

Standard Top Module Interface

This document is an early draft. The content will change and should not be used without consulting the author.

Revision history

Version	Date	Author	Description
0.1	25-03-2024	MIV AEF	Initial version.
0.2	14-03-2025	AEF	Updates to type and telegrams.
	24-06-2025	PDA	Release

Review history

Version	Date	Reviewer	Notes / description
0.1	03-04-2024	PPO	Review
0.2	14-03-2025	MIV	Review
	24-06-2025	PDA	Release

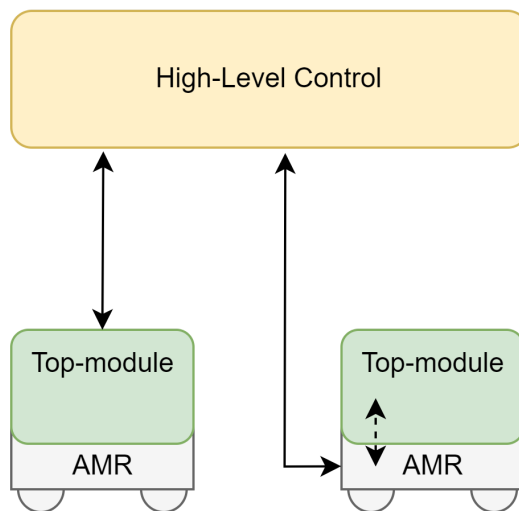
Table of Contents

1 Introduction	4
1.1 References	4
1.2 Glossary	5
1.3 Copyright and right to use under Apache License 2.0	5
1.4 Background / 5G-Robot	5
2 Domain Model	7
2.1 Terms and Concepts	7
2.2 Container Movement Concept	7
2.3 Container Operations Concept	8
2.4 Act on Environment Concept	8
3 Communication	9
3.1 Using the MQTT Protocol	9
4 Common Data Types	10
4.1 Action Property	10
5 Error Types	11
6 Common Telegrams	12
6.1 Info-request Telegram (HLC <-> TMOD)	12
7 Actuators	13
7.1 Actuator State (TMOD -> HLC)	13
7.2 Action Request (HLC -> TMOD)	13
7.3 Action Reply (TMOD -> HLC)	14
7.4 Action Completed (TMOD -> HLC)	15
8 Module Type Examples	16
8.1 Move Pallet	16
8.1.1 Actuator Status	16
8.1.2 Module Configuration	16
8.1.3 Actions	16
8.2 Move Containers with Arm	17

1 Introduction

This document is a specialization of the standard equipment interface [1] that describes the standard interface for top modules on mobile robotic systems. This interface is used for communication between a high-level control (HLC) system and a top module.

Depending on the top module and the mobile robotic system used, the telegrams in this interface can be sent directly from the top module to a high-level control (HLC) system, or the telegrams can be sent from the top module through the mobile robotic system to the HLC. Both scenarios will use the same interface.



Detailed descriptions of the communication protocol, telegram format, and common telegrams used by this interface can be read in the standard equipment interface document [1]. Readers are advised to become familiar with that document before implementing the standard palletizing interface.

The document is currently a **draft** for the interface and is purely conceptual. *Major changes will be made to it* and a number of iterations are required before it is ready to be used for the design of a system using it. Notes and comments will be highlighted as **red text** or have an associated comment in google docs.

In order to support top modules that are not described in the interface description, the actual content of the interface can be adjusted to handle a given job. This interface will show a number of top modules and what content may be exchanged with them.

1.1 References

ID	Document	Description
[1]	Standard Equipment Interface v2.2	The base interface description used for all equipment / robot cell interfaces. Found on Intelligent Systems website: https://www.intelligentsystems.dk/products-keep-customers-at-forefront-of-technology/
[2]	Domain Model v1.3	Defines various terms and concepts used for the domain. Found on Intelligent Systems website: https://www.intelligentsystems.dk/products-keep-customers-at-forefront-of-technology/

[3]	MQTT Protocol for Standard Interfaces v1.0	Protocol specification for using MQTT with the standard open interfaces. Found on Intelligent Systems website: https://www.intelligentsystems.dk/products-keep-customers-a-t-forefront-of-technology/
[4]	Standard Interface v1.0	The base interface description used for all the open interfaces. Found on Intelligent Systems website: https://www.intelligentsystems.dk/products-keep-customers-a-t-forefront-of-technology/

1.2 Glossary

Abbreviation	Description
TMOD	Top module - robot controller or PLC controlling the top module
HLC	High-Level Control system - e.g. WCS/WMS, MES or MFS.
AGV	Automated Guided Vehicle - An AGV is a mobile robot that navigates using defined tracks in an environment. Tracks can be defined using magnetic tape or wires or virtually drawn.
AMR	Autonomous Mobile Robot - An AMR is a mobile robot that navigates autonomously with its sensors in an environment. AMRs are highly flexible as these do not require the tracks used by AGVs.

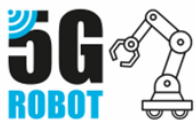
1.3 Copyright and right to use under Apache License 2.0

The copyright on this document and any contained specifications, designs, etc. belongs to the company: Intelligent Systems A/S, Havnevej 11, DK-9560 Hadsund, DENMARK.

Any referenced 3rd party designs, technologies and or other IP shall remain with the original owners.

The reference architecture, designs and the included open standard integration interfaces are open and free to use under Apache License 2.0. For details please see <https://www.apache.org/licenses/LICENSE-2.0>.

1.4 Background / 5G-Robot



Parts of this document / release was made in the **5G-Robot** project also known under the long name **5G-ENABLED AUTONOMOUS MOBILE ROBOTIC SYSTEMS** - the largest innovation project that has been launched under the Innovation Fund Denmark's (IFD) Grand Solutions program.

The groundbreaking project united Denmark's leading robot, automation and factory digitalization companies as technology vendors, research partners and industry-leading end-user companies.



Illustration: Project partner logos.

The aim of the project was to revolutionize manufacturing - paving the way to smart production and smart factories and the application of a number of new technologies in production and manufacturing including 5G wireless communication, cloud and edge computing and digital twin.

Intelligent Systems played a leading role in the project, providing the glue that ties the robotic solutions of the partners together making the work as one - i.e. one connected integrated intelligent manufacturing system.

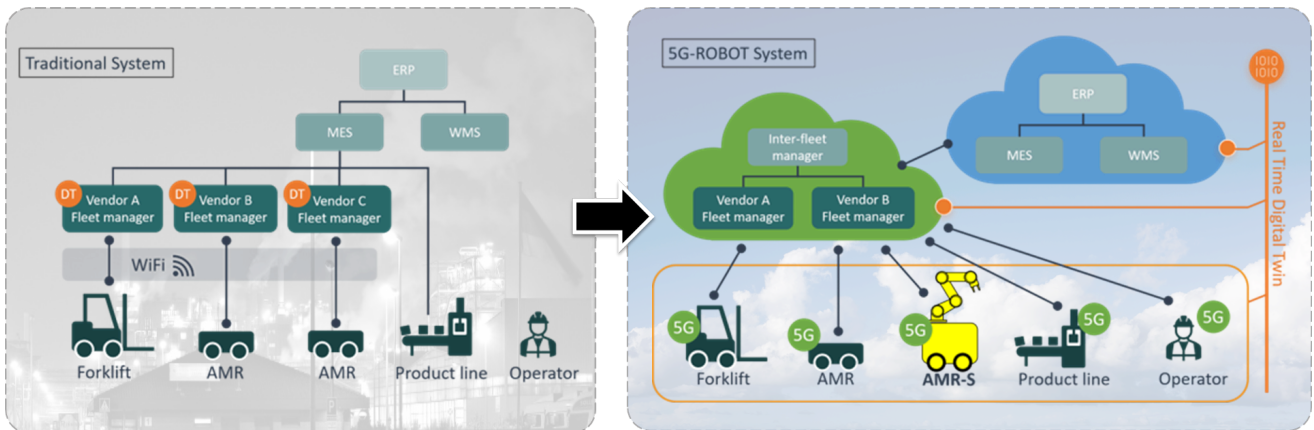


Illustration: The aim is to revolutionize manufacturing paving the way to smart production and smart factories.

Read more about the 5G-Robot project here: <https://www.5gsmartproduction.aau.dk/5g-robot>

2 Domain Model

It is recommended to familiarize with the *equipment* concept in the domain model.

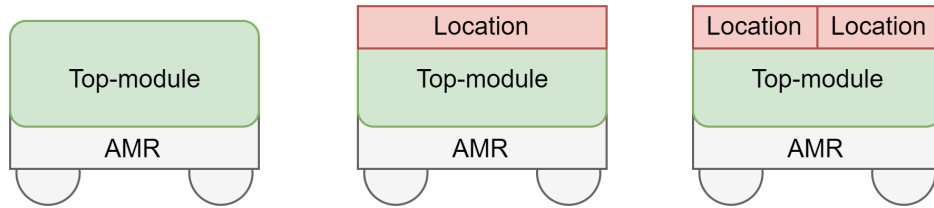
A top module is mounted onto a mobile robot, with both being designated as equipment. This equipment can have 0-to-many locations intended for having containers placed on them. The equipment is described by its equipment profile, which contains information such as how the equipment is configured and what its capabilities are. The capabilities relevant for the instance of equipment depends on the type - for top-modules, one example of a capability is what type of containers it can hold (boxes, pallets, half-pallets etc.).

2.1 Terms and Concepts

Term	Description
Container Concept	
Item	Describes a distinct physical object (or a number of indistinguishable items).
Container	Generally designates entities which can contain other items or containers, primarily for the purposes of transport and packaging. A container can be a box, pallet, carton or other similar items.
Pallet	A generalization of a container that can be palletized with items and/or containers.
Location	A place where an object exists. This can either correlate with a physical location or can be used for logical grouping. Typically used for specifying where containers or items are located.
Equipment Concept	
Equipment	A piece of or a group of physical hardware, such as collaborative robot manipulators, actuators, and barcode scanners.
Equipment Capability	Describes what the equipment can be used for. An example of a capability for a top module can be which types of containers it can support (boxes, pallets, half-pallets etc.)

2.2 Container Movement Concept

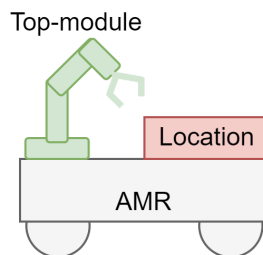
In the example below, the top module is used to move one or more containers on or off an AMR. An AMR can have a single top module. The module may be more advanced and consist of multiple subsystems, but is still regarded as a single unit from the perspective of the HLC. A top module can have 0-to-many locations.



2.3 Container Operations Concept

Here, the top module is able to perform operations on containers/locations it can reach. This means that it can be used to act in areas of other equipment. An example of this could be picking up or delivering a container from/to a location in a production cell.

The top module may have additional features such as a scanner (RF, barcode reader, ...) to verify the identity of the containers.

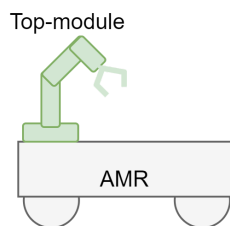


2.4 Act on Environment Concept

In this concept the top module can perform actions on the surrounding environment. The AMR will not have any location on it and may therefore only act upon things in the environment.

Examples:

- Scanning the environment to verify an area is cleared of items.
- Verification after a truck driver has picked up pallets from a staging lane, or check that an area does not contain any objects which may interrupt operations.
- A printer which can apply labels to pallets automatically after wrapping.



3 Communication

The telegram format for the interface can be read in the standard equipment interface document [1].

3.1 Using the MQTT Protocol

Refer to [3] to see details for implementing the interface using the MQTT protocol.

Note that this interface overrides the MQTT protocol with the following values:

- Prefix is set to *top-module*
- Publisher is set to *tmod* for clients and *hlc* for the HLC

Example topic for a top module publishing its state:

top-module/module-174/tmod/system/state

4 Common Data Types

This section contains data types that will be used throughout the specification.

4.1 Action Property

Data type: **ActionProperty**

The action property is a key-value pair of strings. Valid action properties must be specified by the vendor in a specification that documents which version of the equipment is able to accept which action properties.

Property	Type	Description
name	String	Name of the action property. All properties must use underscores instead of dashes.
value	String	Value of the property. Non-string values should be sent as the string-representation of that type.
type	String	The type of the value, in lowercase. Must be one of the formats described in [4]. Used by the receiver to validate and parse the value correctly.

Example

The following example shows action properties of various data types.

```
{
  "name": "type",
  "value": "Conveyor",
  "type": "string"
}

{
  "name": "mat_group_1_speed",
  "value": "5",
  "type": "number"
}

{
  "name": "current_speed",
  "value": "2.4",
  "type": "number"
}

{
  "name": "is_active",
  "value": "true",
  "type": "boolean"
}

{
  "name": "startup_time",
  "value": "2023-01-19T09:01:50Z",
  "type": "date"
}
```

5 Error Types

This section extends the error types presented in the standard equipment interface document [1].

The following list of errors are the ones referenced in this interface. The relevant Error IDs are listed for each telegram of which it is relevant.

Error ID	Error Message	Reason
MISSING_PARAM	Parameter missing	Validation showed some information from HLC is missing. The name of the expected parameter that was missing should be sent as a resource-id.

6 Common Telegrams

6.1 Info-request Telegram (HLC <-> TMOD)

Telegram Type ID: info-requests

This extends the info-request telegram found in the standard equipment interface [1] with an additional request type, namely the ability to request equipment capabilities from HLC or TMOD. The HLC or top module should send a corresponding telegram on the relevant topic.

These messages are primarily used for (re-)synchronization. They will not be used for e.g. requesting a state change for the top module.

Request type	Value	Sender	Response telegram
Capabilities	capabilities	HLC	Capabilities (see [1]).

Properties

Property	Type	Description
request	String	Requested resource.

Example

```
{
  "header": {...},
  "request": "state"
}
```

7 Actuators

7.1 Actuator State (TMOD -> HLC)

Telegram Type ID: actuator.state

Sent by the top module when the state of an actuator has changed, or when requested by the HLC using the [Info-request Telegram](#).

The available states depend on the top module type. For reference, see [Module Type Examples](#).

If the state of an actuator changes frequently, e.g. the current speed of a conveyor or robotic arm, the updates should only be sent at most once per second.

Properties

Property	Type	Description
states	List of Property	See Property .

Example

```
{
  "header": {...},
  "states": [
    {
      "name": "lift_status",
      "value": "raised",
      "type": "string"
    },
    {
      "name": "lift_position",
      "value": "200",
      "type": "number"
    }
  ]
}
```

7.2 Action Request (HLC -> TMOD)

Telegram Type ID: command.request

Telegram transmitted by the HLC. This telegram is sent when a top module action is required. For a given action all the relevant properties are sent and the top module is expected to validate that enough information has been received.

The top module must respond to the request using the telegram [Action reply](#). If the transport is accepted, the top module must send an [Action completed](#) upon completion.

Properties

Property	Type	Description
request_id	UUID	ID of the request.
action_properties	List of ActionProperty	See ActionProperty .

Errors

The following error types can be returned by the top module for this request:

- MISSING_PARAM

Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "action_properties": [
    {
      "name": "From-location",
      "value": "A",
      "type": "string"
    },
    {
      "name": "To-location",
      "value": "AMR-1",
      "type": "string"
    }
  ]
}
```

7.3 Action Reply (TMOD -> HLC)

Telegram Type ID: command.reply

Telegram transmitted by the top module. Used as a response for an [Action request](#).

If accepted, the top module will immediately start the requested action after sending this telegram. This includes changing the PackML state to Starting and then Execute.

Properties

Property	Type	Description
request_id	UUID	ID of the transport request.
response	Boolean	True if confirmed, false if rejected.

Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "response": true
}
```

7.4 Action Completed (TMOD -> HLC)

Telegram Type ID: command.completed

Telegram transmitted by the top module. Used to signal that the requested action has been completed (see [Action request](#) and [Action reply](#)).

When completed, the top module should go into a Completed state, and then automatically reset itself to the Idle state.

Properties

Property	Type	Description
request_id	UUID	ID of the transport request.
timestamp	Date	Timestamp of when the transport was completed.

Example

```
{
  "header": {...},
  "request_id": "62d7a39a-97e2-11ed-a8fc-0242ac120002",
  "timestamp": "2023-01-19T09:01:50Z"
}
```

8 Module Type Examples

In this chapter, a number of top modules are described. These are to be considered initial examples and more examples may be provided in a future iteration.

8.1 Move Pallet

For movements of pallets, the following parameters are considered relevant in different scenarios.

8.1.1 Actuator Status

Property	Type	Description
lift_status	String	For top modules capable of lifting objects, e.g. pallets, this property represents the current state. The actuator is either "raised" or "lowered".
lift_position	Integer	For top modules capable of lifting objects, e.g. pallets, this property represents the current height of the lift in mm, where a value of 0 represents the lift being lowered.
conveyor_speed	Decimal	For top modules with conveyors, this property represents the current speed in m/s.
robot_arm_in_home_position	Boolean	For top modules with robotic arms, this property indicates whether the arm is in a home/safe position.

8.1.2 Module Configuration

Property	Type	Description
default_speed	Decimal	This is the default speed of the top module action. This could for instance be the speed of the rollers.
default_accel	Decimal	This is the default acceleration of the top module action. This could for instance be the acceleration of the rollers when moving a pallet.
actual_speed	Decimal	For some material - like liquids - it is relevant to slow down the top module. This parameter specifies speed for the given action.
actual_accel	Decimal	For some material - like liquids - it is relevant to slow down the top module. This parameter specifies acceleration for the given action.

8.1.3 Actions

Property	Type	Description
pick_up	String	Because top modules are different, this is a generic command to pick up a pallet. Some top modules lift up a pallet from underneath while other top modules activate rollers to roll the pallet on.
deliver	String	Because top modules are different, this is a generic command to deliver a pallet. Some top modules lower a pallet from underneath while other top modules activate rollers to roll the pallet off.

8.2 Move Containers with Arm

For movement of containers the following parameters are considered relevant in different scenarios.

Property	Type	Description
from_location	String	The location to pick up a container from.
to_location	String	The location to deliver a container to.
temp_location	String	If swapping containers then this is a possible temporary location to use.
action_type	String	Action types can be <i>Move</i> or <i>Swap</i> . This depends on the capability of the top module to handle multiple complex operations in one.
add_separator	Boolean	If containers may be stacked on the destination it may be necessary to add a kind of separator. The separator could be a skeleton of some sort or it may be a piece of cardboard.
remove_separator	Boolean	If containers are stacked then it may be necessary to remove a kind of separator before picking up the container. The separator could be a skeleton of some sort or it may be a piece of cardboard.