition Language**

New condition "schedule" for exact weekday time schedules

[4.7.5 - Workflow Condition Object](#)

20180921

- **HMI System Settings**

New system setting for enabling warning sound before bend

[4.3.7.1 System Settings](#)

- **HMI Stations**

New station availability condition to conditionally avoid certain stations

[4.3.2 - Stations](#)

- **HMI Stations**

Link stations now are able to use additional "via" stations to define waiting positions before driving to link station finally.

[4.3.2.4 - Link Stations](#)

- **Workflow Definition Language**

Added "sort by distance" for all target sorting methods as second criteria. [4.7.7 - Workflow Target Object](#)

- **Workflow Definition Language**

New conditions toggle_on, toggle_off, barrier_active, barrier_inactive

[4.7.5 - Workflow Condition Object](#)

- **Toggles**

New Collection "toggle". New individual toggles for conditions. Workflow view now is able to show individual toggles for controlling.

[4.6.9 - Collection toggle](#)

[4.7.5 - Workflow Condition Object](#)

[4.3.7.3 - Workflow Settings](#)

20180831

- **HMI System Settings**

New system settings category "Audio Setup" for volume changing and driving back sound choosing
Volume can no longer be changed in Union Configurations.

- 4.3.7.1 [System Settings](#)

- **JSON API**

Added time_idle, time_inactive, ... to /ci data

- 4.8.4.1 - [Device Overview](#)

- **Workflow Definition Language**

New target sort method "concurrency" in Workflow Target Object.

- 4.7.7 - [Workflow Target Object](#)

20180827

- **HMI Reports**

Show Configuration no longer supported

- 4.3.6 - [Reports](#)

- **Global Positioning**

Global Positioning is no longer supported. Use Local Positioning instead. Local positioning now changes floor too, if it is triggered in other floor.

- 4.3.7 - [Misc. Tools and Settings](#)

- 4.3.1 - [Context Menus - Force Local Positioning](#)

- **Workflow Start by AGILOX IO**

New attribute **rising_edge** for workflow start via AGILOX IO.

- [AGILOX IO Operation Manual 5.3.2 - Workflow triggered by AGILO IO](#)

- **Collections**

Allowed admin to modify "barrier" and "floor" collections

- 4.6 - [Collections](#)

20180823

- **HMI Visualization**

New context menus on action areas and barriers. Action creation and modification implemented to visualization for every kind of svg object.

- 4.3 - [Visualization](#)

- **Workflow Definition Language**

"**no_barcode**" is no longer supported in Workflow Event Object.

- 4.7.4 - [Workflow Event Object](#)

- **Workflow Definition Language**

New conditions "**target_occupied**" and "**target_empty**" to replace "**station_occupied**" and "**station_empty**".

- 4.7.5 - [Workflow Condition Object](#)

- **Workflow Definition Language**

New sort methods "**occupied**" and "**empty**" in Workflow Target Object to sort by occupation.

- 4.7.7 - [Workflow Target Object](#)

20180718

- **Barcode Types**

New supported barcode type **Datamatrix**

[4.7.10.2 - Barcode Processing Object](#)

- **JSON API**

New POST to cancel orders

[4.8.6.3 - POST to Cancel Order](#)

- **Workflow Definition Language**

New functionality for Workflow Attribute "**autoskip**"

[4.7.6 - Workflow Action Object](#)

20180711

- **AGILOX IO**

New command "**text**" for collection AGILOX IO. Added possibility to add text descriptions to agiloxio inputs and outputs

[AGILOX IO Operation Manual 5.3 - Collection agiloxio](#)

- **Map**

New attribute "**agilox_max**" for action areas

[4.5.5.1 - General Actions](#)

- **System Settings**

Red and orange areas of route map can no longer be adjusted in Union Configuration.

[4.3.7.1 System Settings](#)

- **Workflow Definition Language**

New attribute "**max_speed**" for workflow definition language

[4.7.3 - Workflow Order Object](#)

[4.7.6 - Workflow Action Object](#)

- **Workflow View**

New blue number on workflow view

[4.3.7.3 - Workflow Settings](#)

20180703

- **Workflow Definition Language**

New attribute "**concurrency**" for Workflow Order Object

[4.7.3 - Workflow Order Object](#)

20180622

- **Workflow Definition Language**

New attribute "**lock_group**" in workflow action object

[4.7.6 - Workflow Action Object](#)

20180608

- **Workflow Definition Language**

New attribute "**lock_immediate**" in workflow action object

[4.7.6 - Workflow Action Object](#)

20180524

- **Directions**

Different direction definition. Hard and Soft directions possible. [4.5.4.2 - Direction](#)