IPC-HERMES-9852
The global standard for "M2M" in SMT assembly

IPC-HERMES-9852

The Global Standard for Machine-to-Machine Communication in SMT Assembly

Version 1.0.4

Contributing companies:

Achat Engineering GmbH
ASM AS GmbH
ASYS Automatisierungssysteme GmbH
BESI
BTU
cts
CTI Systems
CYBEROPTICS
ECD
ERSA GmbH
EUNIL
eXelsius
FLEXLINK
GKG
GÖPEL electronic GmbH
HANWHA
Heller Industries
IPTE
ITW EAE
JAPAN UNIX Co. Ldt.
Keysight
KIC
KOH YOUNG Technology Inc.
Kulicke & Soffa
MAGIC RAY Technology
MIRTEC
MYCRONIC
Nordson ASYMTEK & MATRIX
Nutek Europe B.V.
OMRON Corporation
OSAI
PARMI
Pemtron
Rehm Thermal Systems GmbH
RG Elektrotechnologie
SAKI Corp
SEHO
SEICA Automation
Shenzhen Rejoint Automachine Equip. Co. Ltd.
SICK
SMT Thermal Discoveries
Solderstar
SONIC Technology
SPEA S.p.A.
Test Research, Inc. (TRI)
UIC
VISCOM AG
ViTrox
YJ Link Co., Ltd.
YXLON

Contents:

1 Scope

The aim of this specification is to create a state-of-the-art communication protocol for surface-mount technology (SMT) production lines. Therefore, this new communication protocol has to cope with the following:

- Replace the electrical SMEMA interface as specified in [IPC_SMEMA_9851]
- Extend the interface to communicate (including):
    - Unique identifiers (IDs) for the handled printed boards
    - Equipment identifiers of the first machine noticing a printed board
    - Barcodes
    - Conveyor speed
    - Product type specific information (including):
        - Product type identifier
        - Length
        - Width
        - Thickness

With respect to version numbers IPC-HERMES-9852 adheres to the rules of Semantic Versioning 2.0.0 [SemVer_2.0.0].

Hints on naming:

- Wherever a feature is described by the word "**shall**", it is mandatory.
- The word "machine" is used for any equipment which can be found in a SMT production line (e.g., printers, placement machines, ovens, automatical optical inspections (AOIs), transport modules, shuttles, stackers, etc.).
- The term "printed board" may also refer to carriers transporting printed boards.
- The word "Hermes" is used as abbreviation for "IPC-HERMES-9852".
- "The Hermes Standard" and IPC-HERMES-9852 are synonyms for the standard specified in this document and might be used interchangeably.

2 Technical Concept

2.1 Prerequisites

This specification is based on the prerequisite, that any application implementing this protocol has to provide connectivity based on Internet Protocol (IP) [IETF_RFC_791] / [IETF_RFC_2460] via Transmission Control Protocol (TCP) [IETF_RFC_793] (International Organization for Standardization / Open System Interconnection (ISO / OSI) model [ISO_7498-1] layer 3) to the adjacent machines.

2.2 Board IDs

Board individuals are identified by board IDs. These must be globally unique identifiers (GUIDs) according to [ITU-T_REC_X.667] (e.g., 123e4567-e89b-12d3-a456-426655440000). They are generated by the first machine in a consecutive row of machines implementing the Hermes protocol. The board ID is passed from machine to machine. If a machine in a line does not implement the Hermes protocol, the board ID is lost and a new one will be generated by the next machine implementing Hermes. This is shown in Figure 1.

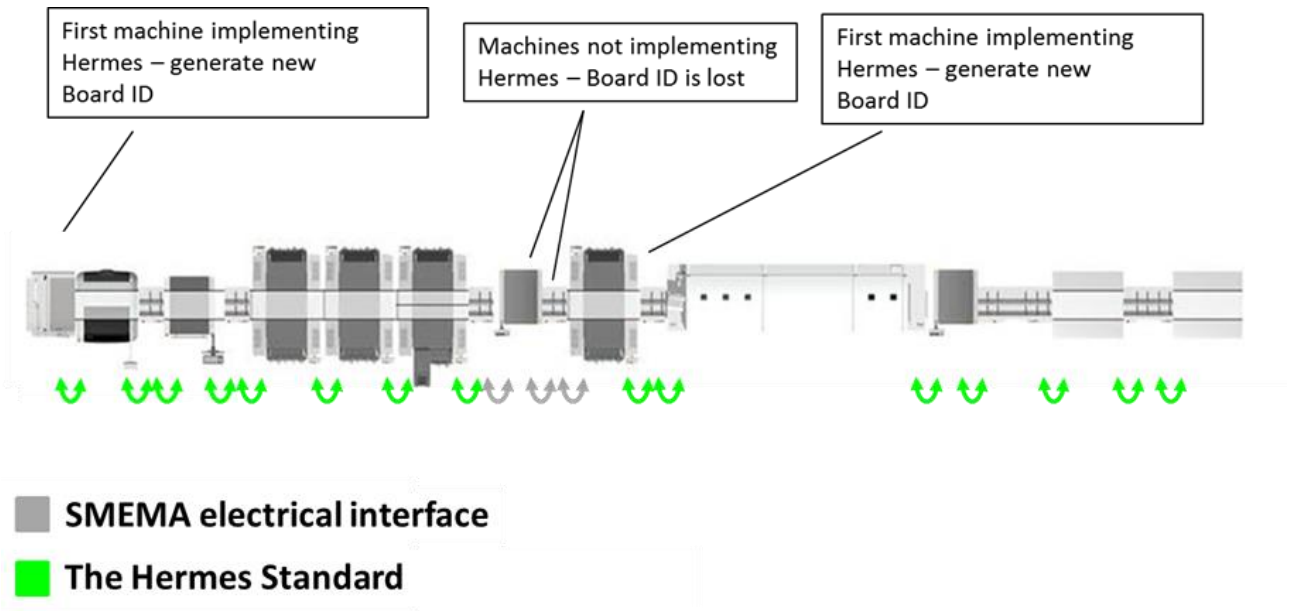

Figure 1 Generation of Board IDs

2.3 Machine-to-Machine Communication (Horizontal Channel)

2.3.1 Topology

Any machine in a line offers one TCP server per lane on its downstream side. Further servers per lane might also be necessary (e.g., if reverse transportation is supported). The TCP port number is not specified but can be configured by the user. The recommended port numbers are 50100 plus lane identifier (ID) with lanes being enumerated looking downstream from right to left beginning with 1 (i.e., for the left lane of a dual lane machine, the upstream machine server accepts connections on port 50102).
The downstream machine opens one connection for every lane on its upstream side to the upstream machine(s). So every printed board handover point corresponds to one TCP connection per exchange direction as shown in Figure 2.

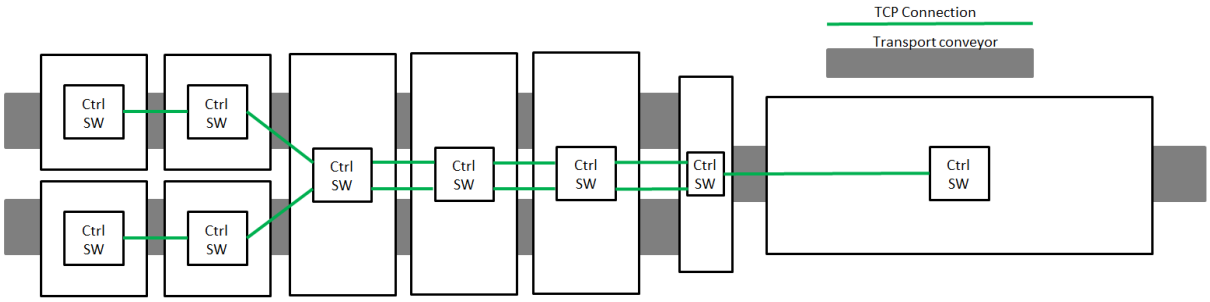

Figure 2 TCP Connections in a Line

2.3.2 Connecting, Handshake and Detection of Connection Loss

After booting, the downstream machine starts cyclic connection attempts to the configured upstream machines. When a connection is established, the downstream machine starts sending a ServiceDescription message whereupon the upstream machine answers with its own ServiceDescription. This ServiceDescription message contains the lane ID of the sending machine related to this TCP connection. It also contains the implemented version and a list of all optional features and additional features of a higher version which are implemented by the machine. The features of the Hermes specification version 1.0 have to be supported by any implementation. See Figure 3.
If a downstream machine is already connected to the lane, this connection will be retained. A Notification message **shall** be sent to the new connection before it is closed.
After exchanging the handshake messages, both machines may begin to send BoardAvailable / MachineReady messages (see 2.3.3).

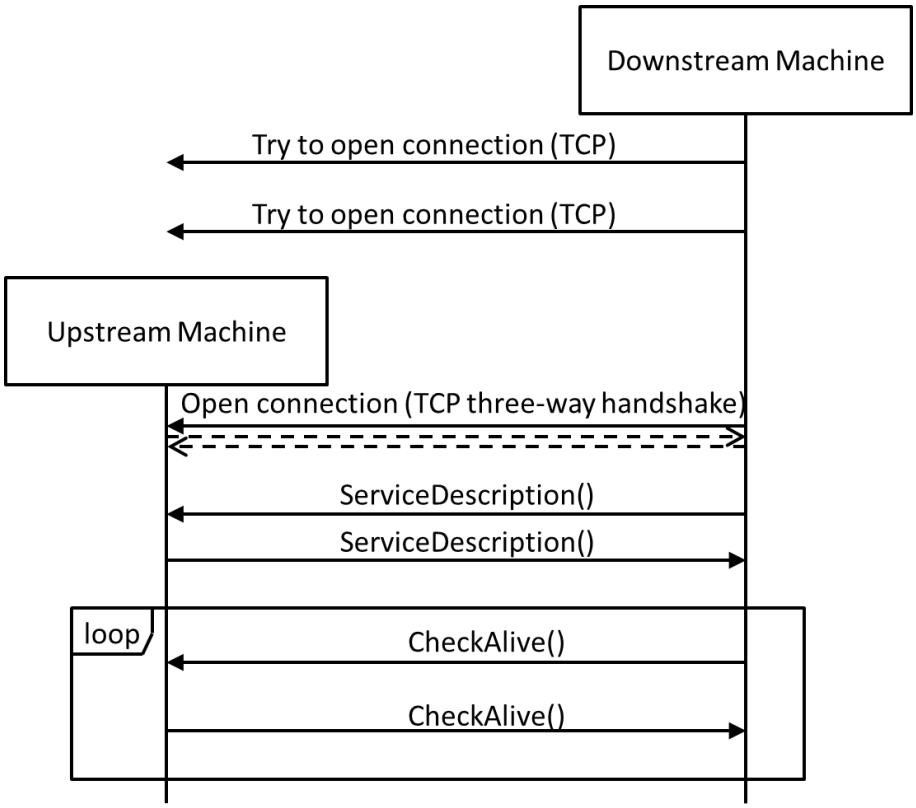

Figure 3 Connection, Handshake and Connection Loss Detection on Horizontal Channel

The connections are kept open all the time. As TCP by itself does not detect connection losses (“half-open connections” caused by, for example, process / computer crash, unplugged network cables, etc.) both sides of a connection have to send cyclic CheckAlive messages. Those messages do not have to be answered by the remote side – the TCP stack will detect a connection loss when trying to send the packet. If the server detects a connection loss, it cleans up the connection and waits for a new connection by the client. If the client detects a connection loss, it cleans up the connection and restarts with cyclic connection attempts.

2.3.3 Normal Operation

When an upstream machine has a printed board available for handover, it sends a BoardAvailable message while a downstream machine ready to accept a printed board sends a MachineReady message. The naming of these messages is inspired by the electrical SMEMA interface. However, the messages do not represent the state of a machine’s interface directly but are events for initiating a printed board handover.

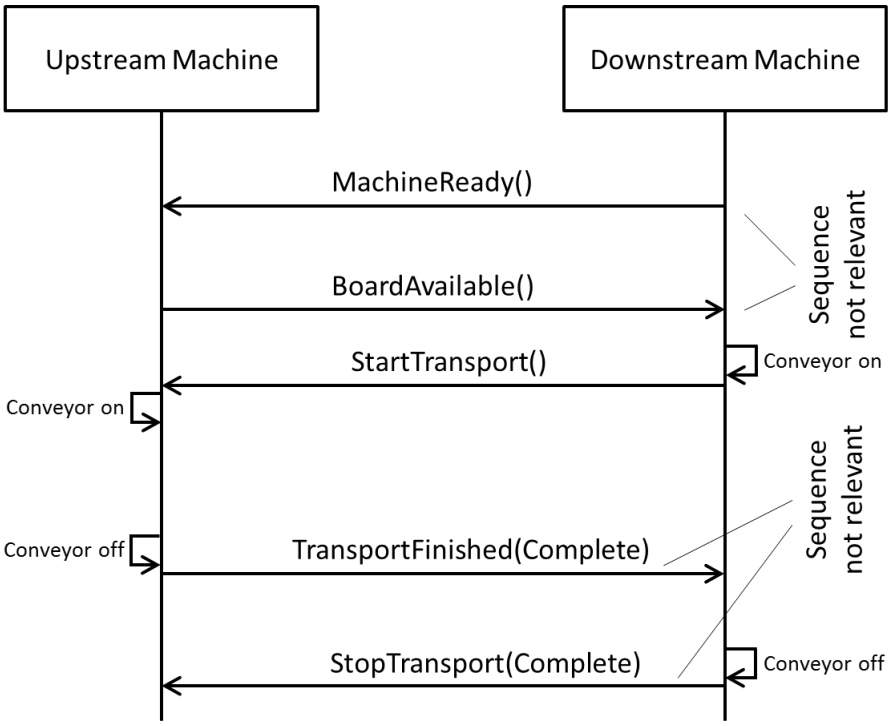

Figure 4 Communication Sequence for Board Transport

When both machines have indicated their readiness to handover the printed board, the downstream machine initiates the transfer by switching on its conveyor and sending the StartTransport message. Upon receiving this message, the upstream machine switches on its conveyor and the printed board moves into the downstream machine.
When the upstream machine is able to state that the printed board has fully left the machine, it sends the TransportFinished message. When the downstream machine has full control of the board, it sends the StopTransport message. The handover of a printed board is finished and is ready to start over.
If the upstream machine receives a StopTranport message and has not sent the TransportFinished message yet, it has to stop its conveyor and send the TransportFinished message.
The MachineReady message does not trigger an action on one of the machines directly. However it still is necessary to realize machines such as shuttles which have to react to the availability of their downstream machines.
See Figure 4.

2.3.4 Transport Error Handling

To keep this protocol hardware independent, the handling of transport errors is described based on a very simple model of the board handover. The handover process is structured into the three phases:

1) NotStarted: The board is fully inside the upstream machine.
2) Incomplete: The board is partly inside both machines.
3) Complete: The board is fully inside the downstream machine.

Any state or event which prevents one or both machines from handing over a printed board is interpreted as an error. An error may be detected by any of the machines in any of the three handover phases. It is up to the

application how to detect the current handover phase, how to detect errors and how to solve them eventually (e.g., sensors, model based prediction, timeouts, user interaction, etc.).
The following sequence charts give an overview of the communication within this protocol depending on the machine which detects the error and the phase in which it is detected. The point in the sequence where the error is detected is marked by the following symbol: Stop

Scenario U1a

- Error detected by the upstream machine
- Printed board fully inside the upstream machine
- Error detected before StartTransport has been received

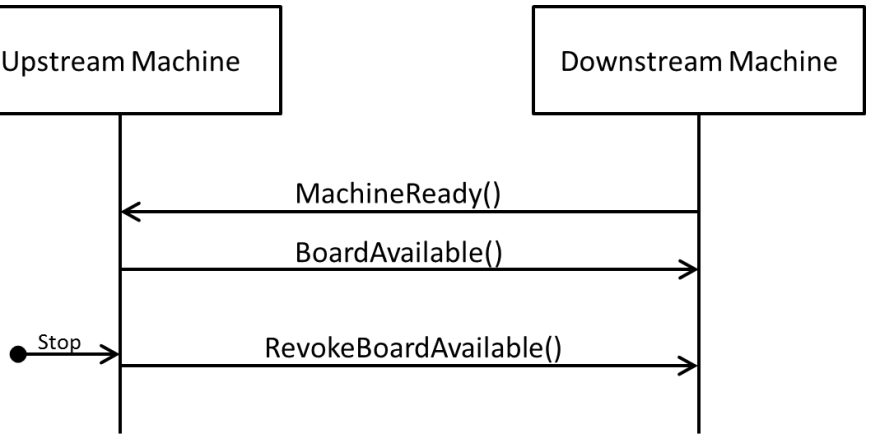

Figure 5 Communication Sequence in Scenario U1a

Error detection: The error is detected before any transport started.
Reaction on upstream machine: The upstream machine sends a RevokeBoardAvailable message.
Reaction on downstream machine: None.
Resolution: After the error is solved, the regular transport sequence can start from the beginning.

See Figure 5.

Scenario U1b

- Error detected by the upstream machine
- Printed board fully inside the upstream machine
- Error detected after StartTransport has been received

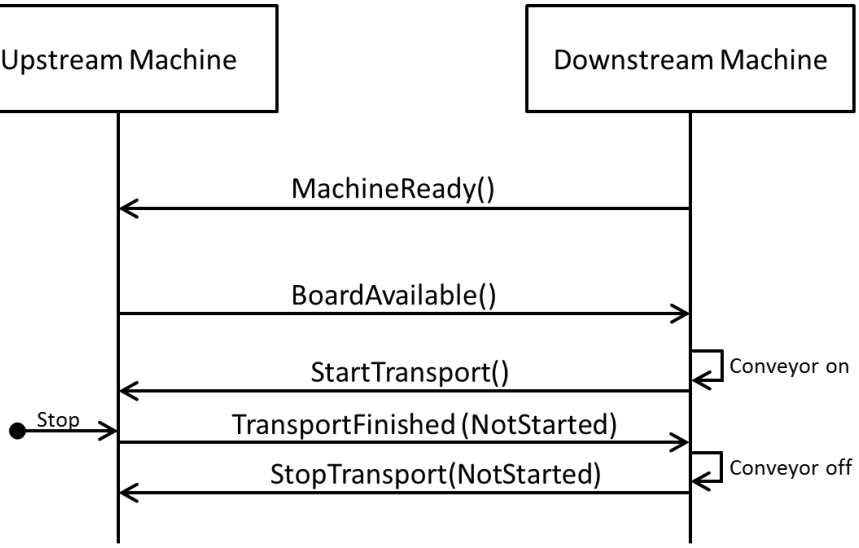

Figure 6 Communication Sequence in Scenario U1b

Error detection: The error is detected after the downstream machine started its conveyor and has sent the StartTransport message.
Reaction on upstream machine: The upstream machine sends a TransportFinished message indicating that it has not started the transport.
Reaction on downstream machine: Upon the TransportFinished message, the downstream machine stops its conveyor and sends a StopTransport message indicating that no transport has started.
Resolution: After the error is solved, the regular transport sequence can start from the beginning.

See Figure 6.

Scenario U2

- Error detected by the upstream machine
- Printed board partly inside both machines

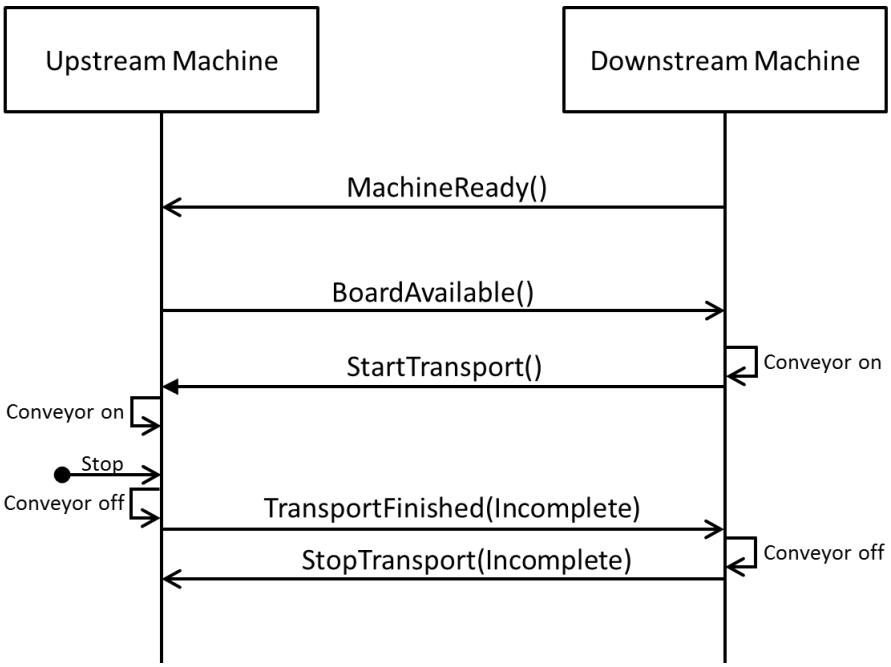

Figure 7 Communication Sequence in Scenario U2

Error detection: The error is detected after both machines started their conveyors. The upstream machine assumes that the printed board may have partly entered the downstream machine.
Reaction on upstream machine: The upstream machine sends a TransportFinished message indicating that the printed board might be located between the machines.
Reaction on downstream machine: Upon the TransportFinished message, the downstream machine stops its conveyor and sends a StopTransport message indicating the state of the printed board handover. Note that in Figure 7 the StopTransport message is represented with parameter “Incomplete”. However in this scenario, the downstream machine could send any of the allowed transport states.
Resolution: After the error is solved, the regular transport sequence can start from the beginning. The regular transport message sequence also applies to a printed board located between the two machines.

See Figure 7.

Scenario U3

- Error detected by the upstream machine
- Printed board fully inside the downstream machine

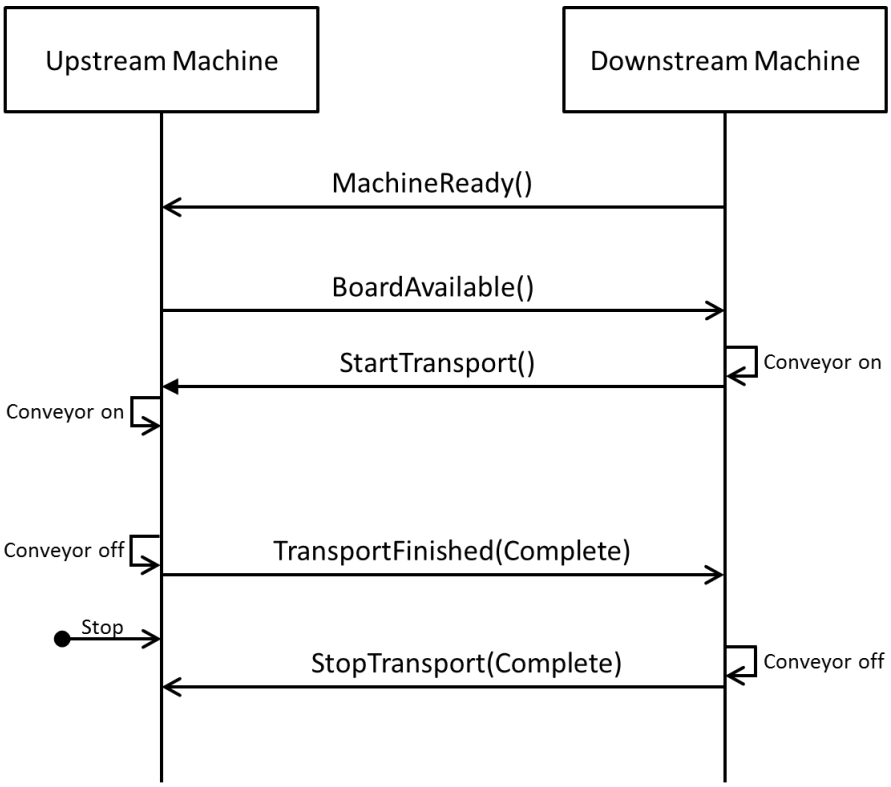

Figure 8 Communication Sequence in Scenario U3

Error detection: The error is detected after the printed board is fully inside the downstream machine.
Reaction on upstream machine: None. Although the machine detected an error, it is irrelevant for the handover process.
Reaction on downstream machine: None. The downstream machine is not aware of any error.
Resolution: This scenario is irrelevant for the Hermes protocol. It is just listed for completeness.

See Figure 8.

Scenario D1

- Error detected by the downstream machine
- Printed board fully inside the upstream machine
- Error detected before StartTransport has been sent

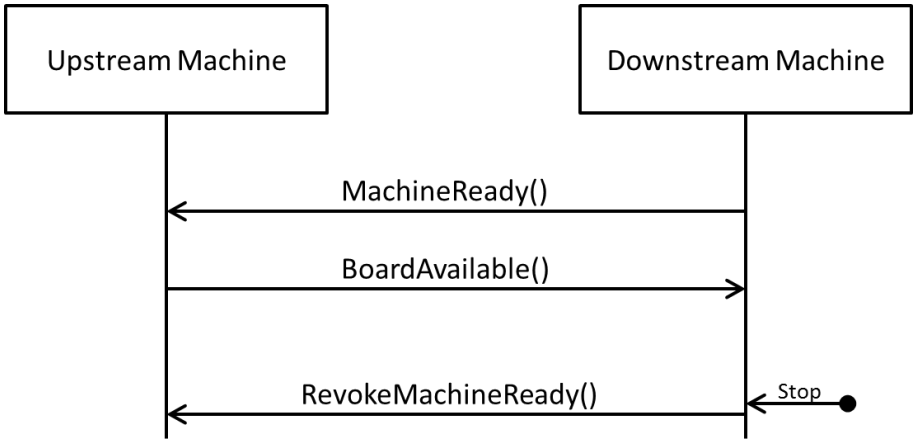

Figure 9 Communication Sequence in Scenario D1

Error detection: The error is detected before any transport started.
Reaction on upstream machine: None.
Reaction on downstream machine: The downstream machine sends a RevokeMachineReady message.
Resolution: After the error is solved, the regular transport sequence can start from the beginning.

See Figure 9.

Scenario D2

- Error detected by the downstream machine
- Printed board partly inside both machines

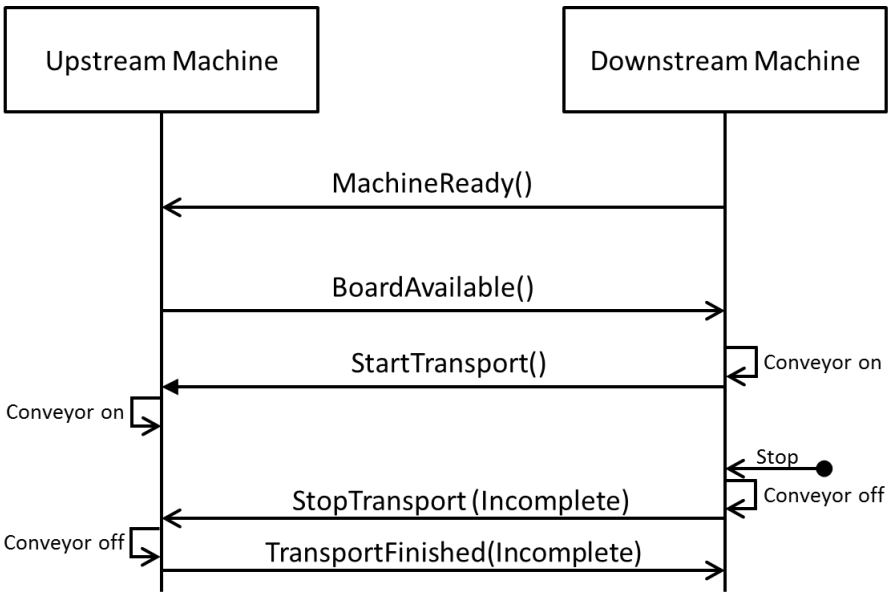

Figure 10 Communication Sequence in Scenario D2

Error detection: The error is detected after both machines started their conveyors. The downstream machine assumes that the printed board may already has entered its conveyor.
Reaction on upstream machine: Upon the StopTransport message from the downstream machine, the upstream machine stops its conveyor and sends a TransportFinished message indicating the state of the printed board handover. Note that in Figure 10 the TransportFinished message is represented with parameter “Incomplete”. However in this scenario, the upstream machine could send any of the allowed transport states.
Reaction on downstream machine: The downstream machine stops its conveyor and notifies the upstream machine of the error by sending a StopTransport message indicating an incomplete printed board handover.
Resolution: After the error is solved, the regular transport sequence can start from the beginning. The regular transport message sequence also applies for a printed board located between the two machines.

See Figure 10.

Scenario D3

- Error detected by the downstream machine
- Printed board fully inside the downstream machine

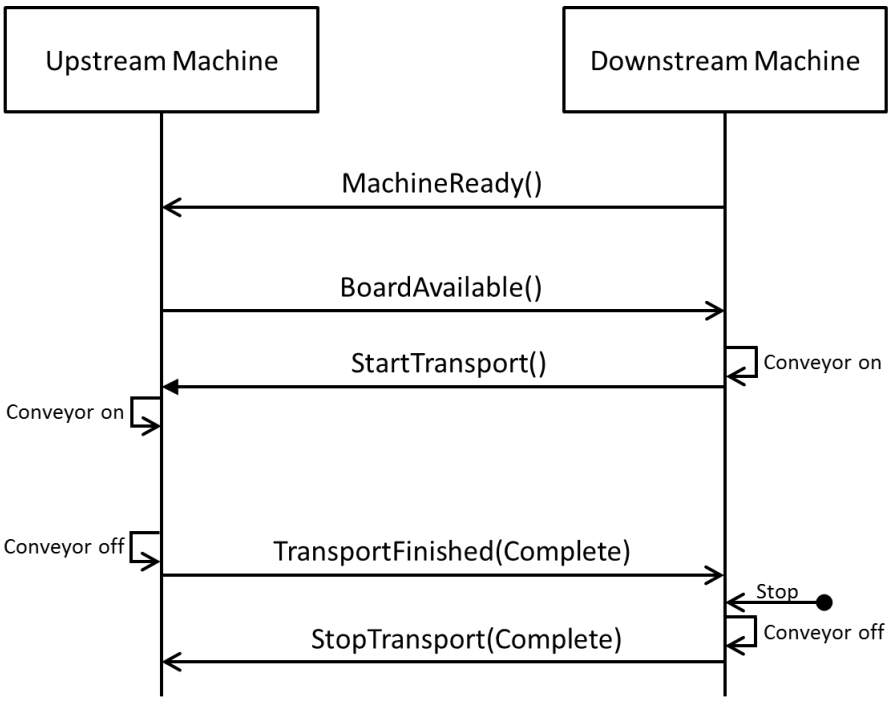

Figure 11 Communication Sequence in Scenario D3

Error detection: The error is detected after the printed board is fully inside the downstream machine.
Reaction on upstream machine: None. The upstream machine is not aware of any error.
Reaction on downstream machine: None (at least in the scope of this protocol).
Resolution: This scenario is irrelevant for the Hermes protocol. As transport sequences are always initiated by the downstream machine sending StartTransport, troubleshooting (possibly including running the conveyor of the downstream machine) can be executed independently from the upstream machine.

See Figure 11.

2.3.5 Protocol States and Protocol Error Handling

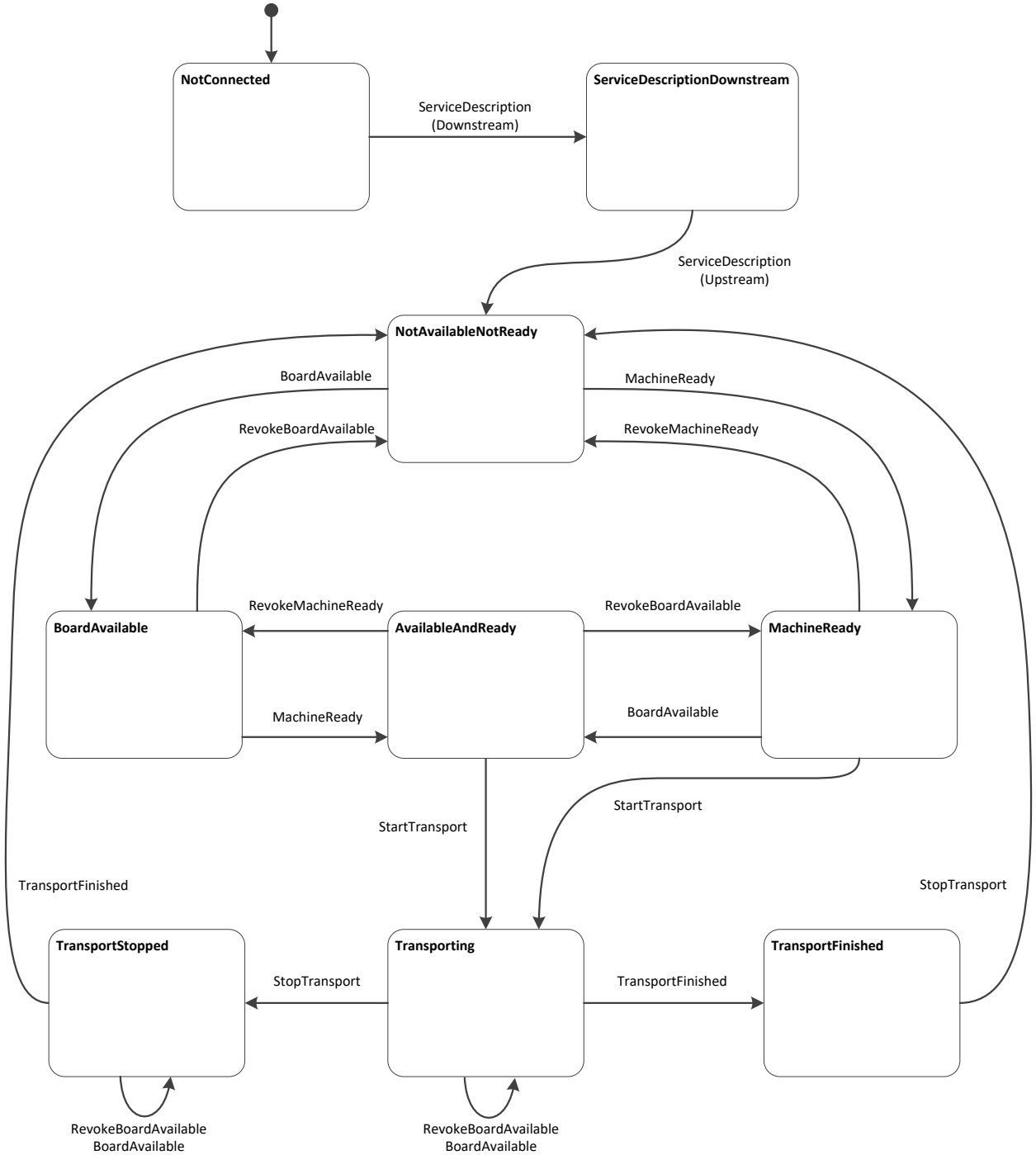

Figure 12 Hermes Interface States on Horizontal Channel

Figure 12 lists all states and transitions of a Hermes interface corresponding to the machine-to-machine (M2M) communication. The state is the comprehensive state of the interface rather than the state of one of the involved machines.

The messages may only be sent if they trigger the corresponding transition shown in the state chart. Any message defined in this standard, except “Notification” and “CheckAlive”, which is received not triggering a transition is interpreted as a protocol error (i.e., a MachineReady message when the interface is in the state Transporting). In case of a protocol error, any running transport **shall** be stopped and the connection is terminated. The interface may start over with a new connection. Any unknown message, which is received, **shall** be ignored and discarded to keep upward compatibility.
Note that due to race conditions, a RevokeBoardAvailable message may overlap with a StartTransport message or even a StopTransport message, so this **shall** not be treated as a protocol error (transition from MachineReady to Transporting and self-transitions on Transporting and TransportStopped).

2.4 Remote Configuration

2.4.1 Topology

Although a machine may offer the possibility to configure the Hermes TCP port(s) and the IP address(es) of its upstream machine(s) locally (i.e., via a graphical user interface of the machine controller), every machine implementing this protocol **shall** offer a possibility to configure these properties remote via TCP. Therefore, the machine **shall** offer a TCP server on port 1248 on at least one network adapter where it accepts configuration messages (see 3.13 to 3.15 for detailed information).
The configuration system opens a connection to each required machine. The connection **shall** only be kept open as long as needed and closed by the configuration system.

2.4.2 Remote Configuration

A SetConfiguration message **shall** contain the full configuration for all Hermes interfaces of a machine. Any existing configuration is overwritten when a SetConfiguration message is received. Whenever a configuration is not applicable (e.g., bad IP address format), the SetConfiguration message is answered with a Notification message (see 3.5). Every time the configuration is changed, affected open Hermes connections will be reset at the next appropriate moment.
It is possible to read the current configuration through the GetConfiguration message answered by a CurrentConfiguration message. The configuration **shall** be persisted until it is changed.

3 Message Definition

3.1 Message Format

Messages use the Extensible Markup Language (XML) format, where at least version 1.1 of XML **shall** be supported [W3C_XML_1.1].
For character encoding UTF-8 has to be used (No other encoding may be specified in the XML declaration).
In the following sections of this document, for a better readable description of the XML data structures, tables are used instead of commonly used schema definitions.
Maximum size for every message is 64 kByte (i.e., 65536 bytes). For every string parameter there is either a fixed or minimum size that must be supported (individual values see tables).
In the tables, XML attributes are marked with the image "  " and XML child nodes are marked with the image "  ", which in turn may consist of more XML structures.
The representation of data types (e.g., floating point numbers, boolean attribute, etc.) **shall** comply with the W3C XML schema recommendation [W3C_XML_Schema].
To keep upward compatibility, any message or attribute unknown by an implementation can be ignored and discarded.

3.2 Root Element

Every message is enveloped by a common root element with tag <Hermes>. The root element optionally includes a timestamp attribute with the following format (based on the W3C note "Date and Time Formats" [W3C_DATE_TIME]):

```
YYYY-MM-DDThh:mm:ss.s
```

where:

```
YYYY = four-digit year
MM   = two-digit month (01=January, etc.)
DD   = two-digit day of month (01 through 31)
hh   = two digits of hour (00 through 23) (am / pm not allowed)
mm   = two digits of minute (00 through 59)
ss   = two digits of second (00 through 59)
s    = one or more digits representing a decimal fraction of a second
```

The decimal fraction of the second **shall** be given with three digit precision.
The timestamp is optional and intended for diagnostic purposes only.

An example for a CheckAlive message would be:

```
<Hermes Timestamp="2017-07-16T19:20:30.452">
  <CheckAlive />
</Hermes>
```

A machine is not required to emit a precise timestamp, since this attribute is intended mainly for debugging purposes.

Recommendation: Synchronize all machines in a line to a common time source. For machines that do not have an absolute time source, the year should be set to "0000". At any rate, the timestamp should be monotonic.

3.3 CheckAlive

The CheckAlive message is used to detect connection losses. It therefore does not have to transport data and can be ignored by the receiver. Accordingly there is no response.

CheckAlive	Type	Range / Multiplicity	Optional	Description

3.4 ServiceDescription

The ServiceDescription message is sent by both machines after a connection is established. The downstream machine sends its ServiceDescription first whereupon the upstream machine answers by sending its own ServiceDescription.

ServiceDescription	Type	Range / Multiplicity	Optional	Description
MachineId	string	any string (minimum supported length: 80 bytes)	no	ID / name of the sending machine for identifying it in a Hermes enabled production line.
LaneId	int	1 .. n	no	The sending machine's lane of this connection relates to. Lanes are enumerated looking downstream from right to left beginning with 1.
Version	string	xxx.yyy (7 bytes)	no	The implemented interface version of the machine.
SupportedFeatures	Feature []	1	no	List of supported features (empty for version 1.0).

xxx.yyy must match the regular expression
`[1-9][0-9]{0,2}\.[0-9]{1,3}`

The features specified in version 1.0 of this protocol have to be provided by any implementation and thus are not listed in the SupportedFeatures list of the ServiceDescription explicitly. The same applies for all mandatory features of the version specified in the Version attribute. All optional features or additional features of a higher version supported by a machine need to be listed in the SupportedFeatures list to indicate there availability.

3.5 Notification

The Notification message is sent by both machines before a connection is terminated (e.g., after protocol errors or before shutdown). It could also be used for general notification purposes.

Notification	Type	Range / Multiplicity	Optional	Description
NotificationCode	int	1 .. n	no	A notification code of the list below. Notification codes above 1000 are not defined by this protocol and may be used by the application.
Severity	int	1 .. 4	no	A value of the list below.
Description	string	any string (minimum supported length: 254 bytes)	no	An English textual description of the notification.

The following NotificationCodes are defined:

1 Protocol error (invalid transition in the state machine, see 2.3.5)
2 Connection refused because of an established connection
3 Connection reset because of changed configuration
4 Configuration error
5 Machine shutdown

Possible values for Severity:

1 Fatal error
2 Error
3 Warning
4 Info

3.6 BoardAvailable

The BoardAvailable message is sent to the downstream machine to indicate the readiness of the upstream machine to handover a printed board.
Optional attributes indicate that it is not required to provide them if the information is not available. But if the information is available at the machine, e.g., barcode, then the optional attributes **shall** be set, if not already present there. When an optional attribute is received from an upstream machine, it must be passed on (possibly altered) to the next downstream machine.

BoardAvailable	Type	Range / Multiplicity	Optional	Description
BoardId	string	GUID (36 bytes)	no	Indicating the ID of the available board.
BoardIdCreatedBy	string	non-empty string (minimum supported length: 80 bytes)	no	MachineId of the machine which created the BoardId (the first machine in a consecutive row of machines implementing this protocol). The MachineId is part of the Hermes configuration.
FailedBoard	int	0 .. 2	no	A value of the list below.
ProductTypeId	string	any string (minimum supported length: 254 bytes)	yes	Identifies a collection of printed boards sharing common properties.
FlippedBoard	int	0 .. 2	no	A value of the list below.
TopBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the top side of the printed board.
BottomBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the bottom side of the printed board.
Length	float	positive numbers	yes	The length of the printed board in millimeter.
Width	float	positive numbers	yes	The width of the printed board in millimeter.
Thickness	float	positive numbers	yes	The thickness of the printed board in millimeter.
ConveyorSpeed	float	positive numbers	yes	The conveyor speed preferred by the upstream machine in millimeter per second.
TopClearanceHeight	float	positive numbers	yes	The clearance height for the top side of the printed board in millimeter.
BottomClearanceHeight	float	positive numbers	yes	The clearance height for the bottom side of the printed board in millimeter.

GUID must match the regular expression

```
[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}
```

FailedBoard may be one of the following values:

0 Board of unknown quality available
1 Good board available
2 Failed board available

FlippedBoard may be one of the following values:

0 Side up is unknown
1 Board top side is up
2 Board bottom side is up

If FlippedBoard is 2 (Board bottom side is up) then TopBarcode is facing downwards and BottomBarcode is facing upwards. Same applies for TopClearanceHeight and BottomClearanceHeight.
The definition of board bottom and board top side is outside of the scope of Hermes and is to be determined the customer.
Figure 13 gives an explanation of TopClearanceHeight and BottomClearanceHeight.

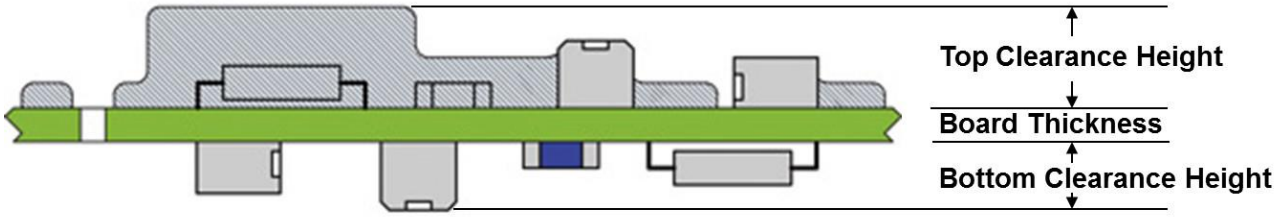

Figure 13 Explanation for Top and Bottom Clearance Height

3.7 RevokeBoardAvailable

With the RevokeBoardAvailable message, the upstream machine signals that it is not ready anymore to handover a printed board.

RevokeBoardAvailable	Type	Range / Multiplicity	Optional	Description

3.8 MachineReady

The MachineReady message is sent to the upstream machine to indicate the readiness of the downstream machine to accept a printed board.

MachineReady	Type	Range / Multiplicity	Optional	Description
FailedBoard	Int	0 .. 2	No	A value of the list below.

FailedBoard may be one of the following values:

0 Ready to accept any board
1 Ready to accept good boards
2 Ready to accept failed boards

3.9 RevokeMachineReady

With the RevokeMachineReady message, the downstream machine signals that it is not ready anymore to accept a printed board.

RevokeMachineReady	Type	Range / Multiplicity	Optional	Description

3.10 StartTransport

The StartTransport message is sent to the upstream machine to initiate the printed board handover process. There is no response to this message.

StartTransport	Type	Range / Multiplicity	Optional	Description
BoardId	string	GUID (36 bytes)	no	The ID of the board for which the transport is started.
ConveyorSpeed	float	positive numbers	yes	Optional parameter indicating the selected conveyor speed for the handover in millimeter per second.

The downstream machine is responsible for selecting the actual conveyor speed according to the preferred conveyor speed sent in the BoardAvailable message. In general the highest possible speed supported by both machines will be selected.
If a StartTransport message is received for a BoardId which is not the one received with the last BoardAvailable message, the transport **shall** be canceled. This case is not to be treated as a protocol error.

3.11 StopTransport

The StopTransport message is sent by the downstream machine after it has finished the transport.

StopTransport	Type	Range / Multiplicity	Optional	Description
TransferState	int	1 .. 3	no	A value of the list below.
BoardId	string	GUID (36 bytes)	no	The ID of the board to which the message relates to.

Transfer states:

1 NotStarted: The printed board never left and hence is fully inside the upstream machine.
2 Incomplete: The transfer was cancelled in progress.
3 Complete: The transfer ended successfully.

If the BoardId does not match the one from StartTransport, this **shall** be treated as a protocol error. Therefore, the connection would need to be re-established.

3.12 TransportFinished

The TransportFinished message is sent by the upstream machine after it finished the transport.

TransportFinished	Type	Range / Multiplicity	Optional	Description
TransferState	int	1 .. 3	no	A value of the list below.
BoardId	string	GUID (36 bytes)	no	The ID of the board to which the message relates to.

Transfer states:

1 NotStarted: The printed board never left and hence is fully inside the upstream machine.
2 Incomplete: The transfer was cancelled in progress.
3 Complete: The transfer ended successfully.

If the BoardId does not match the one from StartTransport, this **shall** be treated as a protocol error. Therefore, the connection would need to be re-established.

3.13 SetConfiguration

The SetConfiguration message is sent by an engineering station to configure the Hermes interfaces of a machine. If the sent configuration is not accepted, the machine is expected to send a Notification message (see 3.5).

SetConfiguration	Type	Range / Multiplicity	Optional	Description
MachineId	string	any string (minimum supported length: 80 bytes)	no	ID / name of this machine for identifying it in a Hermes enabled production line.
UpstreamConfigurations	Upstream Configuration []	0 .. n	no	Configuration for upstream lanes.
DownstreamConfigurations	DownstreamConfiguration []	0 .. n	no	Configuration for downstream lanes.

UpstreamConfiguration	Type	Range / Multiplicity	Optional	Description
UpstreamLaneId	int	1 .. n	no	The lane on the upstream side. Lanes are enumerated looking downstream from right to left beginning with 1.
HostAddress	string	valid IP address or hostname (minimum supported length: 254 bytes)	no	The IP address or hostname of the upstream machine for this lane.
Port	int	0 .. 65535	no	Port number on which connections are established.

DownstreamConfiguration	Type	Range / Multiplicity	Optional	Description
DownstreamLaneId	int	1 .. n	no	The lane on the downstream side. Lanes are enumerated looking downstream from right to left beginning with 1.
ClientAddress	string	valid IP address or hostname (minimum supported length: 254 bytes)	yes	The IP address or hostname of the downstream machine for this lane. If not specified, then connections from any IP address are accepted.
Port	int	0 .. 65535	no	Port number on which the server accepts connections for this lane.

It is up to the user to keep MachineIds unique.

3.14 GetConfiguration

The GetConfiguration message is sent by an engineering station to read out the current configuration of the Hermes interfaces of a machine. The machine is expected to answer with a CurrentConfiguration message.

GetConfiguration	Type	Range / Multiplicity	Optional	Description

3.15 CurrentConfiguration

The CurrentConfiguration message is sent by a machine in response to the GetConfiguration message.

CurrentConfiguration	Type	Range / Multiplicity	Optional	Description
MachineId	string	any string (minimum supported length: 80 bytes)	yes	ID / name of this machine for identifying it in a Hermes enabled production line.
UpstreamConfigurations	Upstream Configuration []	0 .. n	no	Configuration of upstream lanes.
DownstreamConfigurations	DownstreamConfiguration []	0 .. n	no	Configuration of downstream lanes.

For the definition of UpstreamConfiguration and DownstreamConfiguration see 3.13.
If no MachineId has been configured yet, the CurrentConfiguration message does not contain the attribute MachineId.

4 Appendix

4.1 Special Scenarios

The following sections are not part of the Hermes protocol specification. In fact they show the application of this protocol in some special scenarios.

4.1.1 Board Tracking When Board Is Torn Out from the Line

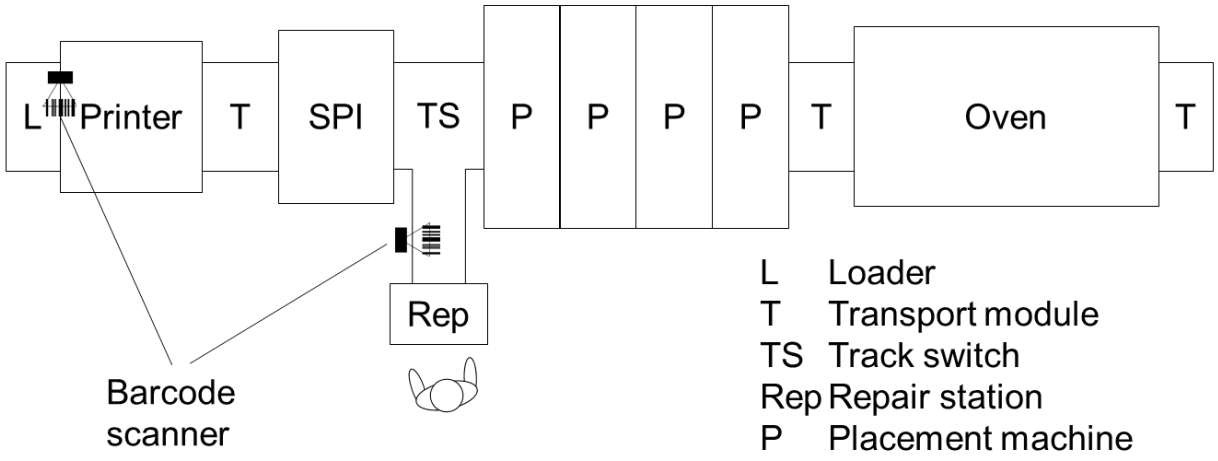

Figure 14 Line Setup with Barcode Readers and Repair Station

In this scenario (shown in Figure 14), a repair station is placed behind the solder paste inspection (SPI). Printed boards failing the solder paste inspection are torn out by the track switch and are presented to an operator at the repair station. The operator may take out the printed board for rework and reinsert it later independent of the printed board sequence.

By removing the printed board from the line, the link between the printed board and the barcode respectively the BoardId is lost. So when the printed board is reinserted, different approaches are possible to re-establish the tracking of the printed board:

a) Create a new Hermes BoardId, read the barcode and report the from now on used tracking information. The tracking information can be merged later by an external system (e.g., manufacturing execution system (MES)) using the barcodes.
b) Read the barcode first and request the corresponding Hermes BoardId from the external system (e.g., MES). The tracking can be continued using the primarily assigned Hermes BoardId.
c) Simplest but most unsecure approach: The repair station prompts the operator to confirm that the inserted printed board is the same which was last removed from the station.

4.1.2 Board Tracking When Board Is Temporarily Removed From the Line

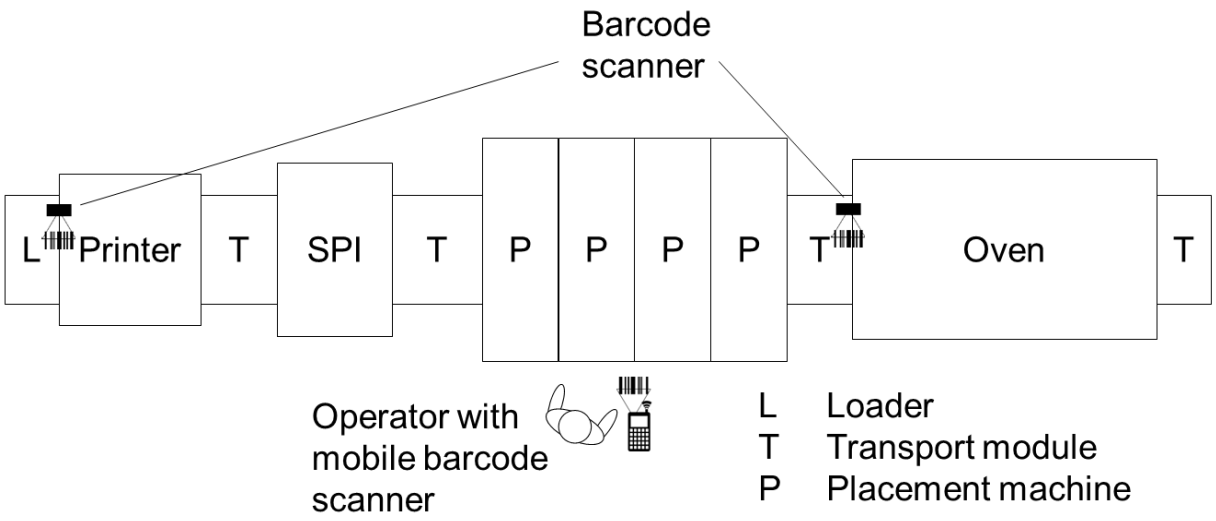

Figure 15 Line Setup with Fixed and Mobile Barcode Readers

In this scenario (shown in Figure 15), the operator removes a printed board for inspection from one of the placement machines. The line continues producing printed boards. At some later point in time, the printed boad is reinserted to complete its production.

By removing the printed board from the line, the link between the printed board and the barcode respectively the BoardId is lost. As in the scenario above, different approaches are possible to re-establish the tracking of the printed board:

a) The machine blocks the production of the reinserted printed board until the operator scans the barcode using a mobile barcode scanner or enters it manually. Then either the original Hermes BoardId is requested from an external system (e.g., MES) using the barcode or a new Hermes BoardId is created and the tracking information is merged by the external system.
b) A new Hermes BoardId is created and production is continued without barcode. At the next barcode reader in the line, the barcode information is complemented to the Hermes BoardId. An external system can later merge all the collected tracking information.

4.2 Glossary / Abbreviations

AOI	Automatical Opticla Inspection
GUID	Globally Unique Identifier
ID	Identifier
IP	Internet Protocol
ISO / OSI	International Organization for Standardization / Open System Interconnection
M2M	Machine-to-Machine
MES	Manufacturing Execution System
SMEMA	Surface Mount Equipment Manufacturers Association
SMT	Surface-Mount Technology
SPI	Solder Paste Inspection
TCP	Transmission Control Protocol
XML	Extensible Markup Language

4.3 References

[IPC_SMEMA_9851]	IPC-SMEMA-9851 Mechanical Equipment Interface Standard
[ISO_7498-1]	ISO/IEC IS 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. 1996
[IETF_RFC_791]	Internet Engineering Task Force: RFC791: Internet Protocol. September 1981
[IETF_RFC_2460]	Internet Engineering Task Force: RFC791: Internet Protocol, Version 6 (IPv6). September 1998
[IETF_RFC_793]	Internet Engineering Task Force: RFC793: Transmission Control Protocol. September 1981
[ITU-T_REC_X.667]	International Standard "Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components
[SemVer_2.0.0]	Tom Preston-Werner: Semantic Versioning 2.0.0. (Internet: https://semver.org/spec/v2.0.0.html, last access: 23. April 2018)
[W3C_XML_1.1]	Extensible Markup Language (XML) 1.1 (Second Edition) - W3C Recommendation 16. August 2006, edited in place 29. September 2006
[W3C_DATE_TIME]	Date and Time Formats - W3C Recommendation 15. September 1997
[W3C_XML_Schema]	XML Schema Part 2: Datatypes Second Edition - W3C Recommendation 28. October 2004

4.4 History

Version	Date	Author	Change
1.0	03/23/17	The Hermes Standard Initiative	Initial Version
1.0.1	11/13/17	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Add Top and Bottom clearance height attribute to Board Available message • When already connected to a downstream machine, reject new connection attempts • Specify the BoardId to be a true globally unique identifier (GUID / UUID) • Remove BoardIdCreatedBy from Start-Transport, StopTransport, TransportFinished
1.0.2	04/23/18	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Application of Semantic Versioning • Define minimum requirements for strings
1.0.3	01/28/19	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Clarification of version number and supported features
1.0.4	11/11/19	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Further clarification of the meaning of "optional" for attributes