



GoPal P35

ATR2101

Autonomous Platform Robot

Top Module Interface Control Document

Doc No.:	012372
Revision:	B-DRAFT4
Date:	August 9. 2024
Author(s):	Anders Netterstrøm Magnus C Olsen Kenn Blomgren Lars Theil Hansen

Release date:	Month, year
---------------	-------------

By (initials):	NN
----------------	----

REVISION HISTORY				
Rev.	Date	App.	Sections	Description/Remarks
Draft 4C	2024-04-15	MCO /LTH/KB	All	Document template repaired, title page aligned with Robotize standards, table and figure numbers added, left margins aligned, general corrections, electrical interface added, mechanical interface added
B-DRAFT2	2024-06-10	MCO	3.1 + 4	Updated incorrect figures in section 3.1. Change of connectors 1, 2, 4 and 7 in section 4 and mating cable parts.
B-DRAFT3	2024-06-11	ANE	5.2, 5.3	<ul style="list-style-type: none"> TOP_e_stop, TOP_s_stop, ROB_s_mon signals are active low. Description refers directly to the signals. A bit more about fast rotation.
B-DRAFT3	2024-06-11	ANE	7.2.1	“interface” section: Port number for JSON_RPC_TCP.
B-DRAFT3	2024-06-11	ANE	7.3	“safety” section: <ul style="list-style-type: none"> Renamed to lidar section TOP_pattern related information moved to 7-8 Only 1 safety configuration is supported. Added “sensor_view_obstacles”, “disable_fast_rotation”
B-DRAFT3	2024-06-11	ANE	7.4	“general” section: <ul style="list-style-type: none"> Added “battery_power”, “mass”, “height” Modified “reverse_docking”
B-DRAFT3	2024-06-11	ANE	7.5	“info_items” section: Added information elements: <ul style="list-style-type: none"> TOP_STOP_REQ, TOP_PATTERN_REQ, ROB_POWER_OFF. Define where information elements can be used.

B-DRAFT3	2024-06-11	ANE	7.7	“actions” section: <ul style="list-style-type: none"> Update “sim” object with simulator info. Define if the robot operator can activate the action. Added “top_pattern” Remove “rob_low_speed”, “rob_muted”
B-DRAFT3	2024-06-11	ANE	7.8	“station” section removed.
B-DRAFT3	2024-06-11	ANE	7.9	Added new section in top module config file with meta data.
B-DRAFT3	2024-06-17	LTH	4.1.5 4.1.4 4.1.6	top_e_stop, top_s_stop, rob_s_mon, defined as negative logic. Specifications for safe logic signals added.
B-DRAFT4	2024-08-05	ANE	5.3	Emphasizing that area definitions is the only part of the LIDAR configuration that may be altered. Add reference to Lidar area settings.
B-DRAFT4	2024-08-05	ANE	5.2	ROB_s_monitor: What activates this signal.
B-DRAFT4	2024-08-05	ANE	7.4	“general” section: <ul style="list-style-type: none"> Added “passive”
B-DRAFT4	2024-08-05	ANE	7.4	Fix problem with missing “_” in “mute_safety” values.
B-DRAFT4	2024-08-05	ANE	7.4+7.7	Information required to simulator top_pattern signal.
B-DRAFT4	2024-08-05	ANE	6.2	Rename t_ready_min to t_not_ready_min
B-DRAFT4	2024-08-09	LTH	4.1.7	Information about alternative source for Top Module power added
B-DRAFT4	2024-08-09	LTH	8.	Appendix A added, to match para. 4.1.7

All information in this document is the property of Robotize ApS and may not be reproduced either in whole or partly without prior written acceptance by Robotize ApS.

Copyright © 2016-2024 by Robotize ApS

Table of Contents

1	General.....	1
1.1	Scope and purpose	1
1.2	Related documents.....	1
1.3	Referenced documents.....	1
1.4	Abbreviations.....	1
1.5	Terminology	1
2	Overview	2
2.1	Block diagram, Robot/Top Module logic interaction.....	2
2.2	Use cases.....	3
3	Mechanical Interface	5
3.1	Top Module mounting	5
3.2	Stability	9
4	Electrical Interface	10
4.1	Connector types and pin-out	11
4.1.1	Ethernet	11
4.1.2	CAN	11
4.1.3	WiFi	12
4.1.4	Safety Reset	12
4.1.5	Safety A	13
4.1.6	Safety B	13
4.1.7	Power	15
4.1.8	Fuse	16
4.2	List of mating cable part connectors w. pictures.....	17
5	Safety Interface.....	19
5.1	Overview of safety system.....	19
5.2	Safety signals between Robot and Top Module	20
5.3	Lidar zone sets	22
6	Control Interface.....	24
6.1	Action concept	24
6.2	Control over CAN	26
6.2.1	CAN message	30
6.3	Control over Ethernet.....	31
6.3.1	Method: get_top_module_info	32

6.3.2	Method: start_action	32
6.3.3	Method: abort_action	34
6.3.4	Method: get_log	35
6.3.5	Method: sync_state	36
7	Top Module Configuration	38
7.1	"top_module" section	38
7.2	"interface" section	39
7.2.1	Ethernet	39
7.2.2	CAN	39
7.3	"lidar" section	41
7.4	"general" section	42
7.5	"info_items" section	43
7.6	"action_args" section	45
7.7	"actions" section	46
7.8	Removed	48
7.9	"meta" section	49
8	Appendix A	50

List of Tables

Table 1.1	Terminology	1
Table 4.1	Top Module Interface connector reference numbers and functions	10
Table 4.2	Top Module cable part connectors	18
Table 5.1	Robot/Top Module safe signals	21
Table 5.2	Selection of lidar patterns and areas	23
Table 5.3	Area selection vs Robot speed	23
Table 6.1	CAN messages	26
Table 6.2	CAN signal timing	27
Table 6.3	CAN messages, robot	30
Table 6.4	CAN messages, top module.....	30
Table 6.5	CAN messages, coding.....	31

List of Figures

Figure 2.1	Robot/Top Module logic interaction block diagram.....	2
Figure 2.2	Robot/Top Module activity sequence.....	3
Figure 2.3	Robot/Top Module location specific activity sequence.....	4
Figure 2.4	Robot/Top Module periodical activities	4
Figure 3.1	P35 Robot M-ICD dwg. 1	6
Figure 3.2	P35 Robot M-ICD dwg. 2	7
Figure 3.3	P35 Robot M-ICD dwg. 3	8
Figure 4.1	Top Module Interface connectors and reference numbers.....	10
Figure 5.1	Robot/Top Module safety system overview	19
Figure 6.1	Action concept	24
Figure 6.2	Signalling states for robot and top module.	28
Figure 8.1	Location of Power Control interface Board (green arrow)	50
Figure 8.2	Location of the “TOP” terminal on the Power Control interface Board (green arrow)	51
Figure 8.3	Location of Brake Load Dump Board (red arrow)	52
Figure 8.4	Location of the “LIFT” terminal on the Brake Load Dump Board (red arrow)	53

1 General

1.1 Scope and purpose

This document describes the interface between the GoPal P35 robot and a top module for this robot.

Logic control interface, as well as electrical/physical interface is covered.

1.2 Related documents

- (1) 012683-720, ATR2101, P35 Robot to Top Module Safety Interface Block diagram

1.3 Referenced documents

- [1] <https://www.jsonrpc.org/specification>
[2] https://en.wikipedia.org/wiki/CAN_bus
[3] 012682-710, ATR2101, P35 Robot, Specification Sheet

1.4 Abbreviations

ICD	Interface Control Document
RPC	Remote Procedure Call
JSON	JavaScript Object Notation

1.5 Terminology

Table 1.1 below lists the terminology used in this document.

Charging station	Position where the robot charges the batteries.
GoControl	GoControl (dispatcher) – a central application running on a server found in the company which guides and checks all GoPal devices (robots, buttons, doors, charging stations etc.). The dispatcher decides which jobs are performed and how the various jobs and robots are prioritized
parking position	Parking positions are used for idle robots or robots waiting for some condition (like station full or empty, zone capacity). Parking positions can also be used as intermediate goals when driving from the pick-up to the delivery station where special handling can be performed.
site configuration	The site configuration includes all elements relevant for the robot to work on a site. That includes robots, top modules, stations, charging stations, push buttons, sensors, doors, parking positions, configurable zones etc. The site configuration is created in the site editor and pushed to all devices.

Table 1.1 Terminology

2 Overview

2.1 Block diagram, Robot/Top Module logic interaction

The logic control block diagram is shown in Figure 2.1 below.

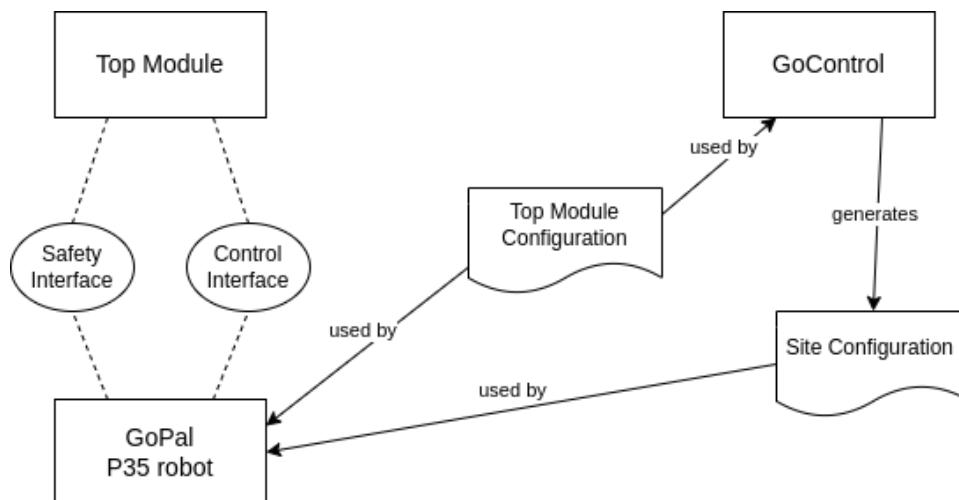


Figure 2.1 Robot/Top Module logic interaction block diagram

The top module and the robot interact via two interfaces:

- The safety interface that supports the signaling needed to implement a safe system.
- The control interface that supports the signaling needed to make the top module work with the robot in the typical application.

The configuration of a particular site is given by the site configuration. The site configuration includes detailed information about all system elements (robots, stations, chargers, doors, ...).

GoControl is the fleet manager that coordinates all activities in the system. GoControl also enables a user to change and distribute the site configuration.

Robots must have knowledge about the specific top module to be able to use it correctly. However, neither GoControl nor robot can have any built-in knowledge about any top module as this would require a new software every time a new top module is to be integrated with the GoPal system. Therefore, the required top module information must be provided by the top module configuration. The top module configuration is required by both robot and GoControl and includes information about the communication interface, when and how the top module is activated by the robot etc.

2.2 Use cases

A robot with top module may support moving goods from a pickup station to a delivery station. Top modules for this use case could be based on a lift or conveyor function but it also includes top modules for moving carts. The activity sequence is shown below in Figure 2.2.

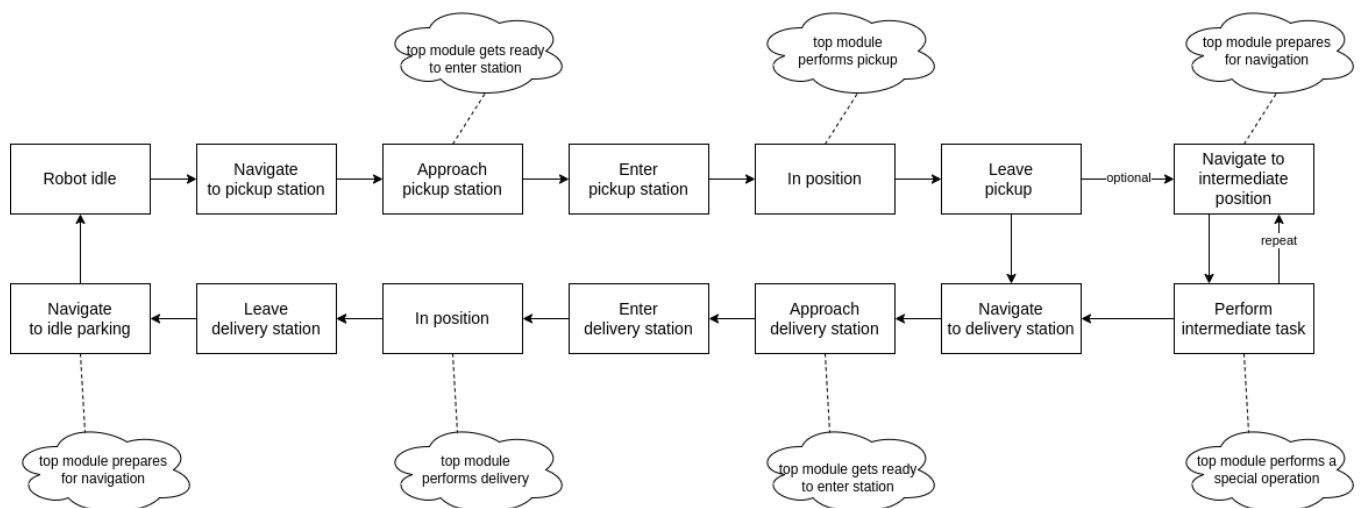


Figure 2.2 Robot/Top Module activity sequence

The top module may have different tasks to perform:

- Before the robot enters a station to pick up, the top module must enter a state that ensures that the robot with top module will not collide with the station or goods on the station as the robot enters the station.
- Before the robot enters a station to deliver, the top module must enter a state that ensures that the robot with top module and goods will not collide with the station as the robot enters the station.
- When robot is in the pickup/delivery position in the station the top module must perform the actual pick-up or delivery.
- When the robot starts to navigate the top module must enter a state suitable for navigation.
- The GoPal system has a concept where an intermediate task can be executed at specific positions between pick-up and delivery. An intermediate task may include activities on the top module.

A robot with top module may also be a movable tool/device that can be activated at specific locations. The location specific activity sequence is shown in Figure 2.3 below.

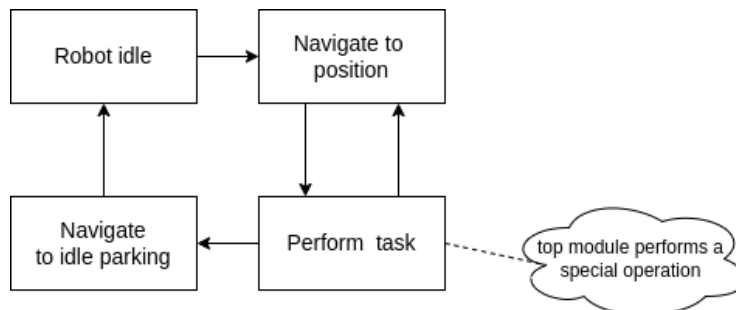


Figure 2.3 Robot/Top Module location specific activity sequence

A robot must charge the batteries periodically. It may also be useful for the top module to perform activities that have to be done from time to time. See Figure 2.4 below.

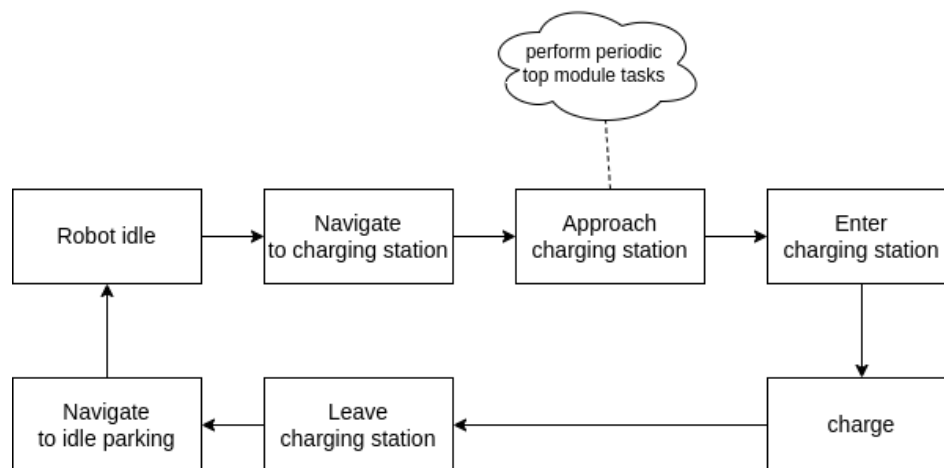


Figure 2.4 Robot/Top Module periodical activities

3 Mechanical Interface

3.1 Top Module mounting

See Figure 3.1, Figure 3.2 and Figure 3.3 on following pages.

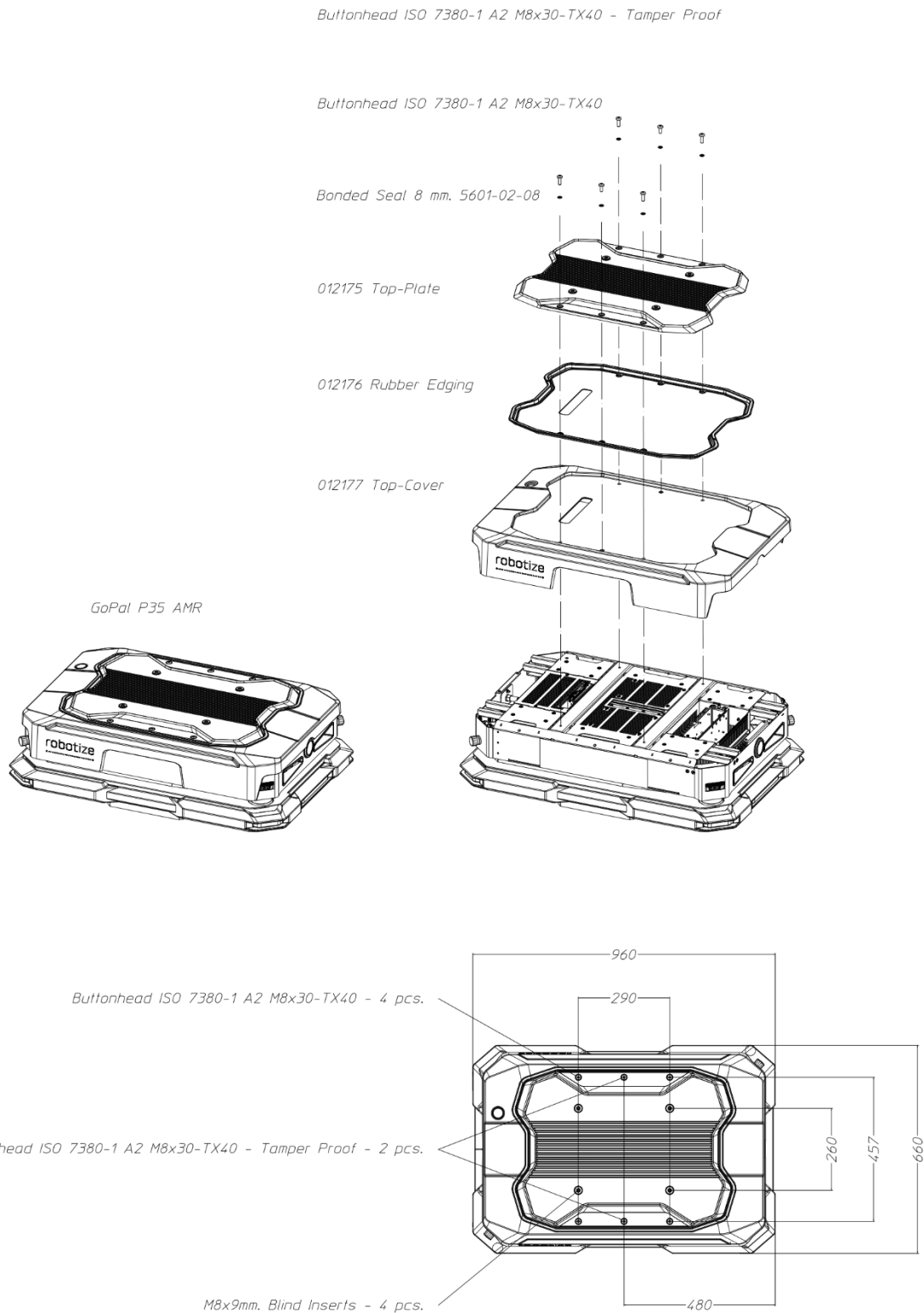


Figure 3.1 P35 Robot M-ICD dwg. 1

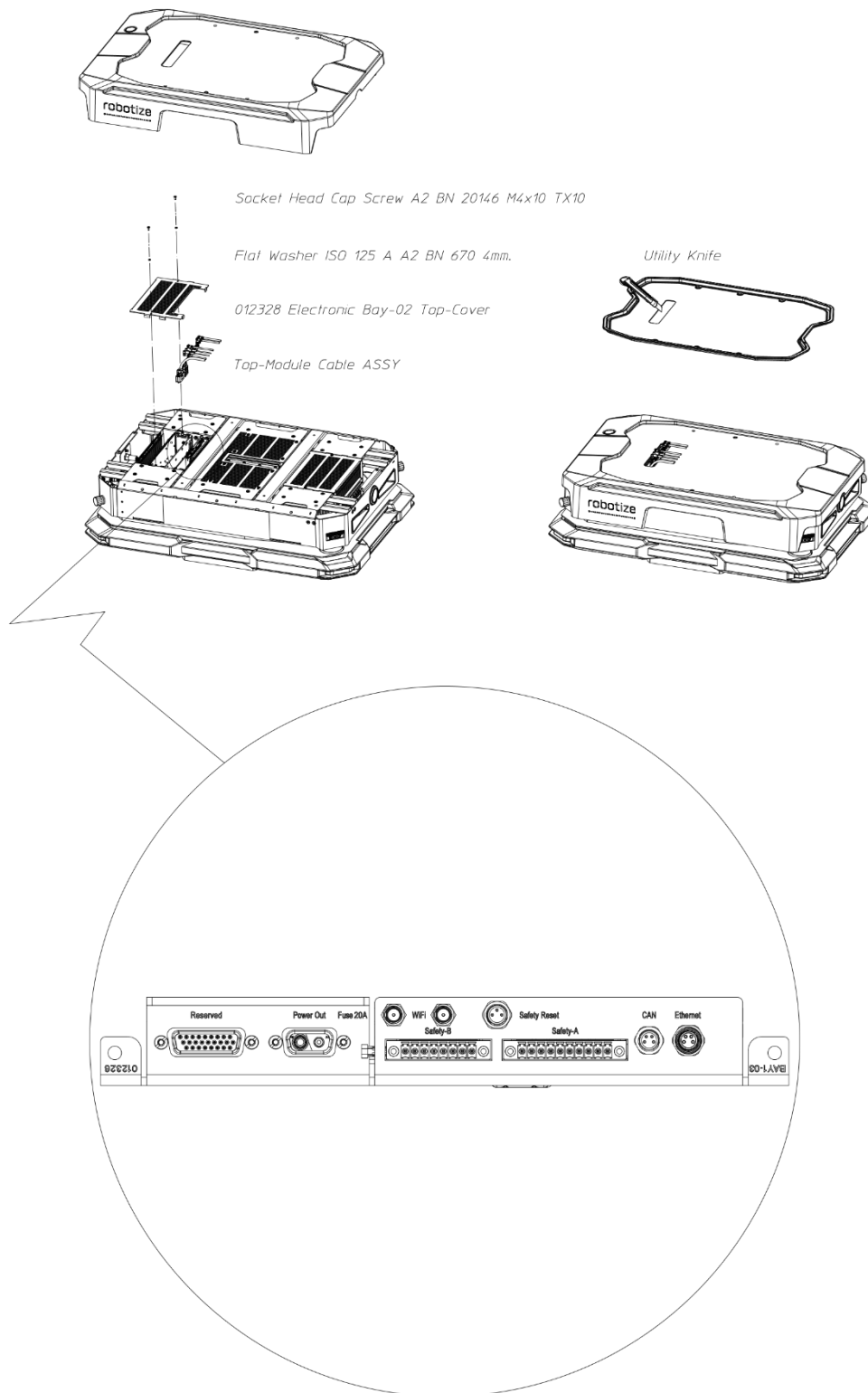
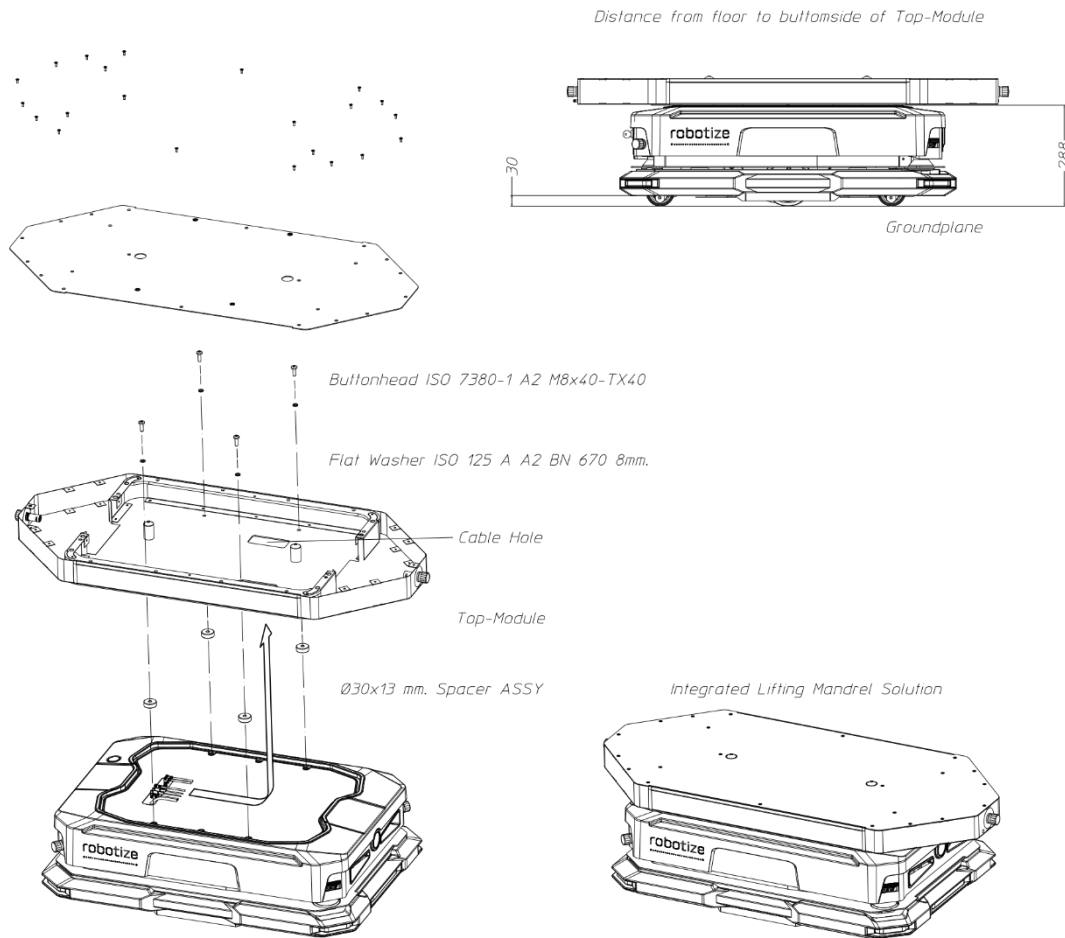


Figure 3.2 P35 Robot M-ICD dwg. 2

NOTE:EMERGENCY STOP BUTTONS:

The AMR is fitted with two Emergency Stop Buttons:

Mfr.: ABB

Mfr. No.: CE3T-10R-02

It is preferable that the Top Module uses the same type.

PAINT SPECIFICATION:

The AMR Top-Cover is painted with a Powder Coating System:

AkzoNobel Interpon Powder Coating System:

- Primer Interpon 100, AN001T
- Top Coat Interpon 810, 89-72E/T
- Colour: RAL 9011 Graphite Black / Fine Structure / GLOSS 02
- Thickness: 60-120 µm

It is preferable that the Top Module uses the same type.

Figure 3.3 P35 Robot M-ICD dwg. 3

3.2 Stability

Stability diagrams TBA.

4 Electrical Interface

The top module interfaces through the connectors illustrated below in Figure 4.1. They are accessible through the robot's cover aft of the battery compartment. The connectors are described further in their respective sections. Connector functions are listed in Table 4.1.

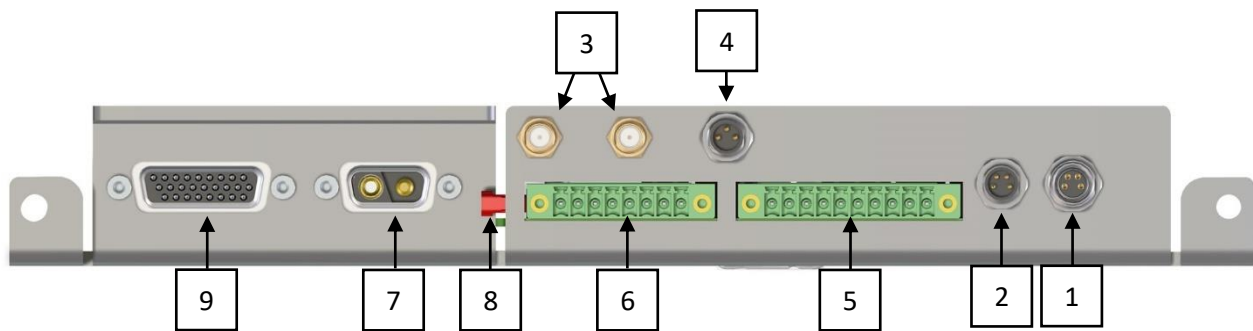


Figure 4.1 Top Module Interface connectors and reference numbers

Conn. No.	Function
1	Ethernet
2	CAN
3	WiFi
4	Safety Reset
5	Safety/EMG A
6	Safety/EMG B
7	Power
8	Fuse
9	Reserved

Table 4.1 Top Module Interface connector reference numbers and functions

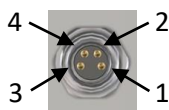
In para. 4.1 each connector is described in more detail. **All connector pinouts below are shown as viewed in Figure 4.1.**

4.1 Connector types and pin-out

4.1.1 Ethernet

Ethernet connection is accessible through connector (1). Speed max. 100 Mb.
See para. 6.3 for functional details.

	Ethernet Connector	Recommended mating/cable part
MFN:	Tensility	MURR Elektronik GmbH, DK
PN:	54-00274	7000-89781-7910100

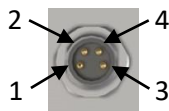


Ethernet Connector Pinout		
Function	Pin #	Description
TX+	1	Positive transmit signal
RX+	2	Positive receive signal
RX-	3	Negative receive signal
TX-	4	Negative transmit signal

4.1.2 CAN

CAN connection is accessible through a dedicated CAN channel to the robot through connector (2). Speed max. 1 Mb. See 6.2 for functional details.

	CAN Connector	Recommended mating/cable part
MFN:	Tensility	TE Connectivity
PN:	54-00272	1-2273011-1



CAN Connector Pinout		
Function	Pin #	Description
CAN V+	1	(Inactive)
CAN High	2	CAN high signal
CAN V-	3	GND
CAN Low	4	CAN low signal

4.1.3 WiFi

If the robot WiFi coax cables are reconfigured accordingly, it is possible to connect WiFi antennas via the two SMA connectors (3).

	WiFi Connector	Recommended mating/cable part
MFN:	TE Connectivity	Amphenol RF
PN:	ADP-SMAF-SMAF-B-G	135104-01-M0.25

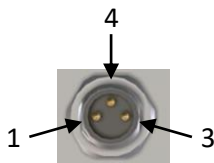


It is not important which WiFi-antenna is connected to which connector.

4.1.4 Safety Reset

If desired, it is possible to equip the Top Module with an extra safety system reset switch in parallel with the already existing switch on the robot. This is done through connector (4).

	Safety Reset Connector	Recommended mating/cable part
MFN:	Tensility	TE Connectivity
PN:	54-00290	1-2273009-1



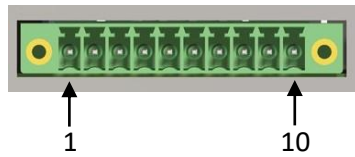
Safety Reset Connector Pinout		
Function	Pin #	Description
TOP_e_stop_reset	1	24 V logic in. Reset when pulled to 24 V for min 0.5 s
ROB_e_stop_indc	3	6 V signal "reset required", for indication, e.g. driving LED with series resistor. 20 mA available.
GND	4	Ground reference

Safety reset signal specifications	
Function	Specification
24 V logic in	EN61131-2 input type 1. Input low voltage < 5 V. Input high voltage > 15 V

4.1.5 Safety A

Safety A signals and EMG A is accessed through connector (5). Signals are 24 V logic signal for the robot safety-PLC.

	Safety/EMG A Connector	Recommended mating/cable part
MFN:	Phoenix Contact	Phoenix Contact
PN:	1817699	1847204



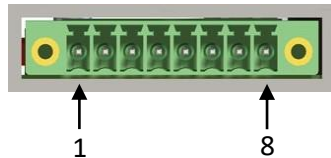
Safety/EMG A Connector Pinout			
Signal	Pin #	Logic polarity	Functional description
ROB_e_stop_man A	1	NA (coded signal)	24 V logic out. See also para. 5.1
TOP_e_stop_man A	2	NA (coded signal)	24 V logic in. See also para. 5.1
TOP_pattern_1 A	3	Positive (active high)	24 V logic in. See also para. 5.2
TOP_pattern_0 A	4	Positive (active high)	24 V logic in. See also para. 5.2
NC	5	NA (not conn.)	Not connected
TOP_e_stop A	6	Negative (active low)	24 V logic in. See also para. 5.2
TOP_s_stop A	7	Negative (active low)	24 V logic in. See also para. 5.2
ROB_s_mon A	8	Negative (active low)	24 V logic out. See also para. 5.2
24V_AUX	9	NA (supply voltage)	24 V supply voltage
GND	10	NA (ground)	Ground for the above signals

Safety/EMG A Signal specifications	
Function	Specification
24 V logic in	EN61131-2 input type 1. Input low voltage < 5 V. Input high voltage > 15 V
24 V logic out	OSSD-type, PNP, current limited, monitored (short voltage drops). Max. load 400 mA. Max. capacitive load 820 nF. Max. inductive load 30 mH. Output high voltage > 21 V. Output low < 3 V.
24 V supply	24 V +1/-2 V AUX power supply for Top Module safety circuits – if relevant. Current limited to 1 A +20 % / -10 %. Max. capacitive load for guaranteed turn ON: 1000 uF.

4.1.6 Safety B

Safety B signals and EMG B is accessed through connector (6). Signals are 24 V logic signal for the robot safety-PLC.

	Safety/EMG B Connector	Recommended mating/cable part
MFN:	Phoenix Contact	Phoenix Contact
PN:	1817673	1847181



Safety/EMG B Connector Pinout			
Signal	Pin #	Logic polarity	Description
ROB_e_stop_man B	1	NA (coded signal)	24 V logic out. See also para. 5.1
TOP_e_stop_man B	2	NA (coded signal)	24 V logic in. See also para. 5.1
TOP_pattern_1 B	3	Positive (active high)	24 V logic in. See also para. 5.2
TOP_pattern_0 B	4	Positive (active high)	24 V logic in. See also para. 5.2
NC	5	NA (not conn.)	Not connected
TOP_e_stop B	6	Negative (active low)	24 V logic in. See also para. 5.2
TOP_s_stop B	7	Negative (active low)	24 V logic in. See also para. 5.2
ROB_s_mon B	8	Negative (active low)	24 V logic out. See also para. 5.2

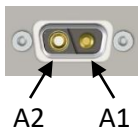
Safety/EMG B Signal specifications	
Function	Specification
24 V logic in	EN61131-2 input type 1. Input low voltage < 5 V. Input high voltage > 15 V
24 V logic out	OSSD-type, PNP, current limited, monitored (short voltage drops). Max. load 400 mA. Max. capacitive load 820 nF. Max. inductive load 30 mH. Output high voltage > 21 V. Output low < 3 V.

4.1.7 Power

Output drive power from battery (raw battery voltage in the 22 – 29 range) for Top Module loads up to 20 A. The output is secured from shorts and over-currents through the fuse indicated at (8).

The Top Module power can be sourced from two different points inside the robot. See Note below for details.

	Power Connector	Recommended mating/cable part
MFN: PN:	Amphenol 302W2CSXX56N40X	Amphenol CONEC 302W2CPXX41A10X + Amphenol FCI 8655MH0901BLF



Power Connector Pinout		
Function	Pin #	Description
+Vbat	A1	Positive supply for Top Module
-Vbat	A2	Negative supply for Top Module

Note: Top Module alternative power source

The default source for the Top Module power is the robot battery (raw battery voltage) through a solid-state switch. In this configuration power for the Top Module is available when the robot is on, configured for a Top Module, and ready to interface with a Top Module (the robot CPU is running, and no serious robot faults are present).

In cases where the Top Module is simple and does not have its own safety relay(s), but perhaps is only equipped with simple emergency stop buttons, it might be attractive to let the robot's safety relays control the power for the Top Module. In this case it is possible to physically change the internal power source to a alternative point after the robot safety relays. The available voltage is still raw battery voltage, but the Top Module power will now be off whenever the robot is safety, or emergency, stopped. This will be the case regardless of the source of the stop (robot or Top Module).

Appendix A describes how to change power source for the Top Module power.

4.1.8 Fuse

The fuse holder is compatible with ATO FKH 32V Blade Fuse series from Littelfuse. Compatible ATO blade car fuses with a 20 A rating can be used.



4.2 List of mating cable part connectors w. pictures

No	Pic	PNO	MFG
1		7000-89781-7910100	MURR Elektronik GmbH, DK
2		1-2273011-1	TE Connectivity
3		135104-01-M0.25	Amphenol RF
4		1-2273009-1	TE Connectivity
5		1847204 - MC 1,5/10-STF-3,5	Phoenix Contact GmbH & Co. KG
6		1847181 - MC 1,5/ 8-STF-3,5	Phoenix Contact GmbH & Co. KG
7		302W2CPXX41 A10X	Amphenol CONEC
7		8655MH0901 BLF	Amphenol FCI


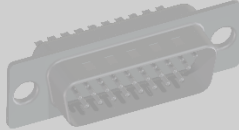

8		20 A standard ATO blade fuse	Any
9		NB! Reserved 10090769-P264ALF	 Amphenol FCI
9		NB! Reserved 8655MHRA1501LF	 Amphenol FCI

Table 4.2 Top Module cable part connectors

5 Safety Interface

5.1 Overview of safety system

The robot's safety system monitors and halts the robot in case of a dangerous situation. Figure 5.1 gives an overview of the Robot/Top Module safety system.

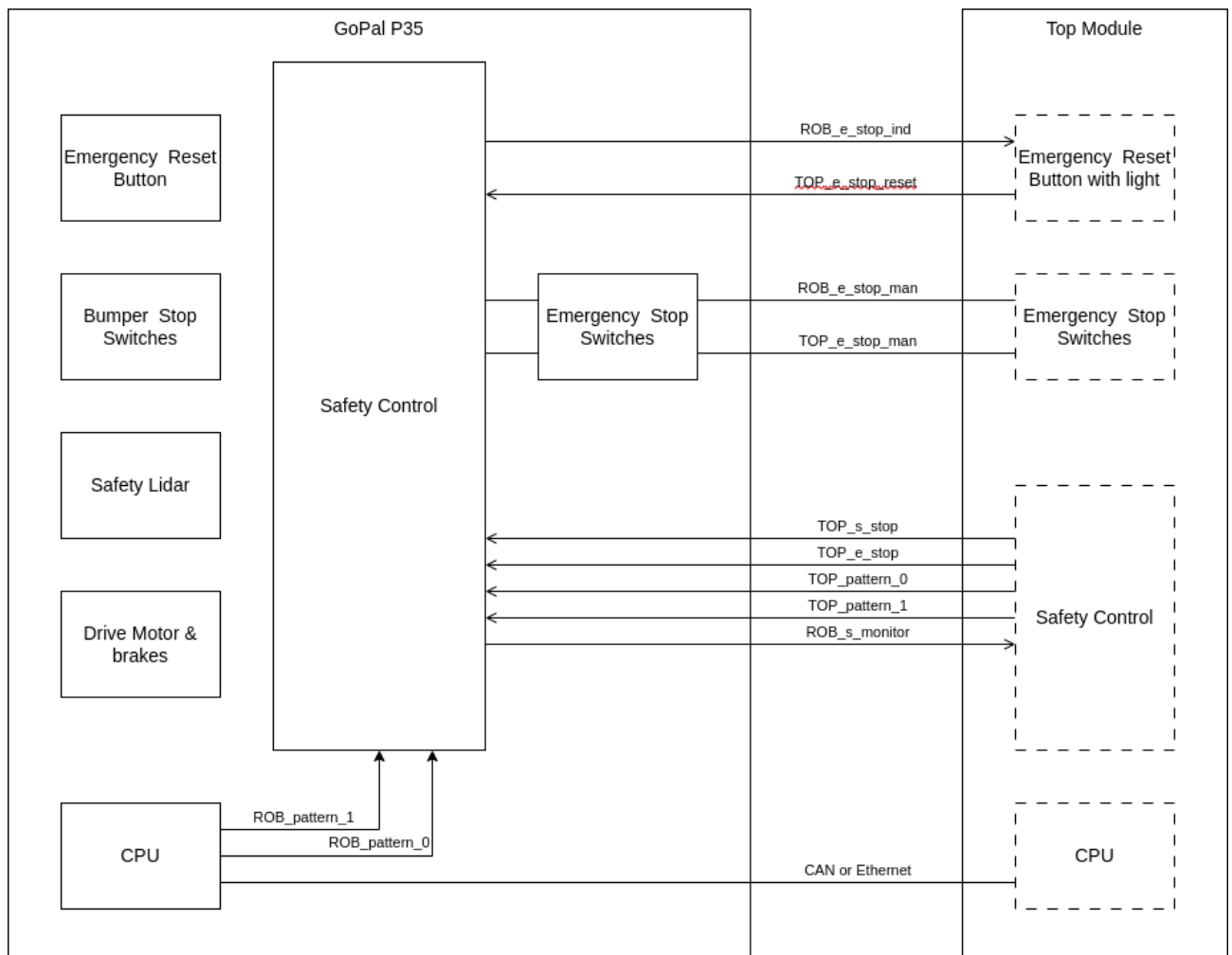


Figure 5.1 Robot/Top Module safety system overview

The safety lidars survey the robot surroundings for obstacles, and if an obstacle gets too close to the robot at its current speed, the robot's safety system will initiate safety braking. Once the obstacle is removed, the robot will automatically resume operation.

The safety lidar has a zone set, that for different speed intervals determines the positions relative to the robot where an obstacle triggers a safety stop.

At low speeds, safe operation is ensured by the robot bumper. If the bumper collides with an obstacle, the robot will emergency stop. The difference between safety stops and emergency stops is that the robot will not automatically resume an emergency stop, it requires a user to reset the emergency stop by pressing the emergency stop reset button.

Also, a person in the vicinity of the robot can activate an emergency stop by pressing an emergency stop button.

Integration of a top module with the robot's safety system can occur in several ways:

- The top module can extend the chain of emergency stop buttons on the robot, especially if the top module obstructs the robot's emergency stop buttons. Maximum 2 extra emergency stop buttons are allowed.
- Top modules may feature its own emergency stop reset button, serving as a supplement to the robot's reset button.
- The top module may have its own safety system (safety PLC), capable of triggering the robot's emergency or safety stop.
- The top module may switch between different safety configurations, each having different robot outline and lidar zone set.
- When a robot enters a station or other specific areas, muting the lidar-based safety system may be necessary. The robot, based on site configuration and top module configuration, may mute the safety system. Note that this typically requires special markings on the floor where the muting happens.

5.2 Safety signals between Robot and Top Module

To facilitate exchange of safe information between Robot and Top Module the safe signals listed below in Table 5.1 are available.

Signal	Direction	Function	Specification
TOP_s_stop	T->R	Generate safety stop. This stops the robot and prevents it from moving until the signal is deactivated.	ISO13849-1: PLd Note this signal is active low, see 4.1.5 and 4.1.6.
TOP_e_stop	T->R	Generate emergency stop. This stops the robot and prevents it from moving until the signal is deactivated and the emergency condition has been manually reset by a user.	ISO13849-1: PLd Note this signal is active low, see 4.1.5 and 4.1.6.
TOP_e_stop_reset	T->R	Manual emergency reset signal.	-
TOP_pattern_0 TOP_pattern_1	T->R	A two-bit number used for selecting the lidar zone set used for navigation and fast rotation. Four zone sets can be selected by the top module, only two of these has a special zone for fast rotation.	ISO13849-1: PLd
ROB_s_monitor	R->T	Indicate if the robot including top module is emergency stopped, where the emergency reset must be used for restarting the system after the emergency stop condition has been resolved. The following conditions triggers an emergency stop condition on the robot: 1. E-stop buttons pressed. 2. Bumper activated. 3. Robot key switched to emergency. 4. Top module e-stop activated.	ISO13849-1: PLd Note this signal is active low, see 4.1.5 and 4.1.6.
ROB_e_stop_ind	R->T	Signal for driving the reset button light. Active when reset is needed.	-

Table 5.1 Robot/Top Module safe signals

5.3 Lidar zone sets

Lidar zone sets are selected from ROB_pattern and TOP_pattern from the top module.

The TOP_pattern selects a navigation area/zone set and a fast rotation zone.

The navigation area/zone set consist of 31 areas each corresponding to a specific speed interval. The fast in-place rotation area/zone consist of 1 area and is only supported for TOP_pattern 00 and 01. Selecting the fast rotation zone enables the safety system to set the wheel speeds independently in the range from [-0.25, +0.25] m/s, in comparison to the default [-0.09, +0.09] m/s.

The ROB_pattern overrides the TOP_pattern when activating the stop or mute function, otherwise the ROB_pattern determines if the navigation area/zone set, or the rotation area/zone should be used.

For details, see Table 5.2 and Table 5.3 below.

In the default configuration only the Stop, Mute and Navigation zone set 0 are used.

The Lidar configuration may be modified by changing the area settings only. No other lidar configuration setting may be changed.

ROB_pattern_1	ROB_pattern_0	TOP_pattern_1	TOP_pattern_0	Lidar Input (EBA)	Lidar area/pattern set		
					Function	Name (Lidar setting)	Comment
0	0	X	X	111	Stop	A_stop (Area 1)	Area which always triggers a safety stop.
0	1	X	X	110	Mute	A_mute (Area 2)	Area which allows driving at a low speed without safety stop.
1	0	0	0	101	Navigation zone set 0	A_nav_0 (Area 3-33)	Set of 31 areas used for driving. Area selected by speed.
1	1	0	0	100	Rotation zone set 0	A_rot_0 (Area 34)	Area for fast rotation.
1	0	0	1	011	Navigation zone set 1	A_nav_1 (Area 35-65)	Set of 31 areas used for driving. Area selected by speed.
1	1	0	1	010	Rotation zone set 1	A_rot_1 (Area 66)	Area for fast rotation.

1	X	1	0	001	Navigation zone set 2	A_nav_2 (Area 67-97)	Set of 31 areas used for driving. Area selected by speed
1	X	1	1	000	Navigation zone set 3	A_nav_3 (Area 98-128)	Set of 31 areas used for driving. Area selected by speed

Table 5.2 Selection of lidar patterns and areas

min. speed [m/s]	max. speed [m/s]	Zone index
-2.4	-2.2	0
-2.19	-2.0	1
-1.99	-1.8	2
-1.79	-1.6	3
-1.59	-1.4	4
-1.39	-1.2	5
-1.19	-1.0	6
-0.99	-0.8	7
-0.79	-0.6	8
-0.59	-0.4	9
-0.39	-0.3	10
-0.29	-0.28	11
-0.27	-0.25	12
-0.24	-0.2	13
-0.19	-0.16	14
-0.15	0.15	15
0.16	0.19	16
0.2	0.24	17
0.25	0.27	18
0.28	0.29	19
0.3	0.39	20
0.4	0.59	21
0.6	0.79	22
0.8	0.99	23
1.0	1.19	24
1.2	1.39	25
1.4	1.59	26
1.6	1.79	27
1.8	1.99	28
2.0	2.19	29
2.2	2.4	30

Table 5.3 Area selection vs Robot speed

6 Control Interface

The control interface makes it possible for the robot and top module to share state information and enables the robot to activate functions (actions) on the top module.

6.1 Action concept

The interface is based on a simple action concept. An action is a function call with arguments started by the robot, executed by the top module, and yielding a result. See Figure 6.1.

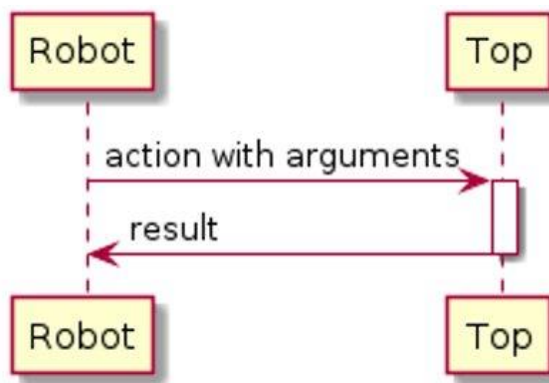


Figure 6.1 Action concept

Actions can be defined for various applications, such as:

- starting or stopping a motor,
- reading sensors,
- performing a test procedure.

The robot can automatically initiate actions in certain predefined scenarios:

- When the robot is approaching a station to allow the top module to prepare station entry while the robot is approaching the station entry.
- When the robot is at the pickup or delivery position inside the station to make the top module perform the actual pickup or delivery.
- When the robot must prepare for normal navigation to allow the top module to get into a state suitable for navigation at full speed.
- When the robot arrives at parking positions to allow the top module to execute specific tasks at a given position.
- Before the robot enters a charging station to allow the top module to perform various periodical tasks.

Actions can also be executed from the user interface. The top module will not perform any actions unless explicitly started by the robot.

Before an action can be started, its arguments must be assigned values. This can be done in the following ways:

- For actions associated with a specific site configuration element, such as a station or a parking position, the argument value can be found in the GoPal site configuration element.
- For actions activated from the user interface, the user interface should include elements to set the argument values.

The robot uses the action results for various purposes, for example to decide if a failed action should be retried or not.

6.2 Control over CAN

The CAN bus [2] allows the robot and top module to broadcast messages (data frames) to all devices attached to the bus.

A message may have between 0 and 8 data bytes. The base frame format with 11 identifier (ID) bits must be used.

The CAN bus prevents data from being corrupted but cannot guarantee that each message broadcasted will be received by a particular device.

There is one ID for the robot messages and one ID for top module messages. The information in the messages is given listed in Table 6.1 below. The “ROB_” signals are in the message broadcast by the robot and the “TOP_” signals are in the message broadcast by the top module.

Signal	Description
ROB_ready	Indicates if the robot is actively using the interface.
ROB_execute	Initiate action execution on the top module.
ROB_abort	Indicate that the top module should stop action execution if possible.
ROB_data	Robot state, action and action arguments. Robot state must always be valid while action and action arguments only must be valid when ROB execute is activated. Action and action arguments must not change while ROB execute is activated.
TOP_ready	Indicates if the top module is actively using the interface. This means that the top module must act on the information sent by the robot and that the top module must be able to execute actions when requested by the robot.
TOP_busy	Indicates if an action handling is started.
TOP_done	Indicates if an action handling is completed.
TOP_data	Top module state and action results. Top module state must always be valid while action results only must be valid when TOP_done is activated. Actions results must not change when TOP_done is activated.

Table 6.1 CAN messages

In general, the robot and the top module must send signal changes as soon as possible. The timing requirement is given in Table 6.2 below. If no signal changes occur within a period, the most recent message must be re-sent.

Time	Description
t_idle_timeout	The maximum acceptable time between consecutive two messages received from the other device. If this timeout is exceeded it must be assumed that the other side is currently not working correctly.
t_not_ready_min	The minimum duration of the signal in “not ready” state. This time must be large enough to ensure that “not ready” state can always be detected, typically a few times t_idle_timeout.
t_resp_max	<p>The maximum response time for handshakes (from signal sent to response received):</p> <ul style="list-style-type: none"> • action request to action start (ROB_execute activated to TOP_busy activated) • action done to request clear (TOP_done activated to ROB_execute and ROB_abort deactivated) • request clear to top idle (ROB_execute and ROB_abort deactivated to TOP_busy and TOP_done deactivated). <p>If the response time is exceeded it must be assumed that the other side is currently not working correctly. The response time should typically be a 3 x t_idle_timeout.</p>

Table 6.2 CAN signal timing

Figure 6.2 shows the signaling states for both the robot and the top module. In the diagram a state is represented by a named box and in the box 3 small boxes each containing a signal letter show the output signals in the state, where capital letter indicates signal is activated and lower case indicates that signal is deactivated. For the robot, the 3 boxes show the state of ROB_ready, ROB_execute and ROB_abort and for the top module the 3 boxes show the state of TOP_ready, TOP_busy and TOP_done. Arrows are events: 3 times one letter in brackets represents the input signals from the other device, timeouts are given by the name of the timing parameter.

The first robot state is ‘NOT READY’. In this state the robot signals ‘not ready’ (ROB_ready and the other signals are deactivated). After t_not_ready_min, the robot will change to ‘WAIT TOP READY’.

In ‘WAIT TOP READY’ the robot signals ready (ROB_ready activated) and where waits for the top module to signal ready and idle. When the top module signals ready and idle (TOP_ready activated, TOP_busy and TOP_done deactivated) the robot assumes that the top module is ready to execute actions. The robot changes to ‘IDLE’ state.

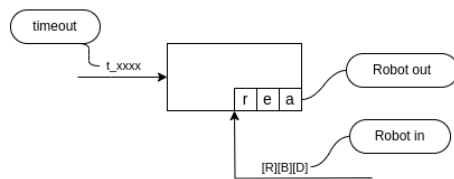
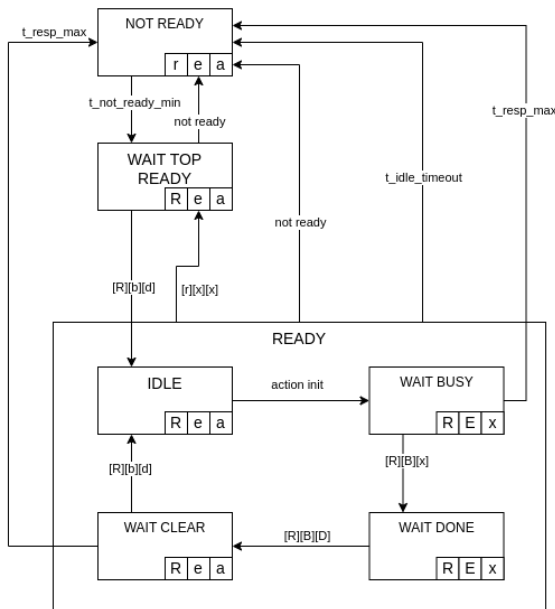
In ‘IDLE’ state the waits for system to start an action. The robot signals ready and idle (ROB_ready activated, ROB_execute, ROB_abort deactivated). When the system starts an action the robot changes to ‘WAIT BUSY’.

In 'WAIT BUSY' the robot encodes action and arguments in ROB_data and request the top module to start the action (ROB_ready and ROB_execute activated, ROB_abort deactivate). The robots wait for the top module to signal that the action has started (TOP_ready and TOP_busy activated).

In this state the robot may decide to abort the action (ROB_ready, ROB_execute and ROB_abort activated). The robot cannot assume that the top module will abort the action. If the top module aborts before the action have started ROB_busy will not be activated resulting in a t_resp_max timeout. If the top module has started the action the top module will signal that (TOP_ready and TOP_busy activated).

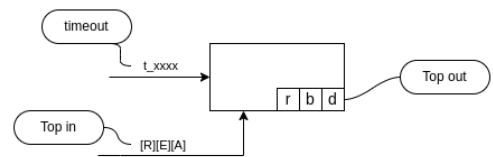
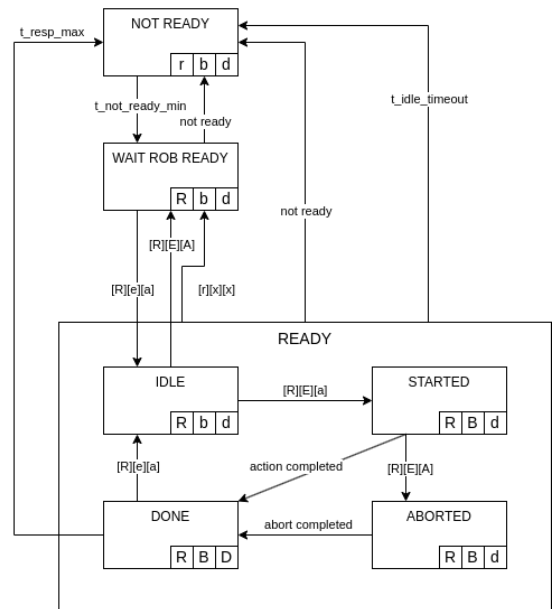
When the top module signals that the action has started the robot changes to 'WAIT DONE'.

Robot



Robot out = Top in			
	0	1	0/1
ROB_READY	r	R	x
ROB_EXECUTE	e	E	x
ROB_ABORT	a	A	x

Top Module



Top out = Robot in			
	0	1	0/1
TOP_READY	r	R	x
TOP_BUSY	b	B	x
TOP_DONE	d	D	x

Figure 6.2 Signalling states for robot and top module.

In 'WAIT DONE' the robot waits for top module signals that the action is completed. Note that there is no upper limit on the waiting time. In this state the robot may decide to abort the action (ROB_ready, ROB_execute and ROB_abort activated). The robot cannot assume that the top module will abort the action.

When the top module signals that the action is completed (TOP_ready, TOP_busy TOP_done activated) the robot changes to 'WAIT CLEAR'.

In 'WAIT_CLEAR' the robot decodes the action results and makes them available for the GoPal system and then the robot signals that the results have been used (ROB_ready activated, ROB_execute and ROB_abort deactivated) and waits for the top module signaling it is ready and idle (TOP_ready activated, TOP_busy and TOP done deactivated). When the top module signals the robot changes to 'IDLE'.

The 'READY' box holding 'IDLE', 'WAIT BUSY', 'WAIT DONE' and 'WAIT CLEAR' represents common behavior for these states:

- top module signals not ready (TOP_busy deactivated).
- top module stopped sending messages (t_idle_timeout).
- robot is no longer ready.

In both cases the robot should perform the first ready handshake as described above.

Most top module states are identical to the corresponding robot states, so no further description is needed. It is the states in the 'READY' box that differ.

In 'IDLE' the top module signals ready and idle (TOP_ready activated, TOP_busy and TOP_done deactivated). The top module waits for the robot to start an action. When the robot starts an action (ROB_ready, ROB_execute activated and ROB_abort deactivated) the top module changes to 'STARTED'. A robot may start and abort an action (ROB_ready, ROB_execute, ROB_abort activated) the top module must change to 'WAIT ROB READY'.

In 'STARTED' the top module waits for the action to complete. The top module changes to 'DONE' when the action is complete. The robot may decide to abort the action (ROB_ready, ROB_execute, ROB_abort activated) the top module may change to 'ABORTED'.

In 'ABORTED' the top module does what is meaningful for the top module and the particular action. It is important that the top module ends in a state that still allows the robot and top module to continue operation without help from an operator. The top module changes to 'DONE' when the abort is completed.

In 'DONE' the top module encodes action result in TOP_data and signals done (TOP_ready, TOP_busy, TOP_done activated) and waits for the robot to accept the results (ROB_ready activated, ROB_execute, ROB_abort deactivated). The top module changes to 'IDLE'

6.2.1 CAN message

This section defines how the information or rather information elements are encoded into the 8 data bytes in the CAN message.

All information elements are encoded as integers where MSB has the largest bit number and LSB has the smallest bit number.

The data bytes of the robot CAN message are shown in the table below. The first byte is used for the control signals, the remaining bytes are used for: robot state, action and arguments are encoded in the bit buffer D55 – D0.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	ROB_ready	ROB_execute	ROB_abort	reserved	reserved	reserved	reserved	reserved
1	D55	D54	D53	D52	D51	D50	D49	D48
...
7	D7	D6	D5	D4	D3	D2	D1	D0

Table 6.3 CAN messages, robot

The top module configuration specifies the following:

- The breakdown of robot state into information elements including number of bits needed for each element.
- The number of bits needed to identify an action.
- The number of bits needed for each action argument.

The data bytes are encoded starting from the highest bit number (D55); robot state is encoded first followed by action identification and action arguments in the sequence defined in the top module configuration.

The top module CAN message is shown in the table below. Top module state and result elements are encoded in the bit buffer D55 – D0.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	TOP_ready	TOP_busy	TOP_done	reserved	reserved	reserved	reserved	reserved
1	D55	D54	D53	D52	D51	D50	D49	D48
...
7	D7	D6	D5	D4	D3	D2	D1	D0

Table 6.4 CAN messages, top module

The top module configuration specifies the following:

- The breakdown of the top module state into information elements including number of bits needed for each element.
- The number of bits needed to identify an action.
- The number of bits needed for each result argument.

The top module CAN message is encoded starting from the highest bit number (D55); top module state is encoded first followed by the action identification and result elements in the sequence defined in the top module configuration.

Robot CAN message, data coding

D55	D54												D1	D0
robot state elements				action		arguments					not used			

Top module CAN message, data coding

D55	D54												D1	D0
top state elements				action		result elements				not used				

Table 6.5 CAN messages, coding

6.3 Control over Ethernet

The JSON RPC [1] protocol with the top module in the server role and the robot in the client role is used for this interface. The robot calls a remote method on the server and waits for a result. The JSON RPC on top of either HTTP or TCP can be used for this interface.

The top module must have a static IP configuration:

- Subnet net mask: 255.255.255.0
- Address 192.168.3.2

The top module must implement the following methods:

- `get_top_module_info`: supply information about the top module
- `start_action`: start an action
- `abort_action`: stop an action
- `get_log`: retrieve log record
- `sync_state`: synchronize interface state

6.3.1 Method: get_top_module_info

Return information contained in the “top_module” section in the top module configuration. This information may be used for validating that the top module configuration matches the actual top module.

Method parameters:

Parameter	Status	Description
None	-	-

Method results:

Parameter	Status	Description
"manufacturer"	Mandatory	See “top_module” section in the top module configuration.
"type"	Mandatory	
"version"	Mandatory	

Example:

```
Robot --> Top Module {
  "method": "get_top_module_info",
  "params": {},
  "jsonrpc": "2.0",
  "id": 15
}
Robot <-- Top Module {
  "result": {
    "manufacturer": "East",
    "type": "234-456-AB",
    "version": 3
  },
  "id": 15,
  "jsonrpc": "2.0"
}
```

6.3.2 Method: start_action

The robot calls this method to start a top module action. The top module is expected to validate the action and the action arguments and start the action if the top module is in an appropriate state.

The method takes both action and action arguments as parameters and does not return any result. The possible errors are listed below.

Method parameters:

Parameter	Status	Description
"action"	Mandatory	String that identifies the action.

"args"	Mandatory	Object containing the arguments. The arguments are defined in the top module configuration.
--------	-----------	---

Method results:

Parameter	Status	Description
None	-	-

Method error:

Error code	Error message
1001	Missing parameter
1002	Unsupported action
1003	Action argument missing
1004	Illegal action argument value
1005	Top module busy executing another action
1006	Top module is not in a state appropriate of executing actions

Example: normal

```
Robot --> Top Module {
  "method": "start_action",
  "params": {
    "action": "pickup",
    "args": {
      "test_case": 3,
      "motor_speed": 1
    }
  },
  "jsonrpc": "2.0",
  "id": 27
}
```

```
Robot <-- Top Module {
  "result": null,
  "id": 27,
  "jsonrpc": "2.0"
```

```
}
```

Example: error

```

Robot --> Top Module {
  "method": "start_action",
  "params": {
    "action": "pickup",
    "args": {
      "test_case": 7,
      "motor_spd": -128
    }
  },
  "jsonrpc": "2.0",
  "id": 82:
}
Robot <-- Top Module {
  "error": {
    "code": 1003,
    "message": "\"motor_speed\" action argument is missing"
  },
  "id": 82,
  "jsonrpc": "2.0"
}

```

6.3.3 Method: abort_action

The robot calls this method to stop or abort a top module action. The top module is expected to stop the action as soon as it makes sense for the specific action.

Method parameters:

Parameter	Status	Description
None	-	-

Method results:

Parameter	Status	Description
None	-	-

Example:**Robot --> Top Module:**

```

{
  "method": "abort_action",
  "params": {},
  "jsonrpc": "2.0", "id": 19
}

```

Robot <-- Top Module:

```

{

```



```

    "result": null,
    "id": 19, "jsonrpc": "2.0"
  }

```

6.3.4 Method: get_log

The robot uses this method to retrieve log records from the top module.

A log record is a 4-tuple (<sequence number >, <severity>, <timestamp>, <message>):

- The sequence number is modulo 65536.
- The severity ['Emergency', 'Alert', 'Critical', 'Error', 'Warning', 'Notice', 'Informational', 'Debug'] defined in RFC 5424.
- The timestamp is seconds since 1/1 1970 (Unix epoch).
- The message is limited to 1000 characters UTF-8 encoded.

Method parameters:

Parameter	Status	Description
max	Mandatory	The maximum number of log records to be included in the result

Method results:

Parameter	Status	Description
List of log records	Mandatory	List of 4-tuples (<sequence number >, <severity>, <timestamp>, <message>)

Example:

Robot --> Top Module

```

{
  "method": "get_log",
  "params": {
    "max": 3
  },
  "jsonrpc": "2.0",
  "id": 18
}

```

Robot <-- Top Module

```

{
  "result": [
    [ 725, "Debug", 1699444007, "message 725" ],
    [ 726, "Debug", 1699444107, "message 726" ],
    [ 727, "Debug", 1699444207, "message 727" ]
  ],
  "id": 18,
  "jsonrpc": "2.0"
}

```

6.3.5 Method: sync_state

The robot uses this method to synchronize the interface and retrieve top module status. The method will be called several times each second.

Method parameters:

Parameter	Status	Description
"log_ack"	Mandatory	"log_ack" is the sequence number of the most recently retrieved log record. The top module must use this to remove the retrieved log records from the log records not yet retrieved (pending).
"robot_time"	Mandatory	"robot_time" is the actual robot time in seconds since 1/1 1970. The top module should adjust its time accordingly. Note that robot time cannot be expected to be a monotonic increasing function as it may be adjusted significantly over time.
"robot_state"	Mandatory	Robot state elements defined in the rob_state list in top module config.

Method results:

Parameter	Status	Description
"control"	Mandatory	Object with the following elements: <ul style="list-style-type: none"> "state": The general state of the top module: TOP_READY or TOP_ERROR. "uptime": Number of seconds since top module (re)start. "action": name of for most recently accepted action. "action_state": is one of: "TOP_ACTION_IDLE", "TOP_ACTION_BUSY", "TOP_ACTION_DONE". "action_result" object with result values.
"log"	Mandatory	Object with the following elements: <ul style="list-style-type: none"> "pending": The number of pending log records.
"top_state"	Mandatory	Top module state elements defined in the top_state in top module config.

Example:

```
Robot --> Top Module {
  "method": "sync_state",
  "params": {
    "log_ack": 724,
    "robot_time": 1708010191,
    "robot_state": {
      "ROB_E_STOPPED": 0,
      "ROB_S_STOPPED": 0
    }
  },
  "jsonrpc": "2.0",
  "id": 84
}

Robot <-- Top Module {
  "result": {
    "control": {
      "state": "TOP_READY",
      "uptime": 360,
      "action": "pickup",
      "action_state": "TOP_ACTION_DONE",
      "action_result": {
        "TOP_OK": 1,
        "TOP_RETRY": 0
      }
    },
    "log": {
      "pending": 10
    },
    "top_state": {
      "TOP_LOADED": 0,
      "status": 1,
      "front_sensor": 0,
      "rear_sensor": 1,
      "motor_speed": 0,
      "height": 255
    }
  },
  "id": 84,
  "jsonrpc": "2.0"
}
```

7 Top Module Configuration

The top module configuration defines the properties of the top module for the GoPal system.

The top module configuration is a json formatted file that consist of the following sections:

- "top_module": identifies the top module type and revision
- "interface": specifies the actual interface configuration
- "lidar": specifies the actual lidar configuration
- "general": specifies general top module properties
- "action_args": specifies the action arguments required by the top module
- "info_items": specifies the information elements that can be used in robot state, top module state and action results
- "actions": specifies the actions supported by the top module
- "meta": additional information about the top module configuration file.

The top module configuration must be loaded into a robot when the top module is mounted on the robot.

The top module configuration must also be loaded into GoControl before robot with the top module can be used in the GoPal system.

7.1 "top_module" section

Key	Status	Type	Value description
"manufacturer"	Mandatory	String	The registered name of top module manufacturer. This name should be the manufacturer name registered by Robotize.
"type"	Mandatory	String	Top module type name.
"version"	Mandatory	Integer	Interface version number, see below

About "version"

- The version number specifies the version of the top module configuration and the implied interface and function of the top module, this means that the config version only needs to change if the interface between the robot and the top module has changed.
- GoControl will attempt to handle top modules of the same type from the same manufacturer (identical "manufacturer" and "type" key) in the same way, so the user doesn't have be concerned about top module versions. GoControl will assume that an action argument with a given name always means the same to any top module version.
- It is acceptable to add new arguments over time, but once defined the meaning and limits of the parameter must not change within top modules of the same manufacturer and type.
- The above also applies to action names.

Example:

```

"top_module": {
  "manufacturer": "East",
  "type": "234-456-AB",
  "version": 2
}

```

7.2 "interface" section

7.2.1 Ethernet

Key	Status	Type	Value description
"type"	Mandatory	String	Either "JSON_RPC_HTTP" or "JSON_RPC_TCP". "JSON_RPC_HTTP" means that the JSON RPC data is carried in the POST request and response. "JSON_RPC_TCP" means that JSON RPC data is carried in ZERO terminated frames.
"url"	Mandatory	String	URL for JSON RPC service (JSON_RPC_HTTP only)
"port"	Mandatory	Integer	Port number for the JSON RPC service (JSON_RPC_TCP only)
"interface_version"	Mandatory	Integer	Must be "1", the interface defined in this document. The interface may be extended in a backwards compatible way without changing the "interface_version".
"t_poll_period_s"	Optional	Number	Time between sync_state calls. Min 0.2s, max 2.0s, default 0.5s
"t_resp_max_s"	Optional	Number	Maximum time between request and response. Max 1.0s, default 0.5s.

Example:

```

"interface": {
  "type": "JSON_RPC_TCP",
  "port": 11000,
  "interface_version": 1,
  "t_poll_period_s": 0.1,
  "t_resp_max_s": 0.5
}

```

7.2.2 CAN

Key	Status	Type	Value description
"type"	Mandatory	String	Must be "CAN"
"bit_rate"	Mandatory	Integer	Number of bits per second on the CAN bus.

			The supported bit rates are: 250000, 500000, 1000000.
"rob_can_id"	Mandatory	Integer	CAN ID reserved for robot for outgoing messages.
"top_can_id"	Mandatory	Integer	CAN ID reserved for top module for outgoing messages.
"interface_version"	Mandatory	Integer	Must be 1, the interface defined in this document. The interface may be extended in a backwards compatible way without changing the "interface_version".
"action_bits"	Mandatory	Integer	Number of bits for coding action in the CAN message
"t_idle_timeout_s"	Optional	Number	<p>The maximum time in seconds between CAN messages sent by a device. A device must resend the most recent message if no new messages are to be sent.</p> <p>Min 0.5, max 2.0s, default 1.0s.</p> <p>The following times must be derived from t_idle_timeout_s:</p> <ul style="list-style-type: none"> • The maximum acceptable time between CAN messages received from peer device: 2*t_idle_timeout_s • The maximum acceptable response time for signal handshake (like execute to busy): 3*t_idle_timeout_s • The minimum duration of the ready signal: 3*t_idle_timeout_s

Example:

```

"interface": {
  "type": "CAN",
  "bit_rate": 500000,
  "rob_can_id": 1024,
  "top_can_id": 1025,
  "interface_version": 1,
  "action_bits": 4,
  "t_idle_timeout_s": 1.0
}

```

7.3 "lidar" section

The "lidar" section contains a list of supported lidar configurations. Currently only one lidar configuration is supported.

Key	Status	Type	Value description
"name"	Mandatory	String	Configuration name.
"front_lidar_checksum"	Mandatory	String	Front lidar checksum.
"rear_lidar_checksum"	Mandatory	String	Rear lidar checksum.
"top_pattern_0"	Mandatory	Object	See table below
"top_pattern_1"	Mandatory	Object	See table below
"top_pattern_2"	Mandatory	Object	See table below
"top_pattern_3"	Mandatory	Object	See table below

"top_pattern_x" objects:

Key	Status	Type	Value description
"robot_max_speed"	Mandatory	Number	Maximum robot speed in m/s.
"robot_outline"	Mandatory	List	Defines the robot outline used for maneuvering and navigation. The outline must be a convex polygon and is defined as a list of points in the robot frame (see below). Each point (x,y) is defined as a list (first element x, second element is y).
"sensor_view_obstacles"	Optional	List	Defines how the top module blocks the robot sensors (LIDARs and 3D CAMs). This blocking is defined as a list of blocking objects. The blocking object can be a "box" type where all box points (x,y,z) in robot frame is given by x in "x_range" and y in "y_range" and z in "z_range", where the each range is defined by a list (first element is the minimum value, second element is the maximum value). Only blocking objects in the field of view of the sensors should be defined.
"disable_fast_rotation"	Optional	Bool	"disable_fast_rotation" request not to use rotation zones, see 5.3. Default false.

Example:

```

"lidar": [
{
  "name": "normal"
  "front_lidar_checksum": "8650632DD6A34826F8B9B17EC1D24B5DC9CDABD973C1",
  "rear_lidar_checksum": "2A39AD94329971A624015A15AE53098ED92FDCF5737A53",
  "top_pattern_0":{
    "robot_outline": [[1.0,1.0], [1.0,-1.0], [-1.0,-1.0], [-1.0,1.0]],
    "robot_max_speed": 1.6,
    "sensor_view_obstacles": [
      { "type":"box", "x_range":[0.4, 0.6], "y_range":[-0.1, 0.1], "z_range":[0.3, 0.4]},
      { "type":"box", "x_range":[-0.6, -0.4], "y_range":[-0.1, 0.1], "z_range":[0.3, 0.4]},
    ],
    "disable_fast_rotation": true
  },
  "top_pattern_1":{
    ... like "top_pattern_0"
  },
  "top_pattern_2":{
    ... like "top_pattern_0"
  },
  "top_pattern_3":{
    ... like "top_pattern_0"
  }
}
]
}

```

7.4 “general” section

Key	Status	Type	Value description
“rob_state”	Mandatory	List	Specifies the info items in the robot state.
“top_state”	Mandatory	List	Specifies the info items in the top module state.
“mass”	Optional	Number	Mass [kg] of the (unloaded) top module.
“height”	Optional	Number	Height [m] of the (unloaded) top module.
“mute_safety”	Optional	String	<p>Specifies when the robot should mute safety automatically. One of the following:</p> <ul style="list-style-type: none"> • “all_stations” • “all_charging_stations” • “selected_stations” • “selected_charging_stations” • “no_stations” <p>Default “no_stations”</p>
“battery_power”	Optional	Bool	<p>Specifies whether battery power for the top module should be switched on or off.</p> <p>Default false.</p>

"passive"	Optional	Bool	Specifies whether the top module supports a communication protocol as described in Control Interface or not. True means that no control interface is supported. Default false.
"reverse_docking"	Optional	Bool	Specifies if the robot with top module can dock in both forward and reverse direction. Default false. False means that the robot can only dock in forward direction.
"sim"	Optional	Object	Object with information required by the simulator: "top_pattern": The initial top_pattern value.

Example:

```

"general": {
  "mute_safety": "all_stations",
  "rob_state": ["ROB_E_STOPPED", "ROB_S_STOPPED"],
  "top_state": ["TOP_LOADED", "status", "front_sensor", "rear_sensor", "motor_speed",
"height"],
  "sim": {"top_pattern": 1 }
}

```

7.5 "info_items" section

The "info_items" section defines all information elements used for state and action results; each information element is defined in the table below.

Key	Status	Type	Value description
"name"	Mandatory	String	The result element name. The element name must not be one of the predefined result elements below.
"type"	Mandatory	String	The type of result element is either "float", "sint", "uint" or set of specific values ("float" not for CAN). The set is an object where each key-value pair stands for a set value, the key is the string representation used for the UI, and the value is an integer representation used in the interface between the robot and top module.
"tool_tip"	Optional	String	A brief description of the argument that can be used as a tool tip in the UI.
"bits"	Mandatory	Integer	the number of bits used for the value in the CAN message (CAN only).
"sim"	Optional	Object	Object with information required by the simulator: "init": The initial value.

Example CAN:

```
"info_items": [
  { "name": "status", "bits": 2, "type": { "operational": 0, "service": 1, "alarm": 2 }, "sim": { "init": 0 } },
  { "name": "front_sensor", "bits": 1, "type": { "blocked": 0, "clear": 1 }, "sim": { "init": 1 } },
  { "name": "rear_sensor", "bits": 1, "type": { "blocked": 0, "clear": 1 }, "sim": { "init": 1 } },
  { "name": "motor_speed", "bits": 8, "type": "sint", "sim": { "init": 33 } },
  { "name": "height", "bits": 8, "type": "uint", "sim": { "init": 0 } }
]
```

“info_items” can be used as input or output from the robot control. These information elements are predefined and must not be redefined in “info_items”.

- "TOP_OK": Indicates if the most recent action was successful or not. The action is always successful if not included in the action result. Must only be included in action result.
- "TOP_RETRY": Indicates if an unsuccessful action should be retried after the robot has re-positioned. The top module configuration gives the value for this if not included in the action result. Must only be included in action result.
- "TOP_LOADED": Indicates that the goods are loaded on the top module. If not included the load status will be derived from the result of the most recent pickup or delivery action. Must only be included in top module state.
- "TOP_STOP_REQ": Indicates that the top module needs the robot to stop. Must only be included in top module state.
- "TOP_PATTERN_REQ": Indicates that the top module will switch to the TOP_pattern (see 5.2) indicated by the information element and expect the robot to change speed to avoid safety stop when the top module a while later sets up the TOP_pattern.
- "ROB_E_STOPPED": Indicates that the robot is emergency stopped. Must only be included in robot state.
- "ROB_S_STOPPED": Indicates that the robot is safety stopped. Must only be included in robot state.
- "ROB_POWER_OFF": Indicates that the robot is about to switch off power. Must only be included in robot state.

The definitions are:

```
{ "name": "TOP_OK", "bits": 1, "type": { "nok": 0, "ok": 1 } },
{ "name": "TOP_RETRY", "bits": 1, "type": { "no": 0, "yes": 1 } },
{ "name": "TOP_STOP_REQ", "bits": 1, "type": { "no": 0, "yes": 1 } },
{ "name": "TOP_PATTERN_REQ", "bits": 2, "type": "uint",
{ "name": "TOP_LOADED", "bits": 1, "type": { "unloaded": 0, "loaded": 1 } },
{ "name": "ROB_E_STOPPED", "bits": 1, "type": { "no": 0, "yes": 1 } },
{ "name": "ROB_S_STOPPED", "bits": 1, "type": { "no": 0, "yes": 1 } },
{ "name": "ROB_POWER_OFF", "bits": 1, "type": { "no": 0, "yes": 1 } },
```

7.6 "action_args" section

The "action_args" section defines all arguments for actions; each argument is defined by the properties in the table below.

Key	Status	Type	Value description
"name"	Mandatory	String	The argument name.
"type"	Mandatory	String	The type of result element is either "float", "sint", "uint" or set of specific values ("float" not for CAN). The set is an object where each key-value pair stands for a set value, the key is the string representation, and the value can be a "sint" or an "uint" dependent on the set values. If no negative values the type is assumed to be "uint" otherwise the value will be "sint".
"min", "max"	Mandatory	"float", "sint", "uint"	The valid range for "float" and "int" only.
"default"	Mandatory	"float", "sint", "uint", or set	The default value. The default value for a set type is the equivalent integer.
"tool_tip"	Optional	String	A brief description of the argument that can be used as a tool tip in the UI.
"bits"	Mandatory	Integer	the number of bits used for the value in the CAN message (CAN only).

Example (Ethernet):

```
"action_args": [
  { "name": "height", "type": "float", "min": 0.35, "max": 0.55, "default": 0.40,
    "tool_tip": "height above the floor" },
  { "name": "speed_in", "type": "int", "min": 2, "max": 8, "default": 6 },
  { "name": "speed_out", "type": "int", "min": 2, "max": 8, "default": 6 },
  { "name": "mode", "type": {"good": 0, "bad": 1, "default": 1}
}
```

Example (CAN):

```
"action_args": [
  { "name": "height", "bits": 2, "type": {"low": 0, "normal": 1, "high": 3}, "default": 0 },
  { "name": "motor_speed", "bits": 8, "type": "sint", "min": -100, "max": 100, "default": 0 },
  { "name": "test_case", "bits": 8, "type": "uint", "min": 0, "max": 255, "default": 0 }
]
```

7.7 "actions" section

The "action" section defines all actions; each argument is defined by the properties in the table below.

Key	Status	Type	Value description
"name"	Mandatory	String	The action name must be unique.
"type"	Mandatory	List	Each list element defines a scenario where the action may be used. List element can be any of the following: <ul style="list-style-type: none"> • PREP_CHARGE, • INTERMEDIATE_PARKED, • PREP_PICKUP, • EXEC_PICKUP, • PREP_DELIVER, • EXEC_DELIVER, • PREP_NAV, • OTHER
"code"	Mandatory	Integer	The integer number used for the action (CAN only)
"tool_tip"	Optional	String	A brief description of the action that can be used as a tool tip in the UI.
"args"	Mandatory	List	list of arguments
"results"	Mandatory	List	list of results
"rob_stopped"	Optional	Bool	The robot must remain stopped while the action is running. Default false.
"top_pattern"	Optional	Integer	The top module will switch to the "top_pattern" (see 5.2) when the action is being executed. The value -1 indicate that no specific change is expected. The default value is -1. The robot must change speed before the action is started, so the top module can select "top_pattern" without triggering a safety stop.
"auto_retry"	Optional	Bool	An action must be retried if it fails. Default false.
"operator_ui"	Optional	Bool	This action should be available for the normal robot operator. Note that the UI will not allow the robot operator to edit arguments, so all arguments will have default value.
"sim"	Optional	Object	Object with information used by the simulator: "normal": array of objects that defines a simulation step as "wait": <seconds to sleep>, "top_state": <top module state change>,

			<p>"top_pattern": <top pattern state change> "result": <action result> (last simulation step only). "on abort": like "normal" if the action is aborted.</p>
--	--	--	---

Example CAN:

```

"actions": [
  { "name": "before_navigation", "type":["PREP_NAV"], "code":0, "args": [],
    "results":["TOP_OK"], "robot_stopped" : false },

  { "name": "before_entry", "type":["PREP_PICKUP","PREP_DELIVER"], "code":1, "args":
    ["height"], "results":["TOP_OK"], "robot_stopped" : false,
    "sim": {
      "normal": [
        { "wait": 1.0, "top_state":{"height":3}, "results": {"TOP_OK":1}}
      ]
    }
  },

  { "name": "pickup", "type":["EXEC_PICKUP"], "code":2, "args": ["motor_speed"],
    "results":["TOP_OK", "TOP_RETRY"], "robot_stopped" : true,
    "sim": {
      "normal": [
        { "wait": 0.1, "top_state":{"motor_speed":100}},
        { "wait": 1.0, "top_state":{"front_sensor":0}},
        { "wait": 5.0, "top_state":{"TOP_LOADED":1, "top_pattern": 2},
        { "wait": 1.0, "top_state":{"front_sensor":1}},
        { "wait": 0.3, "top_state":{"motor_speed":0}},
        { "results": {"TOP_OK":1, "TOP_RETRY":0}}
      ]
    }
  },

  { "name": "delivery", "type":["EXEC_DELIVER"], "code":3, "args": ["motor_speed"],
    "results":["TOP_OK"], "robot_stopped" : true },

  { "name": "home", "type":["PREP_CHARGE"], "code":4, "args": [], "results":["TOP_OK"],
    "robot_stopped" : false },

  { "name": "raise", "type":["OTHER"], "code":5, "args": [], "results":["TOP_OK"],
    "robot_stopped" : true },

  { "name": "lower", "type":["OTHER"], "code":6, "args": [], "results":["TOP_OK"],
    "robot_stopped" : true,
    "sim": { "normal": [ { "wait": 1.0, "top_state":{"height":0}, "results": {"TOP_OK":1}} ] }
  }
]

```

7.8 Removed

7.9 "meta" section

Key	Status	Type	Value description
"format_version"	Mandatory	Integer	<p>The definition of the top module config file (the json file with objects and structure) defines an interface to the system and the version of this interface is defined as "format_version".</p> <p>The interface may be extended in a backwards compatible way without changing the "format_version".</p> <p>"format_version" must be 1 for the interface described in this document.</p>

8 Appendix A

By default, the power for the Top Module is taken from the robot's Power Control Interface Board. This board is located in the robot's rear electronics compartment. To get access to the board, the top cover over the electronics compartment must be removed. For board location see Figure 8.1 below.

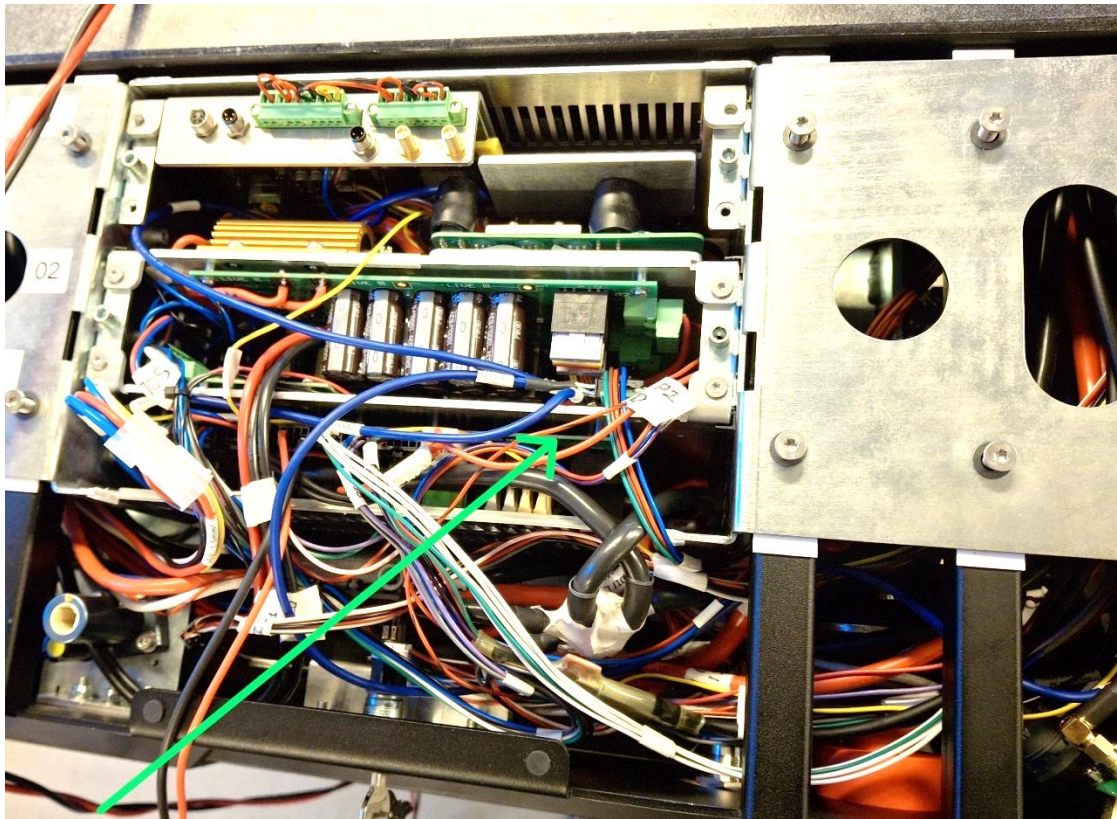


Figure 8.1 Location of Power Control interface Board (green arrow)

The relevant wire is mounted at the screw terminal marked TOP. See Figure 8.2 below. Connection is by a ring terminal.

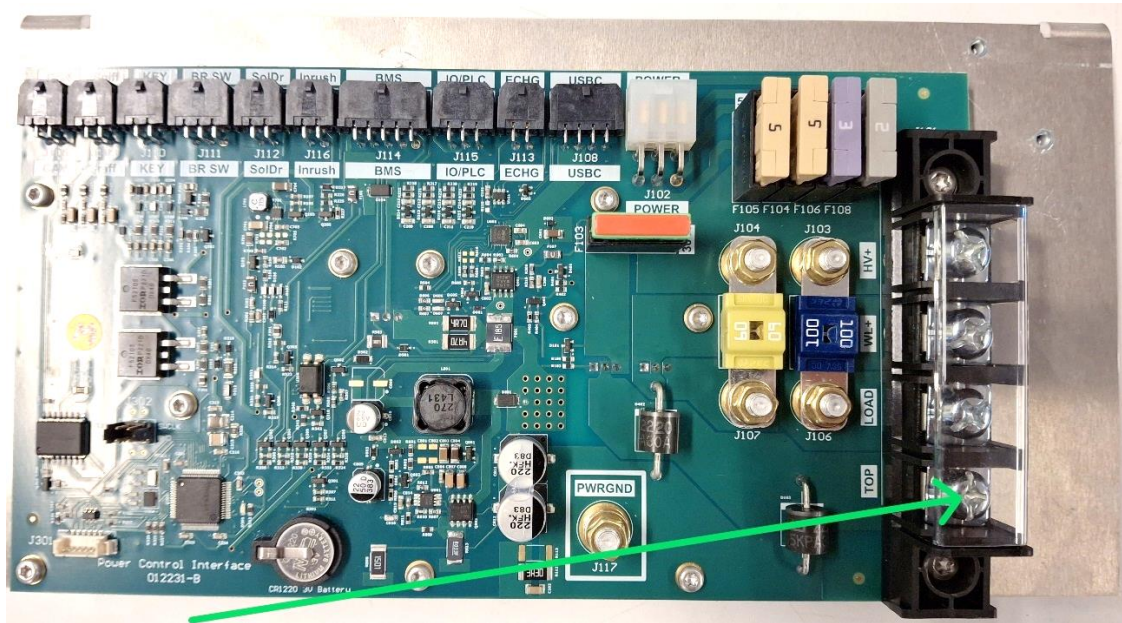


Figure 8.2 Location of the “TOP” terminal on the Power Control interface Board (green arrow)

To change to the alternative power source, the ring terminal must be removed from Power Interface Board screw terminal.

The ring terminal must be cut off the wire end, and the wire end stripped (~10 mm).

To source power from after the robot’s safety relays, the wire end must be connected at the Brake Load Dump Board. This board is also located the robot’s rear electronics compartment. See Figure 8.3 below.

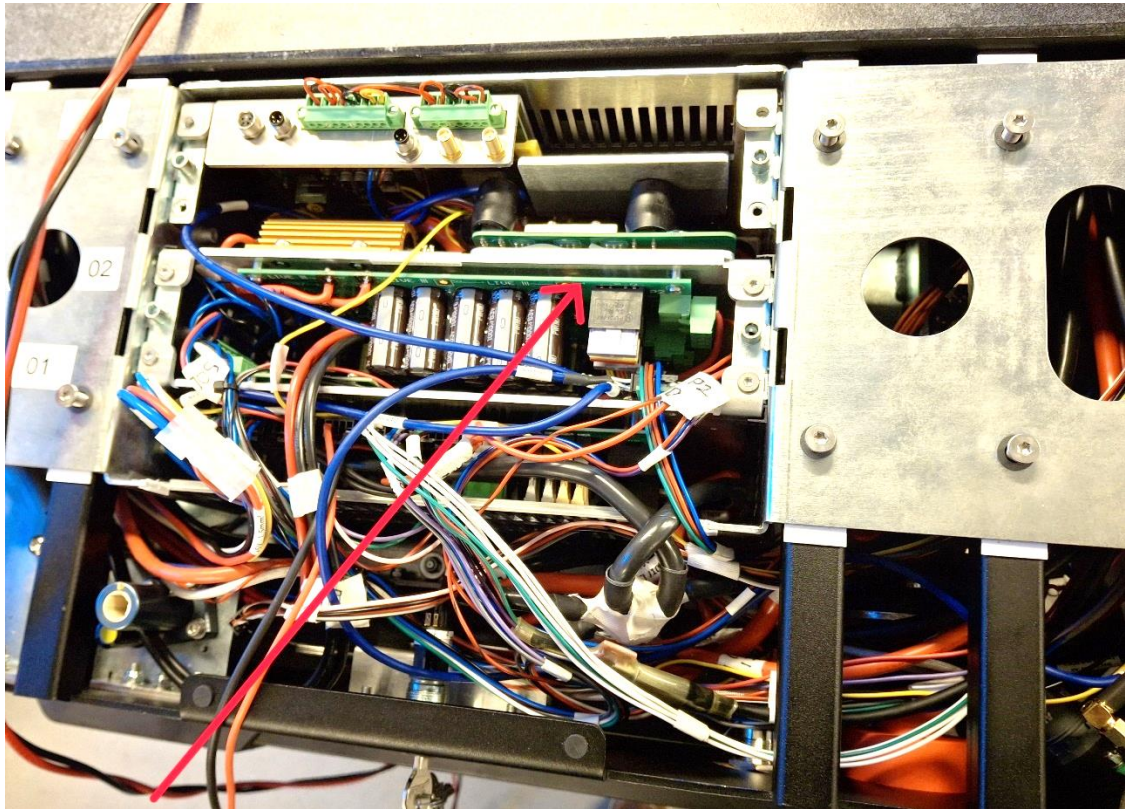


Figure 8.3 Location of Brake Load Dump Board (red arrow)

Connect the stripped wire end to connector P2 “LIFT” screw terminal output. See Figure 8.4 below.

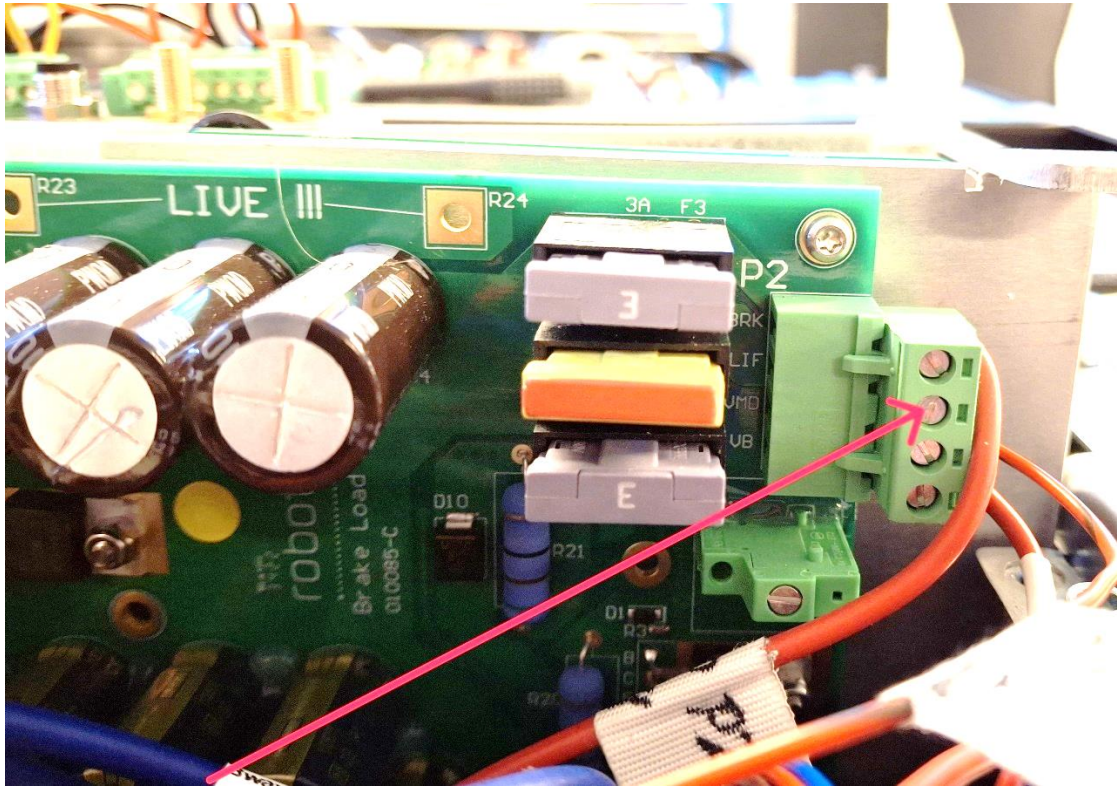


Figure 8.4 Location of the “LIFT” terminal on the Brake Load Dump Board (red arrow)