Technical Delivery Specification

MC01 – Machine Connectivity



Revision status:

This technical delivery specification, „MC01 – Machine Connectivity“ replaces all previous specifications.

Version:	Site/s:	Description of the revision:	Date:
V 1.0	all	Creation of the Document	27.05.2021
V 1.1	7,8,9	Changes in the Specification	15.06.2022
V 1.2	all	Clarifications and Corrections	19.10.2022
V 2	all	Changes in the Specification	09.05.2024
V 2.1	all	Clarifications and Corrections	09.09.2024
V 2.2	all	Changes in the Specification	21.07.2025
V 2.3	8,9	Changes in mqtt-topic layers	03.12.2025
V 2.4	all	Clarifications and Corrections	09.02.2026
V 2.5	11,12,13	Small changes	03.04.2026
V 2.6	14,15,17,18	Small naming changes, combine EPP in one topic, add reset topic after ACK, add prefered data type	06.11.2026

Responsible:	Hirschmann Automotive GmbH
Department:	IT_SS_SF

Copyright © 2025

All rights reserved.

Reprinting, including excerpts, is only permitted with the approval of Hirschmann Automotive GmbH.

HIRSCHMANN AUTOMOTIVE

Table of Contents

1 General 4
1.1 Area of application 4
1.2 Deviations 4
1.3 Regulations, norms, and industry standards 4
1.4 Pre-acceptance 4
2 Machine connectivity 5
2.1 General 5
2.2 Important Notes 5
2.3 General workflow 5
3 Network interface 6
3.1 Interface for smart connection (single PLC) 6
3.2 Interface for smart connection (multiple devices) 6
3.3 Interface for Remote access 6
3.4 Network configuration 6
4 MQTT 7
4.1 MQTT security and connections 7
4.2 Quality of service level (QoS) 7
4.3 Topic naming structure 8
4.4 Last will message 9
4.5 Topic data 9
5 OPC UA 10
5.1 Network interface 10
5.2 OPC UA server security and connections 10
5.3 OPC UA Node ID naming methodology 10
6 MES capability level 1 11
6.1 Basic machine information 11
6.2 Multiple stations 12
6.3 Error proof part test (EPP-test) 12
6.4 Packing unit 12
6.5 HMI 13
7 MES capability level 2 13
8 MES capability level 3 14
8.1 Machine information 14
8.2 Triggers 14

1 General

1.1 Area of application

This Technical Delivery Specification (TDS) defines the standards for the connectivity of machines, systems, and production facilities of Hirschmann Automotive (HA).

The purpose of this document is for the supplier to fully understand what hardware and software they need to prepare for HA to be able to monitor and control their machine. Since the machines are built project specific, some minor deviations are possible, but this document should be enough for an official offer to be done without additional costs for machine connectivity later during the project.

1.2 Deviations

Any deviations from this TDS require written approval from HA. If there is any ambiguity or unclear specifications, please contact HA.

1.3 Regulations, norms, and industry standards

The contractor is fully responsible for adhering to and fulfilling any requirements resulting from applicable regulations, norms, and industry standards, even if not explicitly stated in this TDS.

Any specifications referenced in this TDS should be checked by the contractor against the most current regulations, norms, and industry standards.

Suppose the contractor identifies any problems arising from the specifications in this TDS that would render the services provided by the contractor partially or completely unsuitable for the intended purpose, in that case, the contractor must inform HA about this immediately.

1.4 Pre-acceptance

The pre-acceptance is carried out remotely. The supplier should present the implemented structure and demonstrate the functions before the delivery of the machine.

Ideally, the pre-acceptance will take place in a Microsoft Teams meeting in which the control technicians who programmed the machine present the interface and the functions. Questions and ambiguities can be clarified quickly here.

2 Machine connectivity

2.1 General

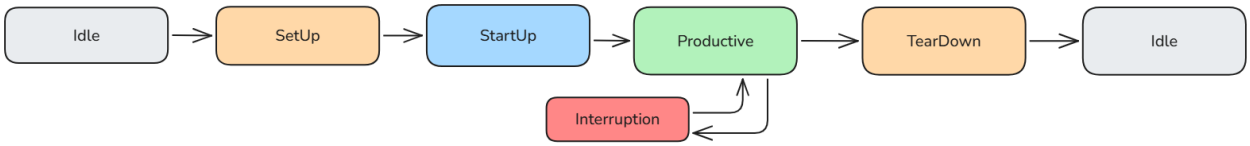
HA uses different communication protocols to communicate and control the production machines. The standards are in the following order, 1. MQTT, 2. OPCUA, any of them can be used, with MQTT being preferred.

2.2 Important Notes

Since communication with the machine involves triggers to ensure the delivery of the data in a safe storage space, the machine **must** provide a trigger-free option where the machine is not dependent on external systems. **This switch must be available in the HMI under higher administrative permissions**. After the button has been pressed, the machine operates for 10 minutes without a trigger, after which it returns to normal operation. When MQTT protocol is used, a broker-free option where the machine is not waiting for QoS, (see section 4.2) must also be provided.

If neither MQTT nor OPCUA is available, the contractor should inform HA to establish the minimum requirements for the machine.

2.3 General workflow



Idle: No recipe/order loaded on the machine, the machine can be used in this state for other purposes (planned maintenance, validation, tests, etc.). **Any other state can go to idle by canceling the order.**

Set Up: Order was received from HA + **any process that doesn't consume raw materials**

Start Up: Finished parts or semi-finished parts that are **not** part of the order**, will be destroyed by HA** following quality processes

Productive: Normal productive mode**, only these parts will be counted towards the order quantity**

Interruption: Error mode, when machine is stopped for any reason

Tear down: After reaching the Order target amount the machine sends the Teardown state and make a soft stop.

The current state should be found in the **OperationMode** data point. The actual triggers should be established together with HA after the design of the machine is finished. Going back to a state is not possible. Start Up can be omitted if it's not done.

3 Network interface

3.1 Interface for smart connection (single PLC)

When a single PLC is used for controlling the machine, the PLC should have multiple networks available (at least two RJ45 ports) one for the machine local network, the other for the OPC UA / MQTT access.

Note: These must be independently configurable networks, not just two switched ports within the same network.

3.2 Interface for smart connection (multiple devices)

The machine should have an RJ45 port as an interface. This port must be connected to a router that provides externally accessible services via port forwarding. The router firewall must be configured to allow connections from inside the machine to the outside while blocking connections from the outside to the machine's internal devices (except for the forwarded ports).

HA must have access to configure this router as well as change the port forwarding rules.

3.3 Interface for Remote access

It's mandatory that suppliers have remote access to their machines. For this purpose, a deactivatable RJ45 port must be provided, separate from the one used for the smart connection.

The standard for this access is the eWon router from HMS Networks. If it is not possible to use this device or if there are any questions regarding its use, HA must be contacted to clarify the questions and available options.

3.4 Network configuration

Static IP addresses are assigned to the router and the connection for remote access. HA provides the configurations to the supplier. The supplier shall provide a list of the ports forwarded by the router on request.

The machine should also be correlated with a time server. The IP addresses for the time servers will be provided by HA.

4 MQTT

4.1 MQTT security and connections

HA operates MQTT brokers at each location, minimum version 3.1 to ensure (publish, subscribe and QoS). The respective connection data to the broker (IP and port) is provided on request. The communication with the broker should use the MQTT protocol.

The machine is authenticated via username and password. It must be possible to change any connection data in the machine settings. These settings should be available in the HMI under higher administrative permissions.

4.2 Quality of service level (QoS)

MQTT provides three levels of QoS:

QoS 0(At most once): Can be used for less relevant data that is retransmitted with each machine cycle. These are, for example, (error / good) counters of individual stations.

QoS 1(At least once): Must be used for all data relating to parts and the system status. This includes, for example, measurement data, data on parts placement, operating mode of the machine, machine faults, etc. Furthermore, all information that triggers a reaction from the MES system (Triggers) must be published with QoS 1.

QoS 2(Exactly once): Not currently used, if this is necessary, it must be clarified with HA.

4.3 Topic naming structure

The topic names consist of minimum 4 layers which are:

1. **Prefix:** namespace for the MQTT broker (e.g. rw, tm).
   For machines that require cross-communication, the prefix of all cross-communication topics must consist the broker namespace "_machineCommunication" (e.g., rw_machineCommunication, tm_machineCommunication).
2. **Cluster:** Used for connected machines that are part of one group. If there is no machine grouping use "standaloneMachines" to match the topic-layers
3. **Machine Identifier:** HA machine identification
4. **Internal Identifier:** Path to the topic
5. **Topic name:** name of the topic

The Prefix and the Machine Identifier should be available for modifications in the HMI under higher administrative permissions. For Internal identifier and topic name check Section 6.

Example of a topic that provides information about the cycle time of a station:

Prefix	rw
Cluster	standaloneMachines
MachineIdentifier	H743
InternalIdentifier	station01
TopicName	cycleTime
Full Topic structure	rw/standaloneMachines/H743/station01/cycleTime

Examples for cross communication topics:
- rw_machineCommunication/standaloneMachines/H123/station01/…
- rw_machineCommunication/segment01/H123/station01/…
- tm_machineCommunication/standaloneMachines/H456/station01/…

If the machine does not have stations, the tag is located below the resource itself.

Examples for MES relevant topics:
- rw/standaloneMachines/H123/cycleTime
- rw/standaloneMachines/H410/station01/cycleTime
- rw/standaloneMachines/H410/station01/nest01/mesaurement/protectiveShroud
- rw/standaloneMachines/H410/station01/nest01/quality/partPosition
- rw/standaloneMachines/H410/station12/nest01/maintenance/pressingSpring/cylceCounter

4.4 Last will message

When the connection is established, a **Last Will Message** is to be transmitted from the machine to the broker in which the machine status is displayed as “disconnected”.

The topic used should be **TriggerActive,** and here we should also see **the other states of the machine** which are mentioned under **section 2.2** in the following priority:

1. **connected:** Machine waiting for Trigger
2. **disabledTrigger:** Machine sends signals to MQTT with QoS but ignores Triggers
3. **disabledCommunication:** Machine works without MQTT broker
4. **disconnected:** Last will message

4.5 Topic data

The data content of the published topics should be in **JSON format**. The top object level should contain a **UTC timestamp** with the key “ts”. The timestamp is **generated when the object is created** and should correspond to the **ISO 8601** format (e.g. “2025-04-24T09:31:03.885Z”). The other two object levels should be the **data type of the value and the value itself**.

Examples:

{"ts":"2025-04-24T09:31:03.885Z","type":"number","unit": "s","value": 1.586}
{"ts":"2025-04-24T09:45:53.799Z","type":"object","value":{"active":true,"code":300501}}
{"ts":"2025-04-24T09:23:00.424Z","type":"string","value":"Automatic"}
{"ts":"2025-04-24T10:02:45.852Z","type":"string","value":true}
{"ts":"2025-12-03T11:51:16.407000000Z","type":"object","value":
{"serialnumber":94128,
"unit":"mm",
"value":-0.045,
"lowerLimit":-0.300,
"upperLimit":0.300}}

5 OPC UA

5.1 Network interface

HA does not host the OPC UA server, this should be hosted by the contractor locally on the machine while **HA will be a client**.

5.2 OPC UA server security and connections

The OPC UA server should be set up to make multi-client sessions possible. There will be a maximum of 5 connections at the same time. However, there should be some leeway in case additional connections need to be made later.

Unsecured connections are not allowed. The **minimum-security** requirement for any connection to the OPC UA server is an **authorization with username and password**.

5.3 OPC UA Node ID naming methodology

The node ID should consist of several components that together form a unique and descriptive identifier. The use of a TagName is mandatory. The structure can be defined as:

1. **Namespace:** A prefix that denotes the application or system namespace.
2. **SourceFolder:** The primary folder or module from which the tag originates.
3. **SubFolder:** Any sub-categories or sub-modules within the primary folder.
4. **TagName:** The specific tag or variable name.

Example of a tag that provides information on the cycletime of a station:

Namespace	ns=2
Source Folder	MES
SubFolder	Machine
Tag Name	CycleTime
Node ID	ns=2; s=MES.station01.CycleTime

6 MES capability level 1

Note: The levels are built on each other. If a level of 2 is specified, all information from level 1 is also required. Depending on the project requirements, additional data points / triggers can be requested during the project review.

The supplier must provide a translation table for all error codes with their description and meaning (ex: 1234 = station 1 error 234).

The internal identifier/subfolder follows the prefix, cluster, and machine identifier as described in 4.3. This is followed by the identifier described in column 2 of the following tables and then the respective topic.

6.1 Basic machine information

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
machineMode	Machine	Automatic/Manual (defined by supplier)	string
operationMode	Machine	Idle/SetUp/StartUp/Productive/Interruption/TearDown	string
totalPartsOK	Machine	Total produced OK parts of lifetime	integer
totalPartsNOK	Machine	Total produced NOK parts of lifetime	integer
totalParts	Machine	Total produced parts of lifetime	integer
totalPartsQuality	Machine	OPTIONAL: Total Quality parts of lifetime	integer
totalPartsStartUp	Machine	OPTIONAL: Start up parts of lifetime	integer
totalHours	Machine	Total hours of lifetime	numeric
softwareVersion	Machine	Actual installed software version on the machine, should be created automatically CRC	string:x.x.x
mesVersion	Machine	Version of this document as string	string
capabilityLevel	Machine	Maximum capability level of the machine	string
errorID	Machine	Machine ErrorID (breakdown reason)	integer
errorDescription	Machine	OPTIONAL: if no clean error text can be provided , a table with ID translation is needed.	string
connectionState	Machine	Check section 4.4	string
waitingForMES	Machine	Must be true while machine is waiting for a trigger, false otherwise	boolean
article	Machine	Current article set on the machine (ex. 908-123-545)	string
orderNumber	Machine	Current order set on machine	string
orderTarget	Machine	The quantity to be produced for the current order, may be adjusted during the order by HA	integer
cycleTime	Machine	The time between two parts (every time a part leaves the machine)	numeric
cycleTimeTarget	Machine	Target time between two parts produced	numeric
recipe	Machine	Ex: measurement limits, additional explanations below	string

Recipe: Any value that can be changed by HA on the machine which is related to the article/recipe/machine setting must be visible in the data points. Since this is heavily dependent on the design of the machine, the supplier must define the structure of the data.

6.2 Multiple stations

In case the machine has multiple stations, we need the following information from each station.

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
totalPartsOK	station[X]	Total OK parts of station X	integer
totalPartsNOK	station[X]	Total NOK parts of station X	integer
totalPartsQuality	station[X]	OPTIONAL: Total Quality parts of station X	integer
totalPartsStartUp	station[X]	OPTIONAL: Start up parts of station X	integer
totalParts	station[X]	Total produced parts of station X	integer
active	station[X]	Whether the station is active or not	string
identifier	station[X]	Identifier of the station, X can also be used	string
cycleTime	station[X]	Must only account for the station's processing time	numeric
materialOverConsumption	station[X]	If there is **excess** material consumption (e.g. assembly retry), this is reported here. Topic must include material number and quantity.	object
task	station[X]	Description on what the station does ex. Seal testing	string
result	station[X]/quality/nest[X]	The result from the part at station X in nest Y Accepted option: OK/NOK	string
errorCode	station[X]/quality/nest[X]	The error code in case of NOK (scrap reason), If error code is not possible, all the test steps of the station must be shown or clear Error description provided.	string
<Name>	station[X]/quality/nest[X]	Quality evaluation of the test step. <TestStep> should be a descriptive name for the test step.	object
<Name>	station[X]/measurement]/nest[X]	If available, measured values of the <TestStep> e.g.	object
<Name>	station[X]/data/nest[X]	If available, additional data of the <TestStep> e.g.	object

[X] Placeholder for a two-digit number with optional subdivisions separated by '_' (e.g., 01, 02_01, 03_01_02 etc.)
For all topics under the quality layer the serial number should be included if we have multiple stations
<Name> means that the topic name can be freely chosen; the name must be chosen according to the information it contains.

6.3 Error proof part test (EPP-test)

In case the machine needs to make a dummy test for production release.

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
<Name>	station[X]/eppTestResult/nest[X]	The result and measurement for the EPP test.	object

Exemple:
{
"ts": "2026-04-08T09:20:51.042000000Z",
"type": "object",
"value": {
"productCarrierNumber": 10,
"limitType": "upper",
"result": "OK",
"unit": "mm",
"contacts": {
"contact1": 0.475,
"contact2": 0.470,
"contact3": 0.465,
"contact4": 0.480
}
}
}			

6.4 Packing unit

This section describes the output packaging units used by the machine (e.g., blisters, boxes, etc.).

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
packingUnitFinished	station[X]/packingUnit[X]	Signal when the packing unit is finished	boolean
packingUnitParts	station[X]/packingUnit[X]	Live number of parts in the packingUnit	integer
packingUnitIdentifier	station[X]/packingUnit[X]	Identifier of packingUnit (if available)	string
packingUnitDescription	station[X]/packingUnit[X]	Description, e.g., “Blister”, “Left Box”, “Bag”	string

6.5 HMI

Provides information about the currently active user lvl on the machine

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
activeUserLevel	Machine/hmi[X]	Access level of the user logged in to the HMI	string
activeUserName	Machine/hmi[X]	Name of the user logged in to the HMI	string

7 MES capability level 2

HA uses **triggers** to start certain events or behaviors within a machine.

1. **For MQTT:** this will be handled by using defined topics where the supplier must subscribe and make decisions based on the value that HA publishes in the topic

2. **For OPC UA:** this will be handled by using defined methods where the supplier must make decisions based on the value the methods are called with by HA

Name	Input	Description	Prefered data type
automaticRun	true/false	**AutomaticRun** is controlling the machine, any **OperationMode besides Idle** must be completed with **AutomaticRun = true**. If **AutomaticRun = false**, the machine is **not allowed to start** its processes, if the machine is already in the middle of a cycle and AutomaticRun is set to false by HA, the machine **must do a soft stop** (complete the current cycle and stop)	**boolean**
orderArticle	Article	When the machine has multiple recipes, the correct recipe must be selected **automatically** based on Article value.	string
orderTarget	Target	**The quantity is a dynamic** value that can be **changed** during the current order, the supplier **must prepare all the necessary calculations** to **stop the feed in of the raw material based on the order quantity** and the number of stations on the machine at the beginning of every machine cycle.	**integer**
orderNumber	Order	The order is just visual data.	string

MQTT source identification: In case of a bidirectional MQTT communication, a "source" variable inside the message value is needed to correctly identify from where the information will be published from.
Example: {"ts": "2026-04-03T14:07:35.394618Z", "value": true, "type": "boolean", **"source": "Equipment name"}**

8 MES capability level 3

8.1 Machine information

If level 3 is required, this are information we needed additionally to the data mentioned in level 1

MQTT -Topic name OPCUA – Tag name	MQTT – Internal Identifier OPCUA – SubFolder	Description	Prefered data type
preparedIdentifier	parts	The scanned part identifier (barcode,DMC)	string
preparedPart	parts	Trigger for HA to read the preparedIdentifier data, if timeout is reached the value must be reset	boolean
preparedSerialNumber	parts	Optional: (only need for machine that have multiple stations) Unique number per part that will increment on each part (helps to trace parts without dmc for saving the results, pictures etc)	integer
finishedIdentifier	parts	The part from the station where **finishedAcknowledge** trigger is happening	string
finishedQuality	parts	Information whether the part from the finishedIdentifier is OK or NOK from overall quality perspective without MES influence.	string
finishedErrorCode	parts	Optional:(Only if requested) The error code in case of NOK, and a separate document where all error codes and error text are mapped and described. (ex. error code = “1234” ; error text = “Electrical test failed, open circuit”)	string
finishedPart	parts	Trigger for HA to read the finishedIdentifier data (value set to true), if timeout is reached or the ACK is received the value must be reset. (reset to false)	boolean
finishedSerialNumber	parts	Optional: (only need for machine that have multiple stations) Unique number per part that will increment on each part (helps to trace parts without dmc for saving the results, pictures etc)	integer

8.2 Triggers

As a standard, the first trigger (preparedAcknowledge) is used to **acknowledge the scanned Part identifier (e.g. Barcode, DMC)**, the second one (finishedAcknowledge) is used to **acknowledge the safe storage of data from HA**. The exact behavior of the traceability level 3 triggers **needs to be discussed with HA to avoid cycle time increases** as this is heavily dependent on how the machine is being built up. After a successful **acknowledge** from HA, selected signals may need to be cleared or reset (e.g., boolean TRUE to FALSE, emptying string values, or resetting variables to their default state) to avoid carrying over data to the next cycle, should to be discussed with HA in the pre-acceptance.

Name	Input	Description	Prefered data type
preparedAcknowledge	Prepared Identifier Response	The machine **must wait** for a response. If no response happens, a **timeout must trigger** after 10 seconds. If the **preparedIdentifier input is not the expected one, an error must be shown.** **1.** If called true, the machine is allowed to work further **2.** If called false the part must be removed and the scanning process restarted on a different part	boolean
finishedAcknowledge	Finished identifier Response	The machine **must wait** for a response. If no response happens, a **timeout must trigger** after 10 seconds. If the **finishedIdentifier input is not the expected one, an error must be shown.** **1.** If called true, the machine is allowed to work further **2.** If called false, the part must be scrapped	boolean