Modula WMS

Think Vertical, Think Modula

Modula-Link Communication Protocol
Technical specifications document

Document version	**2.0 REV 1**
Last change	02/07/2021
Owner	**Beltrami Thomas**

Index

Software info

Current Protocol version: 2.0

WMS Licenses

□ BASE

□ PREMIUM □ REGISTRY □ PICKING □ MANUAL WAREHOUSE □ RF □ IDOC

□ DRIVER ☒ LINK

Introduction

This document explains protocol implemented by Modula Link software

Description

Modula Link is a software that allows a host system to drive a set of Modula (Only LIFT OS MODEL) machines and accessories using a set of commands through a TCP/IP socket channel.

Modula Link is the server of the communications: it opens a TCP port (configurable: by default it's port 11000) and starts to listen, waiting for incoming commands; when it receives a valid command from the host system, it manages it and then it answers to the host system

Modula Link can be configured to manage commands from a single client (Mono-Client Communication) and from multiple clients (Multi-Client Communication).

Mono-Client: WMS accepts only one incoming connection and rejects any other connections by other clients. In front of each command it is processed and provided the answer to the client. The protocol command allows for PREFIX field the value ALL to negotiate the protocol version.

Multi-Client: the commands will be queued and processed with FIFO logic. At the end of the processing of a single command will be given the answer to the requesting client. The protocol command must be called for each bay to negotiate the protocol version, the value ALL is not allowed for PREFIX field for concurrency reason.

***In case of multiple-clients connected do not exist management restrictions Client <-> Machine so nothing avoid to multiple clients to ask the same command at the same time (i.e. the request of the same tray). In a case like this one the first request that is managed will have correct answer and the following ones (queued) will have an error.*

Both connections Modula Link ⍰ host system and Modula Link ⍰ Modula machines are Ethernet connections

Dictionary

Machine	*A single Modula vertical warehouse*
Tray	*A single Modula loading unit where goods can be stored*
Picking operation	*A single physical operation that involves goods stored on a tray* *Common picking operation types are:* - *Pick* - *Deposit* - *Inventory*
Bay	*Machine zone where picking operations are performed.* *Every machine can be provided with up to 3 bay*[1]*.* *Bays are numbered as 1, 2 or 3*
Position	*A single level of a bay where a tray can be moved to.* *Every bay can be provided with up to 2 positions*[2]*.* *Positions are numbered as* - *1 (lower position)* - *2 (upper position)*

Commands structure

Every command sent by host system or by Modula Link is a pipe – separated string using the form

[1] It depends on machine model/structure

[2] It depends on machine model/structure

<PREFIX>|<REQUEST ID>|<COMMAND/ANSWER TYPE>|<PARAMETER 1>|…|<PARAMETER N>
Every message must end with CR (carriage return, 13 ASCII value).

Request ID

Every message is uniquely identified by a <REQUEST ID>.
Note: the value of <REQUEST ID> could not be greater than 2147483647.
Modula Link answer returns same <REQUEST ID> value passed by command received from the host system (handshaking between the two systems)

Prefix

<PREFIX> is a string where last character identifies bay number and remaining characters identify machine number. Its main function is to identify uniquely the machine/bay where the command is directed to
Examples:

- *11: machine 1, bay 1*
- *52: machine 5, bay 2*
- *101: machine 10, bay 1*

Wrong commands management

In case of a wrong command received from the host system, Modula Link answers to host system using one of following special strings

Cause	Message answer sent by Modula Link
Unknown command	BAD_COMMAND
<REQUEST_ID> not present	MISSING_ID
Bad number of parameters	BAD_PARAMETERS
Machine and/or bay not valid	BAD_PREFIX

List of available commands

PROTOCOL command

This command is used to set the communication protocol version. Every client can select the working protocol, but If this command is not used the protocol version is the 1.22.

***Host → Modula Link**: <PREFIX>|<REQUEST_ID>|**PROTOCOL**|<PROTOCOLVERSION>*
***Modula Link → Host**: <PREFIX>|<REQUEST_ID>|**PROTOCOL**|<PROTOCOLVERSION>|<RESULT>*

<PREFIX> not used for this command. It is always 0
<REQUEST_ID> is message ID

< PROTOCOLVERSION> is the version of the protocol to be used to (es. "2.0"). The allowed values are reported in the title of all version of this document. All version before the 2.0 use the same protocol.

< RESULT> is the request response

- *"0" = ok, the protocol is valid*
- *"-1" = Protocol version not supported*

Examples

Host → Modula Link	Modula Link → Host
31|3454|PROTOCOL|2.0	31|3454|PROTOCOL|2.0|0
31|3454|PROTOCOL|2.5	31|3454|PROTOCOL|2.5|1

STATUS command

This command is used to know status of a bay (if bay is available or on error, if there is a tray on lower position or on upper position, …)

Host → Modula Link:

*<PREFIX>|<REQUEST_ID>|**STATUS***

Modula Link → Host:

*<PREFIX>|<REQUEST_ID>|**STATUS**|<STATUS>|<POS1PICKTRAY>|<POS2PICKTRAY>|<POS1EXETRAY>|<POS2EXETRAY>|<ERRORCODE>|<POS1ONEPICKTRAY>|<POS2ONEPICKTRAY>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<STATUS> is bay status

- *"0" = bay ready (on-line, automatic mode active and user logged)*
- *"1" = bay in manual*
- *"2" = bay not ready (off-line or local)*
- *"3" = no operator logged in*
- *"4" = bay engaged*

<POS1PICKTRAY> is number of the picking tray on position 1 (lower position). If no tray is on lower position its value is "0"
<POS2PICKTRAY> is number of the picking tray on position 2 (upper position). If no tray is on lower position its value is "0"
<POS1EXETRAY> is number of the execute tray on position 1 (lower position). If no tray is on lower position its value is "0"

<POS2EXETRAY> is number of the execute tray on position 2 (upper position). If no tray is on lower position its value is "0"
<ERRORCODE> is the actual machine error code, 0 means no errors.
<POS1ONEPICKTRAY> is the number of the tray on position 1 engaged by the OnePick gripper, when 0 the gripper has completed his task and the box is available in his final position.
<POS2ONEPICKTRAY> is the number of the tray on position 2 engaged by the OnePick gripper, when 0 the gripper has completed his task and the box is available in his final position. (NOT AVAILABLE FOR NOW)
It's not a real situation having a tray on lower position and another tray on upper position, so at least one of <TRAY1> or <TRAY2> value will be set to "0"

Examples for normal bay

Host → Modula Link	Modula Link → Host
31|3454|STATUS	31|3454|STATUS|0|3021|0|03021|0|0|0
21|123|STATUS	21|123|STATUS|1|0|0|3021|0|0|0
131|98|STATUS	131|98|STATUS|0|0|0|0|0|0|0
52|11123|STATUS	52|11123|STATUS|0|0|5002|0|5002|0|0
52|11123|STATUS	52|11123|STATUS|0|0|5002|0|5002|5002|0
13|23|STATUS	BAD_PREFIX
11|4576|STATUS|INFO	BAD_PARAMETERS
22||STATUS	MISSING_ID

Examples for One Pick bay

Host → Modula Link	Modula Link → Host
31|3454|STATUS	31|3454|STATUS|0|3021|0|03021|0|0|0
21|123|STATUS	21|123|STATUS|1|0|0|3021|0|0|0
131|98|STATUS	131|98|STATUS|0|0|0|0|0|0|0
52|11123|STATUS	52|11123|STATUS|0|0|5002|0|5002|0|0
13|23|STATUS	BAD_PREFIX
11|4576|STATUS|INFO	BAD_PARAMETERS
22||STATUS	MISSING_ID

CALL command

This command is used to move a tray to a specified position of an bay

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**CALL**|<TRAY>|<POSITION>*

Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**CALL**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<TRAY> is tray number
<POSITION> is position ("1" = lower position; "2" = upper position) where tray must exit
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = tray number not valid*
- *"-2" = position not valid*
- *"-3" = position is busy*
- *"-4" = tray is busy*

- *“-5” = position disable or operator not logged in*
- *“-6” = machine not in automatic mode*

Examples

Host → Modula Link	Modula Link → Host
31|8328|CALL|3001|1	31|8328|CALL|0
22|11123|CALL|4007|2	22|11123|CALL|-1
62|9088|CALL|6005|4	62|9088|CALL|-2
21|3|CALL|1002|1	21|3|CALL|-4
20|1123|CALL|1008|1	BAD_PREFIX
11|77|CALL|12	BAD_PARAMETERS
22||CALL	MISSING_ID

CALLONEPICK command

This command is used to move a tray to a specified position of an bay and request a task to the OnePick grippers. Only one command per each bay can be sent and managed at the same time: is not possible to enqueuer the requests

Host → Modula Link:
*<PREFIX>|<REQUEST_ID>|**CALLONEPICK**|<TRAY>|<POSITION>|<POSX_C>|<POSY_C>|<POSZ_C>|<DIMX_C>|<DIMY_C>|<DIMZ_C>|<POSX_B>|<POSY_B>|<POSZ_B>|<POSID_B>|<TIPOOP>|<GRIP_VALUE>|<DIMX_S>|<DIMY_S>|<DIMZ_S>|<DIMX_B>|<DIMY_B>|<DIMZ_B>*

Modula Link → Host:
*<PREFIX>|<REQUEST_ID>|**CALLONEPICK**|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<TRAY> is tray number

<POSITION> is position ("1" = lower position; "2" = upper position) where tray must exit

<POSX_C> X coordinate of the compartment from the upper left corner in mm (must be greater than zero)

<POSY_C> Y coordinate of the compartment from the upper left corner in mm (must be greater than zero)

<POSZ_C> Z coordinate of the compartment from the upper left corner in mm (can be zero)

<DIMX_C> X dimension of the compartment in mm (must be greater than zero)

<DIMY_C> Y dimension of the compartment in mm (must be greater than zero)

<DIMZ_C> Z dimension of the compartment in mm (can be zero)

<POSX_B> X coordinate of the location in bay from the upper left corner in mm
(set to zero if the position id is used)

<POSY_B> Y coordinate of the location in bay from the upper left corner in mm
(set to zero if the position id is used)

<POSZ_B> Z coordinate of the location in bay from the upper left corner in mm
(set to zero if the position id is used)

<POSID_B> position id in bay (set to zero if the coordinates are used)

<TIPOOP> operation type: "V" = replenishment, "P" = pick up

<GRIP_VALUE> grip strength value in mm, it indicates the excursion of the grippers to tighten the box

<DIMX_S> X dimension of the Item in mm (must be greater than zero)

<DIMY_S> Y dimension of the Item in mm (must be greater than zero)

<DIMZ_S> Z dimension of the Item in mm (must be greater than zero)

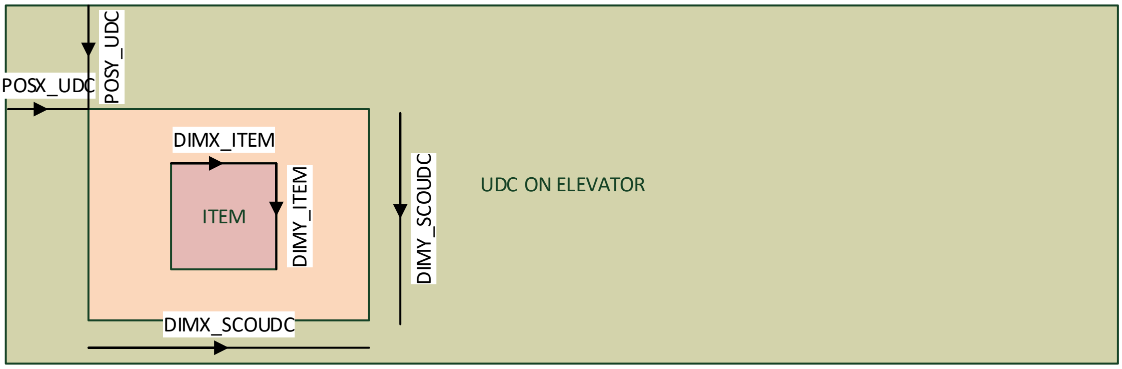
<DIMX_B> X dimension of the Position Bay in mm (set to zero if the position id is used)

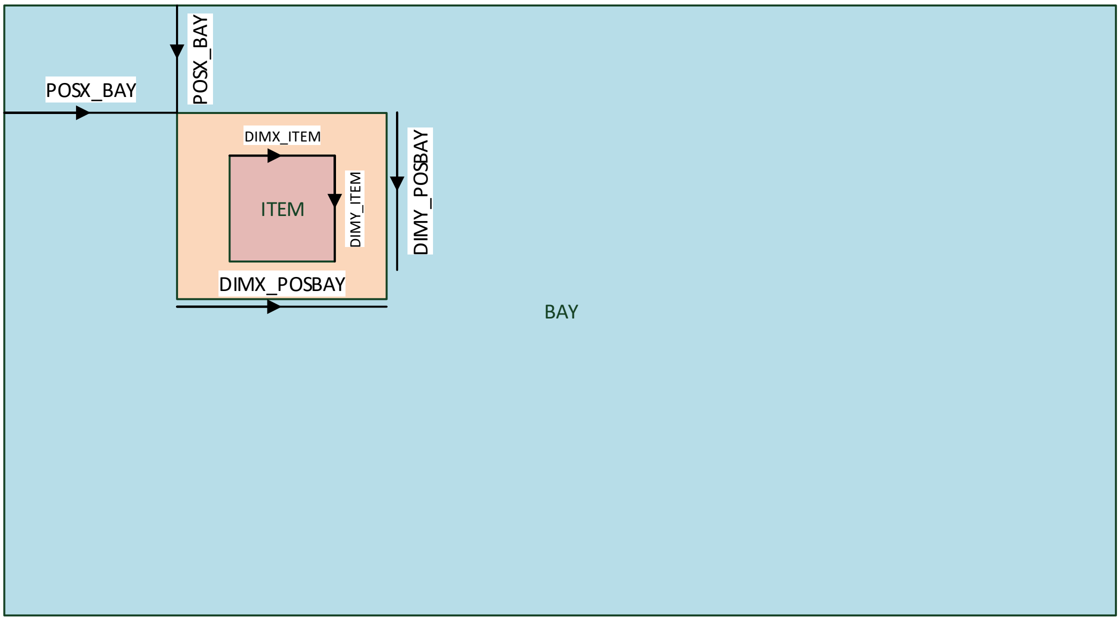
<DIMY_B> Y dimension of the Position Bay in mm (set to zero if the position id is used)

<DIMZ_B> Z dimension of the Position Bay in mm (set to zero if the position id is used)

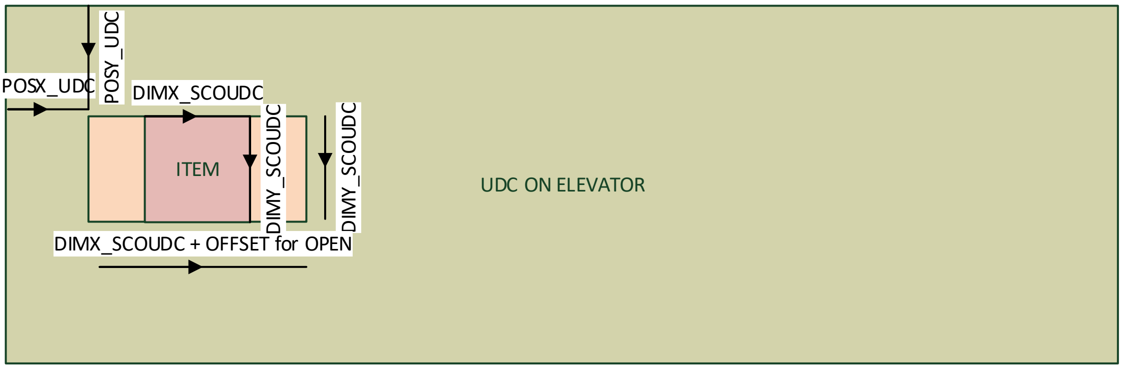
<RESULT> is result of the request. Possible values are:

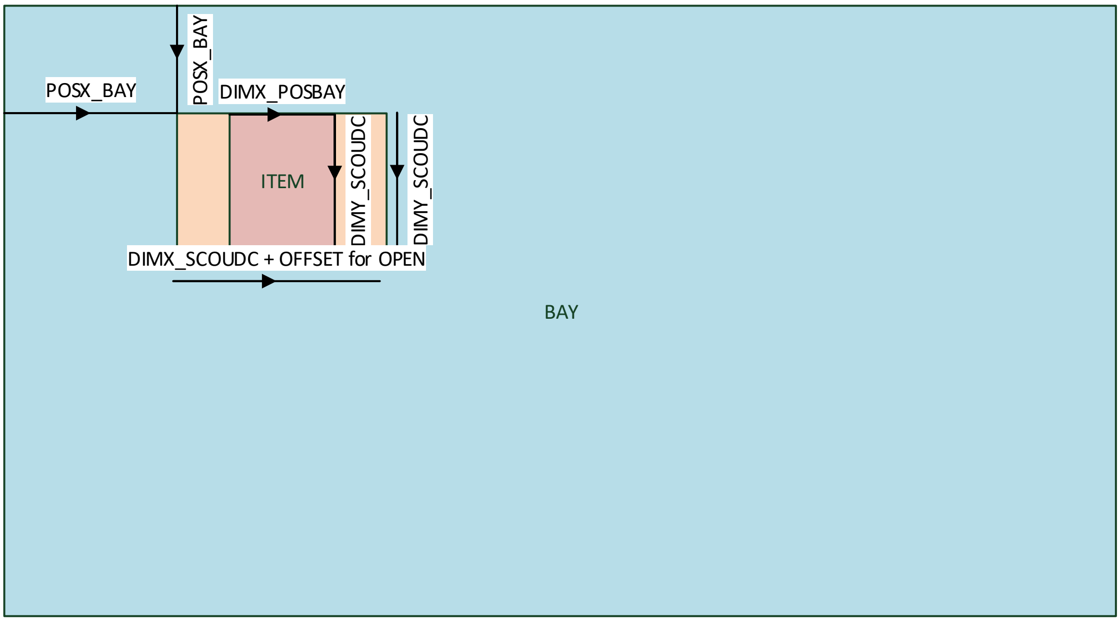
- *"0" = ok*
- *"-1" = tray number not valid*
- *"-2" = position not valid*
- *"-3" = position is busy*
- *"-4" = tray is busy*
- *"-5" = position disabled or operator not logged in*
- *"-6" = one pick gripper is busy*
- *"-7" = compartment coordinates not valid (for example is less than 0)*
- *"-8" = compartment dimensions not valid (for example is less than 1)*
- *"-9" = coordinate of the box in bay not valid (for example is less than 0)*
- *"-10" = position id in bay not valid (for example is less than 0 or not existing position ID)*
- *"-11" = grip strength value not valid (for example is less than 0)*
- *"-12" = box dimension not valid (for example is less than 1)*
- *"-13" = position dimensions in bay not valid (for example is less than 0)*
- *"-14" = operation type not valid (is different from "V" or "P")*


POSY_UDC
POSX_UDC
DIMX_ITEM
DIMY_ITEM
ITEM
DIMY_SCOUDC
UDC ON ELEVATOR
DIMX_SCOUDC

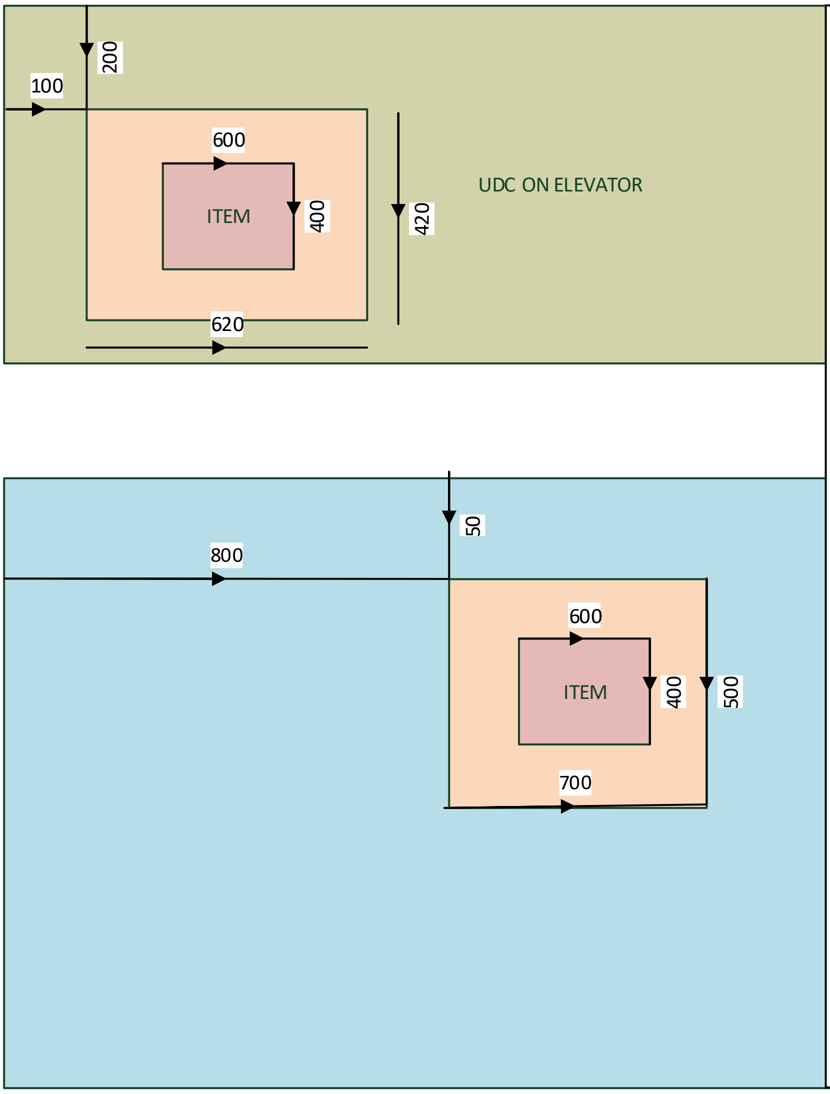

POSX_BAY
POSX_BAY
DIMX_ITEM
DIMY_ITEM
ITEM
DIMY_POSBAY
DIMX_POSBAY
BAY

If DIMX_ITEM =0 and DIMX_POSBAY=0:





Examples



PREFIX = 31
REQUEST_ID = 8328
CALLONEPICK
TRAY = 3001
POSITION = 1
POSX_C = 100
POSY_C = 200
POSZ_C = 0
DIMX_C = 620
DIMY_C = 420
DIMZ_C = 0
POSX_B = 800
POSY_B = 50
POSZ_B = 0
POSID_B = 0
TIPOOP = V
VALPRESA = 8
DIMX_S = 600
DIMY_S = 400
DIMZ_S = 73
DIMX_B = 700
DIMY_B = 500
DIMZ_B = 0

Host → Modula Link	Modula Link → Host
31|8328|CALLONEPICK|3001|1|100|200|0|620|420|0|800|50|0|0|V|8|600|400|73|700|500|0	31|8328|CALLONEPICK|0
31|8329|CALLONEPICK|3002|1|100|200|0|620|420|0|800|50|0|0|V|8|600|400|73|700|500|0	22|8329|CALLONEPICK|-6
20|8330|CALLONEPICK|2000|1|100|200|0|620|420|0|800|50|0|0|V|8|600|400|73|700|500|	BAD_PREFIX
11|77|CALLONEPICK|12	BAD_PARAMETERS
22||CALLONEPICK	MISSING_ID

RETURN command

This command is used to move a tray out of a position and return it to its cell inside the machine.
NB: The RETURN command must be called every time that a tray change is needed. To do a call command without sending back the previous tray will cause the call failure.

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**RETURN**|<POSITION>*
Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**RETURN**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<POSITION> is position ("1" = lower position; "2" = upper position) from where the tray return to its cell
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = empty position*
- *"-2" = position not valid*
- *"-100" = generic error (see the WMS logs)*

Examples

Host → Modula Link	Modula Link → Host
21|11|RETURN|1	21|11|RETURN|0
91|71217|RETURN|6	91|71217|RETURN|-2
62|9|RETURN|2	62|9|RETURN|-1
20|2|RETURN|1	BAD_PREFIX
11|666|RETURN	BAD_PARAMETERS
71||RETURN	MISSING_ID

LASER_ON command

This command turns on laser pointer of a specified bay (if any)

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**LASER_ON***

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**LASER_ON**|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = laser not available or on error*

Examples

Host → Modula Link	Modula Link → Host
21|11|LASER_ON	21|11|LASER_ON|-1
81|71217|LASER_ON	18|71217|LASER_ON|0
M2|2|LASER_ON	BAD_PREFIX
11|666|LASER_ON|1	BAD_PARAMETERS
11||LASER_ON	MISSING_ID

LASER_OFF command

This command turns off laser pointer of a specified bay (if any)

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**LASER_OFF***

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**LASER_OFF**|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = laser not available or on error*

Examples

Host → Modula Link	Modula Link → Host
32|11|LASER_OFF	32|11|LASER_OFF|0
21|71217|LASER_OFF	21|71217|LASER_OFF|-1
87|34|LASER_OFF	BAD_PREFIX
11|666|LASER_OFF|1|1|1	BAD_PARAMETERS
72||LASER_OFF	MISSING_ID

LASER_HOME command

This command performs a homing of laser pointer of a specified bay and execute a device calibration (if any)

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**LASER_HOME***

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**LASER_HOME**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = laser not available or on error*

Examples

Host → Modula Link	Modula Link → Host
51|1111|LASER_HOME	51|1111|LASER_HOME|0
31|71|LASER_HOME	31|71|LASER_HOME|-1
30|1034|LASER_HOME	BAD_PREFIX
11|686|LASER_HOME|24	BAD_PARAMETERS
41||LASER_ HOME	MISSING_ID

LASER_GO command

This command moves laser pointer of a specified bay (if any) to a specified point (X, Y)

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**LASER_GO**|<POSITION>|<X>|<Y>*

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**LASER_GO**|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<POSITION> is position ("1" = lower position; "2" = upper position) where laser cursor must be focused to

<X> is horizontal coordinate (millimeters)

<Y> is vertical coordinate (millimeters)

<RESULT> is result of the request. Possible values are:

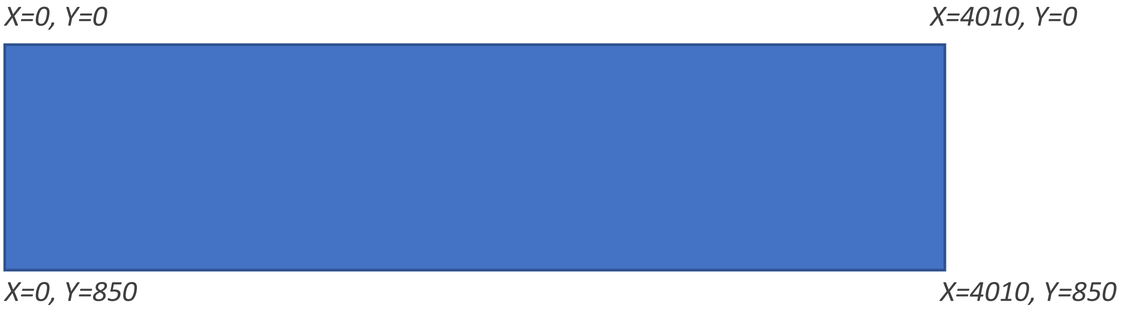
- *"0" = ok*
- *"-1" = laser not available or on error*
- *"-2" = wrong coordinates (X and/or Y set to wrong values)*

Examples

Host → Modula Link	Modula Link → Host
51|1111|LASER_GO|1|1502|230	51|1111|LASER_GO|0
31|71|LASER_GO|2|0|0	31|71|LASER_GO|-1
31|71|LASER_GO|2|9000|-120	31|71|LASER_GO|-2
30|1034|LASER_GO|1	BAD_PREFIX
11|686|LASER_GO|24	BAD_PARAMETERS
41||LASER_GO|2|100|100	MISSING_ID

X and Y coordinates refer to upper/left corner of the tray

For example for Model MLD:



If you have configured UDC height the laser doesn't point to the tray bottom but on tray edge

LASER_STATUS command

This command is used to know status of laser

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**LASER_STATUS***

Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**LASER_STATUS**|<STATUS>|<POSITION>|<X>|<Y>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<STATUS> Laser Status:
IDLE, (not initialized)
STOP, (stopped)
STOP_ON, (stopped and light on)
STOP_OFF, (stopped and light off)
MOVE, (in movement)
ERROR, (in error communication WMS<->Laser or Laser device)
NOT_CONNECT (laser disconnect)

<POSITION> is position ("0" = no position; "1" = lower position; "2" = upper position) where laser cursor is focused to:

- *<X> is horizontal coordinate (millimeters)*
- *<Y> is vertical coordinate (millimeters)*

Examples

Host → Modula Link	Modula Link → Host
21|11|LASER_STATUS	21|11|LASER_STATUS|IDLE|0|0|0
81|71217|LASER_STATUS	81|71217|LASER_STATUS|MOVE|0|0|0
81|71217|LASER_STATUS	81|71217|LASER_STATUS|STOP|1|-1000|200
81|71217|LASER_STATUS	81|71217|LASER_STATUS|STOP_ON|1|-1000|200
99|71217|LASER_STATUS	BAD_PREFIX
11|666|LASER_STATUS|1	BAD_PARAMETERS
11||LASER_STATUS	MISSING_ID

DISPLAY_CLEAR command

This command clears a specified alphanumeric display of a specified bay (if any)

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**DISPLAY_CLEAR***

Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**DISPLAY_CLEAR**|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = display not available or error*

Examples

Host → Modula Link	Modula Link → Host
51|1111|DISPLAY_CLEAR	51|1111|DISPLAY_CLEAR|0
31|71|DISPLAY_CLEAR	31|71|DISPLAY_CLEAR|-1
30|1034|DISPLAY_CLEAR	BAD_PREFIX
11|686|DISPLAY_CLEAR|0|0|0|0	BAD_PARAMETERS
41||DISPLAY_CLEAR	MISSING_ID

DISPLAY_SHOW command

This command shows a message on an alphanumeric display of a specified bay (if any).

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**DISPLAY_SHOW**|<MESSAGE>|<COL>|<ARROW>*

Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**DISPLAY_SHOW**|<MESSAGE>|<RESULT>*

<PREFIX> is bay/machine identifier

<REQUEST_ID> is message ID

<MESSAGE> is text message

<COL> is display starting led column

*On display are present 38 columns in 10 cm. So, on a display long 4000 mm (ML machine) 38*40=1520 columns*

<ARROW> is a parameter for enabling ("1") or disabling ("0") the display of an arrow

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = display not available or error*

Examples

Host → Modula Link	Modula Link → Host
51|1111|DISPLAY_SHOW|Hello|300|1	51|1111|DISPLAY_SHOW|0
31|71|DISPLAY_SHOW|My message|140|0	31|71|DISPLAY_SHOW|-1
30|1034|DISPLAY_SHOW|Test|1|1|	BAD_PREFIX
11|686|DISPLAY_SHOW|Qty = 30	BAD_PARAMETERS
41||DISPLAY_SHOW|Bye|400|1	MISSING_ID

PTL_SHOW_QTA command

This command shows quantity and relative colour in a Put to Light's display.

***Host → Modula Link:** <PREFIX>|<REQUEST_ID>|**PLT_SHOW_QTA**|<ID_DISPLAY>|<QTA>|<COLOR>*

***Modula Link → Host:** <PREFIX>|<REQUEST_ID>|**PTL_SHOW_QTA**|< RESULT>*

<PREFIX> in this message is not managed the bay and the machine so always insert '00'

<REQUEST_ID> is message ID

<ID_DISPLAY> display id

<QTA> Quantity to be displayed

<COLOR> Color (R:Red, O:Orange , G:Green)

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = PTL not available or error*

Examples

Host → Modula Link	Modula Link → Host
00|1111|PTL_SHOW_QTA|101|10|R	00|1111|PTL_SHOW_QTA|0
00|71|PTL _SHOW_QTA|102|10|Y	00|71|PTL_SHOW_QTA|-1
99|1034|PTL _SHO_QTA|102|1|R	BAD_PREFIX
00|686|PTL _SHOW_QTA|102	BAD_PARAMETERS
00||PTL_SHOW_QTA|102|3|R	MISSING_ID

MODULA LINK MANAGE ONLY ONE PUT TO LIGHT GROUP

PTL_SHOW_MESSAGE command

This command shows a message with relative color in a Put to Light's display.

Host → Modula Link:

*<PREFIX>|<REQUEST_ID>|**PLT_SHOW_MESSAGE**|<ID_DISPLAY>|<MESSAGE>|<COLOR>*

Modula Link → Host:

*<PREFIX>|<REQUEST_ID>|**PTL_SHOW_ MESSAGE**|<RESULT>*

<PREFIX> in this message is not managed the bay and the machine so always insert '00'

<REQUEST_ID> is message ID

<ID_DISPLAY> display id

<MESSAGE> Message to be displayed: for example 'F' (free), 'C' (complete), 'I' (incomplete)

**(is possible to display only message compatible with a 7-segments display and long 4 chars at most)*

<COLOR> Colour (R:Red, O:Orange , G:Green)

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = PTL not available or error*

Examples

Host → Modula Link	Modula Link → Host
00|1111|PTL_SHOW_MESSAGE|101|F|R	00|1111|PTL_SHOW_MESSAGE|0
00|71|PTL _SHOW_MESSAGE|102|F|R	00|71|PTL_SHOW_MESSAGE|-1
00|1034|PTL _SHO_MESSAG|102| F|R	BAD_PREFIX
00|686|PTL _SHOW_MESSAGE	BAD_PARAMETERS
00||PTL_SHOW_MESSAGE|102|F|R	MISSING_ID

MODULA LINK MANAGE ONLY ONE PUT TO LIGHT GROUP

PTL_CLEAR command

This command clear a Put to Light's display.

Host →Modula Link: *<PREFIX>|<REQUEST_ID>|**PLT_CLEAR**|<ID_DISPLAY>*

Modula Link →Host: *<PREFIX>|<REQUEST_ID>|**PTL_CLEAR**|<RESULT>*

<PREFIX> in this message is not managed the bay and the machine so always insert '00'

<REQUEST_ID> is message ID

<ID_DISPLAY> display id

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = PTL not available or error*

Examples

Host → Modula Link	Modula Link → Host
00|1111|PTL_CLEAR|101	00|1111|PTL_ CLEAR|0
00|71|PTL _CLEAR |201	00|71|PTL_CLEAR|-1
99|1034|PTL _CLEAR|201	BAD_PREFIX
00|686|PTL _CLEAR|	BAD_PARAMETERS
00||PTL_CLEAR|201	MISSING_ID

MODULA LINK MANAGE ONLY ONE PUT TO LIGHT GROUP

PTL_CLEAR_ALL command

This command clears all Put to Light's displays.

*Host → Modula Link : <PREFIX>|<REQUEST_ID>|**PLT_CLEAR_ALL***

*Modula Link → Host : <PREFIX>|<REQUEST_ID>|**PTL_CLEAR_ALL**|<RESULT>*

<PREFIX> in this message is not managed the bay and the machine so always insert '00'

<REQUEST_ID> is message ID

<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = PTL not available or error*

Examples

Host → Modula Link	Modula Link → Host
00|1111|PTL_CLEAR_ALL	00|1111|PTL_CLEAR_ALL|0
00|71|PTL _CLEAR_ALL	00|71|PTL_CLEAR_ALL|-1
99|1034|PTL_CLEAR_ALL	BAD_PREFIX
00|686|	BAD_PARAMETERS
00||PTL_CLEAR_ALL	MISSING_ID

MODULA LINK MANAGE ONLY ONE PUT TO LIGHT GROUP

PTL_STATUS command

This command is used to know status of a Put to Light's display

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**PLT_STATUS***

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**PTL_STATUS**|<STATUS>|< DISPLAY1_STATUS>;< DISPLAY2_ STATUS >;< DISPLAY3_STATUS >…*

<PREFIX> in this message is not managed the bay and the machine so always insert '00'

<REQUEST_ID> is message ID

<STATUS> Put to Light Status:

- *OK (operative),*
- *KO (on error)*

<DISPLAY1_STATUS>,<DISPLAY2_STATUS>,<DISPLAY3_STATUS> status of PTL displays, formed by:
<Id Display>;<ActualStatus>;< ActualMessage >;<Color>;<Operation Confirm>:

- *<Id Display> Display id*
- *<ActualStatus> Display Status (OK, or KO)*
- *<ActualMessage> Actual Quantity/Message shown on display*
- *<Color> Actual color shown on display (R:Red, Y:Yellow, G:Green)*
- *<Operation Confirm> 'true' if the Operator have confirmed the operation*

- *<Quantity Confirm> Last quantity confirmed*

**The confirmed data are reset when new data are received or a clear display command is sent*

*** When WMS is turned on the list of displays is empty. When a command message for a display is received if the display exists on the PTL it is added to the list of displays*

Examples

Host → Modula Link	Modula Link → Host
00|1111|PTL_STATUS	00|1111|PTL_STATUS|KO
00|1112|PTL_STATUS	00|1112|PTL_STATUS|OK|101;OK;4;R;true;3|102;OK;10;R; false;0
00|1112|PTL_STATUS	00|1112|PTL_STATUS|OK|101;OK;4;R;true;3|102;OK;10;R; false;0
00|1113|PTL_STATUS	00|1113|PTL_STATUS|OK|101;OK;C;R;false;0|102;OK;10;R;true;10|103;OK;10;R;false;0
00|1114|PTL_STATUS	00|1114|PTL_STATUS|OK|101;OK;C;R;true;0|102;KO;C;R;false;0|103;OK;10;R; true;9
99|1034|PTL_STATUS	BAD_PREFIX
00|686|	BAD_PARAMETERS
00||PTL_STATUS	MISSING_ID

MODULA LINK MANAGE ONLY ONE PUT TO LIGHT GROUP

EXCHANGE command

This command is used for the Load Unit exchange in the bay specified.

Host → Modula Link: *<PREFIX>|<REQUEST_ID>|**EXCHANGE***

Modula Link → Host: *<PREFIX>|<REQUEST_ID>|**EXCHANGE**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = exchange LUs not possible*

Examples

Host → Modula Link	Modula Link → Host
31|8328|EXCHANGE	31|8328|EXCHANGE|0
62|9088|EXCHANGE	62|9088|EXCHANGE|-1
20|2|EXCHANGE	BAD_PREFIX
11|666|2|EXCHANGE	BAD_PARAMETERS
71||EXCHANGE	MISSING_ID

LEDBAR_LIGHT command

This command allows to lights on specified led on the simple led bar.

*Host → Modula Link : <PREFIX>|<REQUEST_ID>|**LEDBAR_LIGHT**|<LED NUMBER X>|<LED NUMBER Y>|<LED NUMBER ORDER>*

*Modula Link → Host : <PREFIX>|<REQUEST_ID>|**LEDBAR_LIGHT**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<LED NUMBER X> is the led that has to be lighted on for horizontal bars (Note: number of available leds depends on machine model; for istance a bar of an ML machine has 400 leds)*

<LED NUMBER Y> is the led that has to be lighted on for vertical bars (depth indication)*
<LED NUMBER ORDER> is the led that has to be lighted on for orders bars*
<RESULT> is result of the request. Possible values are:

- *“0” = ok*
- *“-1” = generic error*

**passing 0 as led value means turn off all leds of the led bar.*

Examples

Host → Modula Link	Modula Link → Host
51|1111|LEDBAR_LIGHT|128|0|0	51|1111|LEDBAR_LIGHT|0
31|71|LEDBAR_LIGHT|55|0|0	31|71|LEDBAR_LIGHT|-1

LEDBAR_LIGHT_OFF command

This command allows to light off all led bars.
*Host → Modula Link : <PREFIX>|<REQUEST_ID>|**LEDBAR_LIGHT_OFF***
*Modula Link → Host : <PREFIX>|<REQUEST_ID>|**LEDBAR_LIGHT_OFF**|<RESULT>*

<PREFIX> is bay/machine identifier
<REQUEST_ID> is message ID
<RESULT> is result of the request. Possible values are:

- *“0” = ok*
- *“-1” = generic error*

Examples

Host → Modula Link	Modula Link → Host
51|1111|LEDBAR_LIGHT_OFF	51|1111|LEDBAR_LIGHT_OFF|0
31|71|LEDBAR_LIGHT_OFF	31|71|LEDBAR_LIGHT_OFF|-1

CALL_BIN command

Request is to extend Modula Link CALL command functionality in order to giving host system possibility to activate carousel rotation for moving requested bin to exit bay as well as tray call feature already implemented for VLM machines:

Host →Modula Link: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**CALL_BIN**|<BIN>*
Modula Link →Host: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID| **CALL_BIN** |<RESULT>*

<POD_ID> is the POD ID
<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.
<REQUEST_ID> is message ID
<BIN> is the bin number
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1"= bin number not valid*
- *"-2"= generic error*

STATUS_BIN command

Request is to extend Modula Link STATUS command functionality in order to giving host system possibility to get carousel/exit bay current status information:

Host →Modula Link: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**STATUS_BIN***
Modula Link →Host:
*<POD_ID>|<DEVICE_ID>|<REQUEST_ID|**STATUS_BIN**|<STATUS>|<BINPICK>|<BINEXE>|<DOORSTATUS>*

<POD_ID> is the POD ID
<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.
<REQUEST_ID> is message ID
<STATUS> is exit group status:

- *"0" = exit group ready (on-line, automatic mode active)*
- *"1" = exit group in manual*
- *"2" = exit group off-line*

<BINPICK> is number of the bin on the picking/exit position. If no bin is on picking position, its value is "0"
<BINEXE> is number of the bin in execution
<DOORSTATUS> is exit door status used for HC PODs only (optional):

- *"0" = door closed*
- *"1" = door opened*
- *"2" = door is closing*
- *"3" = door is opening*
- *"-1" = door in error*

END_BIN command

Request is to give host system possibility to close bin mission and automatically close exit door

Host →Modula Link: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**END_BIN***
Modula Link →Host: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID|**END_BIN**|<RESULT>*

<POD_ID> is the POD ID
<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.
<REQUEST_ID> is message ID
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = no bin at exit position*
- *"-2" = generic error*

IMPORTANT

Door closing is automatic managed by HC machine when END_BIN command is received

DOOR_OPEN command

Request is to give host system possibility to open doors of a carousel exit bay

Host →Modula Link: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**DOOR_OPEN***
Modula Link →Host: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID|**DOOR_OPEN**|<RESULT>*

<POD_ID> is the POD ID
<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.
<REQUEST_ID> is message ID
<RESULT> is result of the request. Possible values are:

- *"0" = ok*
- *"-1" = generic error*

IMPORTANT

When requested bin reaches exit bay position, exit door won't ever open automatically (it's in charge of host system to open it using DOOR_OPEN command)

DOOR_CLOSE command

Request is to give host system possibility to close doors of a carousel exit bay

Host →Modula Link: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**DOOR_CLOSE***
Modula Link →Host: *<POD_ID>|<DEVICE_ID>|<REQUEST_ID|**DOOR_CLOSE**|<RESULT>*
<POD_ID> is the POD ID

<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.

<REQUEST_ID> is message ID

<RESULT> is result of the request. Possible values are:

- *“0” = ok*
- *“-1” = generic error*

IMPORTANT

Door closing is also automatic managed by HC machine while bins chain is rotating

RGB_CLEAR command

Request is to give host system possibility to open doors of a carousel exit bay

***Host → Modula Link:** <POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**RGB_CLEAR**|<SIDE>*

***Modula Link → Host:** <POD_ID>|<DEVICE_ID>|<REQUEST_ID|**RGB_CLEAR**|<RESULT>*

<POD_ID> is the POD ID

<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.

<REQUEST_ID> is message ID

<SIDE> refers to which bar should be cleared. Possible values are:

- *“0” = both left/right bars*
- *“1” = first bar*
- *“2” = second bar*

<RESULT> is result of the request. Possible values are:

- *“0” = ok*
- *“-1” = RGB led bar not available or generic error*

RGB_SHOW command

Request is to give host system possibility to open doors of a carousel exit bay

Host → Modula Link:

*<POD_ID>|<DEVICE_ID>|<REQUEST_ID>|**RGB_SHOW**|<COLOR>|<BLINK>|<FIRST_LED>|<LAST_LED>|<SIDE>*

***Modula Link → Host:** <POD_ID>|<DEVICE_ID>|<REQUEST_ID| **RGB_SHOW** |<RESULT>*

<POD_ID> is the POD ID

<DEVICE_ID> is the DEVICE_ID for specific device that belongs to specified POD.

<REQUEST_ID> is message ID

<COLOR> is colour ID

<BLINK> is blinking mode

- *“0” = no blink*

- *“1” = blink*
- *<COLOR> is color to switch on*
- *“0” = white*
- *“1” = yellow*
- *“2” = magenta*
- *“3” = red*
- *“4” = cyan*
- *“5” = green*
- *“6” = blue*
- *“7” = no colour*

<FIRST_LED> is first led row to switch on
<LAST_LED> is last led row to switch on
<SIDE> refers to which bar should be driven. Possible values are:

- *“0” = both left/right bars*
- *“1” = left bar*
- *“2” = right bar*

<RESULT> is result of the request. Possible values are:

- *“0” = ok*
- *“-1” = RGB led bar not available or generic error*

EXTRACTION command

This command is used to start a tray extraction.
This command is strictly connected to trolley management.
Conditions:
The tray must be available on the exit group/machine provided, so generally this command follows the CALL command.
The trolley must be inserted in the exit bay (check trolley presence signal)

Host → Modula Link:
*<PREFIX>|<REQUEST_ID>|**EXTRACTION**|<TRAY>|<POSITION>*
Modula Link → Host:
*<PREFIX>|<REQUEST_ID>|**EXTRACTION**|<RESULT>*

<PREFIX> is exit group/machine identifier
<REQUEST_ID> is message ID
<TRAY> is tray number
<POSITION> is position (“1” = lower position; “2” = upper position) where tray must exit
<RESULT> is result of the request. Possible values are:

- *“0”: ok*

- *“-1”: tray number not valid*
- *“-2“: position not valid*
- *“-3”: position is occupied by different tray*
- *“-4”: position is free (the requested tray is not on exit position)*
- *“-5”: position is disabled*
- *“-6”: user not logged*

ENDEXTRACTION command

This command is used to complete a tray extraction procedure.

This command is strictly connected to trolley management.

Conditions:

The exit position must be free (not occupied by the tray)

Host → Modula Link:

*<PREFIX>|<REQUEST_ID>|**ENDEXTRACTION**|<TRAY>|<POSITION>*

Modula Link → Host:

*<PREFIX>|<REQUEST_ID>|**ENDEXTRACTION**|<RESULT>*

<PREFIX> is exit group/machine identifier

<REQUEST_ID> is message ID

<TRAY> is tray number

<POSITION> is position (“1” = lower position; “2” = upper position) where tray must exit

<RESULT> is result of the request. Possible values are:

- *“0”: ok*
- *“-1”: tray number not valid*
- *“-2“: position not valid*
- *“-3”: position is occupied*

INSERTION command

This command is used to start a tray insertion.

This command is strictly connected to trolley management.

Conditions:

- *The exit position must be free (not occupied by a tray)*
- *The trolley must be inserted in the exit bay (check trolley presence signal)*

Host → Modula Link:

*<PREFIX>|<REQUEST_ID>|**INSERTION**|<TRAY>|<POSITION>*

Modula Link → Host:

*<PREFIX>|<REQUEST_ID>|**INSERTION**|<RESULT>*

<PREFIX> is exit group/machine identifier

<REQUEST_ID> is message ID

<TRAY> is tray number

<POSITION> is position ("1" = lower position; "2" = upper position) where tray must exit

<RESULT> is result of the request. Possible values are:

- *"0": ok*
- *"-1": tray number not valid*
- *"-2": position not valid*
- *"-3": position is occupied*
- *"-4": position is disabled*

ENDINSERTION command

This command is used to complete a tray insertion procedure.

This command is strictly connected to trolley management.

Conditions:

The exit position must be occupied by the inserted tray

Host → Modula Link:

*<PREFIX>|<REQUEST_ID>|**ENDINSERTION**|<TRAY>|<POSITION>|<SIDE_HEIGHT>*

Modula Link → Host:

*<PREFIX>|<REQUEST_ID>|**ENDINSERTION**|<RESULT>*

<PREFIX> is exit group/machine identifier

<REQUEST_ID> is message ID

<TRAY> is tray number

<POSITION> is position ("1" = lower position; "2" = upper position) where tray must exit

<SIDE_HEIGHT> is the height in millimeters of the raised part of the drawer

<RESULT> is result of the request. Possible values are:

- *"0": ok*
- *"-1": tray number not valid*
- *"-2": position not valid*
- *"-3": position is occupied by different tray*
- *"-4": position is free (the requested tray is not on exit position)*
- *"-5": position is disabled*

Commands and WMS versions

Table summarize for each command the WMS version in which they were introduced.

Command	WMS version
STATUS	3.9.8.0
CALL	3.9.8.0
RETURN	3.9.8.0
LASER_ON	3.9.8.0
LASER_OFF	3.9.8.0
LASER_HOME	3.9.8.0
LASER_GO	3.9.8.0
LASER_STATUS	3.9.27.0
DISPLAY_CLEAR	3.9.8.0
DISPLAY_SHOW	3.9.8.0
PTL_SHOW_QTA	3.9.27.0
PTL_SHOW_MESSAGE	3.9.27.0
PTL_CLEAR	3.9.27.0
PTL_CLEAR_ALL	3.9.27.0
PTL_STATUS	3.9.27.0
EXCHANGE	3.10.4
LEDBAR_LIGHT	3.10.4
LEDBAR_LIGHT_OFF	3.10.4
CALL_BIN	3.10.10
STATUS_BIN	3.10.10
END_BIN	3.10.10
DOOR_OPEN	3.10.10

DOOR_CLOSE	3.10.10
RGB_CLEAR	3.10.10
RGB_SHOW	3.10.10
EXTRACTION	3.10.12
ENDEXTRACTION	3.10.12
INSERTION	3.10.12
ENDINSERTION	3.10.12