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

IPC-HERMES-9852 Best Practices

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

Version 1.3

Contributing Companies:

ASMPT GmbH
ASYS Automatisierungssysteme GmbH
Ersa GmbH
Essemtec AG
IPTE Factory Automation NV
Mycronic AB
Rehm Thermal Systems GmbH
Scheid IT GmbH
SICK AG
SYNEO

Contents:

1 Purpose of this Document 4

2 Common Hermes Topics and Recommendations 5

2.1 Common Hermes Topics 5
2.1.1 Handling of Hermes Data 5

2.2 Recommendations 5
2.2.1 Error Handling and Recovery 5
2.2.1.1 Check alive 5
2.2.1.2 Error before Service Descriptions exchanged 5
2.2.1.3 Lost Connection during Board Transfer 5
2.2.1.4 Handling of a refused Connection due to an already existing Connection 5
2.2.1.5 Missing Barcode Data 5
2.2.2 Data Handling 6
2.2.2.1 Handling of Empty Strings 6
2.2.2.2 Encoding of Attributes with Type float 6

3 Hermes Use Cases 7

3.1 Hermes Basic Board Transfer 7

3.2 Hermes Data Driven Use Cases for Factory Automation Support 7
3.2.1 Automatic Width Adjustment 8
3.2.2 Automatic Program Changeover incl. Automatic Width Adjustment 8
3.2.3 Oven Error Loop 10
3.2.4 Capacity-based oven infeed control by downstream buffer 11
3.2.5 Reverse Transportation 12
3.2.6 Upstream Reverse and Flipping 12
3.2.7 Downstream Reverse and Flipping 13
3.2.8 Manual Removal of Boards 13
3.2.8.1 Monitored Removal 13
3.2.8.2 Removal at next possible withdrawal Point 14
3.2.9 Transfer Subboard 14
3.2.9.1 Info 14
3.2.9.2 FailedBoard vs SubBoards State Attribute 15
3.2.10 Automatic Stacker Level Height Adjustment 16
3.2.11 Expediting Machine Response to Upcoming Changes 16
3.2.11.1 Program Change without controlling of Downstream Machine 16
3.2.11.2 Program Change with controlling of Downstream Machine 17
3.2.11.3 *BoardForecast* with ID and cascading the *BoardForecast* downstream 17
3.2.12 Support of Gate Conveyor (Telescopic Conveyor) 17
3.2.12.1 Fast Preparation for normally open Gate Conveyor (Telescopic Conveyor) 18
3.2.12.2 Controlling a normally closed Gate Conveyor (Telescopic Conveyor) 18
3.2.13 Handling of *BoardForecast* by a Shuttle 18
3.2.13.1 Shuttle 1 Lane to N Lanes 18

3.2.13.2 Shuttle N Lanes to 1 Lane 19

4 Appendix 20

4.1 Glossary / Abbreviations 20

4.2 References 20

4.3 History 20

1 Purpose of this Document

The Hermes Standard (IPC-HERMES-9852) provides a sound basis for transfer of PCBs from Machine to Machine together with its Digital Twin, i.e., the PCB related data. Hence, The Hermes Standard ensures consistency of PCB-related data with the physical PCB throughout the entire SMT Line. This data provides a sound basis to implement use cases for an automated, data driven SMT factory.

This document provides guidelines and recommendations

- to facilitate implementation and work with The Hermes Standard, and
- to implement data driven use cases using data provided by Hermes

The content of Hermes messages evolved over time from version to version. Some of the Hermes data is optional. Therefore, this document will describe for each workflow, which Hermes data is needed, and, thus, provide a guideline for an SMT manufacturer to specify for each machine in the SMT Line

- the necessary minimum Hermes version, and
- the required capabilities concerning handling of Hermes data.

2 Common Hermes Topics and Recommendations

2.1 Common Hermes Topics

2.1.1 Handling of Hermes Data

Machines shall handle data that they received in a Hermes message in one of the following ways:

- **Pass on**: Machine receives data and passes it on unmodified, no reaction on data
- **Update**: Machine receives data, updates it if applicable and then passes it on, no reaction on data
  Note: Machine will create data if it is first in the Hermes Line or if this data does not yet exist
- **React**: Machine receives data, reacts to it and then passes it on unmodified
- **React & Update**: Machine receives data, reacts to it, updates it if applicable and then passes it on

2.2 Recommendations

2.2.1 Error Handling and Recovery

2.2.1.1 Check alive

It is strongly recommended to implement Hermes version 1.1 or higher in which the “check alive” feature is mandatory. Without this feature an unconnected network cable will not be detected until a board transfer message is not responded to. For even more robust detection of connection loss the optional “Check alive response” feature is recommended.

2.2.1.2 Error before Service Descriptions exchanged

Machines cannot use the “Check alive” before service descriptions are exchanged and the supported version of The Hermes Standard is determined. In case a connection error happens, machine software can only show connection state and display a warning to the operator about missing response from upstream/downstream machine.

2.2.1.3 Lost Connection during Board Transfer

Errors during transport should be handled according to The Hermes Standard, see chapter “Transport Error Handling”. It is recommended that machine software handles connection loss towards the operator in a similar way for improved usability.
If no transfer is in progress, i.e. no “Start Transport” message has been sent: try restarting, if this fails raise an error.
If transfer was in progress, it is recommended to consider the board status unsafe and raise an error.

2.2.1.4 Handling of a refused Connection due to an already existing Connection

When a client tries to connect to a machine that already has an active Hermes connection, then this machine will refuse the connection and send a notification with *NotificationCode* 2 “Connection refused due to established connection” and it is recommended to send a *Severity* 2 “Error”.

2.2.1.5 Missing Barcode Data

It is recommended that machines that failed to read a barcode mark the Board as failed using the *FailedBoard* attribute and move it out or raise an error to the operator. A machine that failed reading barcodes should not send *TopBarcode* / *BottomBarcode* attributes.

2.2.2 Data Handling

2.2.2.1 Handling of Empty Strings

Attributes in Hermes messages may be of data type string (range / multiplicity: any string). Such a string may be empty.
It is recommended that machines allow empty strings and that they are prepared to handle them. An empty string indicates that the upstream machine can handle the attribute but does not know its value.

2.2.2.2 Encoding of Attributes with Type float

Attributes with type float should be encoded by rounding to three digits after the decimal point, similar to the requirement for encoding the fraction of the second, see section 3.2 in The Hermes Standard. This avoids lengthy series of '9's that do not provide any reasonable information.
Example: A Board Width of 160.000mm is from a machine handling point of view the same as 159.999999mm. The difference of 1nm does not have any measurable impact on the physical machine.

3 Hermes Use Cases

This section describes the basic Hermes use case, i.e. how HERMES replaces SMEMA, followed by additional use cases to support factory automation and traceability.

3.1 Hermes Basic Board Transfer

Description: A board is transferred from upstream to downstream machine. In addition to the handshake, as already given under SMEMA interface, the machines are enabled to transfer the board data set via The Hermes Standard interface.
Workflow: A machine in the SMT line indicates to send a board to its downstream neighbor and thereby also passes the available board data according to The Hermes Standard. The downstream machine initiates the start of transport via Hermes messages and as soon as both machines internally recognize the correct board departure respectively arrival the transport sequence is finished. The board with its data is now under the responsibility of the receiving downstream machine.
Limitations: none, requirements are valid for all machines in the SMT line
Hermes version: 1.0
Required setup:

- first machine in the SMT line has to setup a Hermes connection to its downstream neighbor
- all machines within the SMT line have to setup a Hermes connection to their upstream neighbor and a Hermes connection to their downstream neighbor
- the last machine in the SMT line has to setup a Hermes connection to its upstream neighbor

Required supported features: none
Data requirements:

- *BoardId*: First machine or creator of Hermes data **Update**, all other machines **Pass on**
- *BoardIdCreatedBy*: First machine or creator of Hermes data **Update**, all other **Pass On**
- *FailedBoard* in Hermes is handled similar to the “Failed Board Available” digital signal in SMEMA:
  - First machine or creator of Hermes data **Update**
  - All machines which actively set the optional “Failed Board Available” digital output on SMEMA N+1 interface, e.g. SPI or AOI **Update**
  - All machines which actively react on the optional “Failed Board Available” digital input on SMEMA N-1 interface, e.g. Rework or Classification Conveyors **React**
  - All other machines **Pass On**
- *FlippedBoard*:
  - First machine or creator of Hermes data: **Update**
  - Machines with board flipping capability: **Update**
  - All other: **Pass On**

Error Condition: If a board is not successfully transferred, so the board arrival and/or the board departure was not successful the transport sequence is completed with the *TransferState = 2 (Incomplete)* and alarms occur on the respective machine(s).

3.2 Hermes Data Driven Use Cases for Factory Automation Support

The following data driven use cases can be implemented using Hermes data:

- Automatic width adjustment
- Automatic program changeover (includes automatic conveyor width adjustment)
- Oven error loop

- Reverse Transportation
- Upstream reverse and flipping
- Downstream reverse and flipping
- Manual removal of boards
- Transfer subboard info
- Automatic stacker level height adjustment

3.2.1 Automatic Width Adjustment

Description: When performing a product change at an SMT line, usually the conveyor width of all machines in the SMT line needs to be adjusted. The Hermes Standard provides a *Width* attribute that is communicated M2M together with the board. Machines without program selection shall use this *Width* attribute to adjust their conveyor width accordingly, whereas machines with program selection will get the correct conveyor width from their program. Each machine shall communicate this width via Hermes to subsequent machine.
Machines with program selection may handle Hermes *Width* and width in their program in three different ways:
1) Use width from program to adjust conveyor width and send this width to downstream machine → **Update**
2) Use *Width* from Hermes to adjust conveyor width and send this width to downstream machine → **React**
3) Use width from program to adjust conveyor width and pass on *Width* from Hermes to downstream machine → **Pass on**

Workflow:

- Machine receives *Width* in *BoardAvailable* message:
  - Machine without program selection adjusts conveyor width prior to transferring the board in
  - Machine with program selection already has the correct conveyor width from previously activated program
- Machine sends the width with *BoardAvailable* message to downstream machine
  - Machine without program selection passes on the received *Width* unmodified
  - Machine with program selection sends width from previously activated program or passes on the received *Width* unmodified

Limitations: none
Hermes version: 1.0 or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features: none
Data requirements:

- *Width*:
  - Machine without program selection **React**,
  - Machine with program selection **Update** or **React** or **Pass on**

Error Condition: In case of a mismatch between *Width* in Hermes and program selection the machine shall raise an error.

3.2.2 Automatic Program Changeover incl. Automatic Width Adjustment

Description: Automatic Program Changeover throughout the entire SMT line uses a product identification, which is provided to the first Hermes machine in the SMT line, and this product identification is communicated via Hermes from machine to machine all the way down the SMT line. Product Identification may be Top- or Bottom-Barcode, Flipped Board, Product Type ID or Work Order ID and Batch ID or a combination of these. Prior to transferring a Board into a machine, this product identification will be received using Hermes and

evaluated by the machine to look up the program required to process this Board.
Note: Automatic Program Changeover usually includes Automatic Width Adjustment, see section 3.2.1.
Workflow:

- Upstream machine will send *BoardAvailable* with *Width*, *TopBarcode*, *BottomBarcode*, *FlippedBoard*, *ProductTypeId*, *WorkOrderId*, *BatchId*
- Board Handling Machines will adjust their conveyor width based on received *Width*, see section 3.2.1 Automatic Width Adjustment
- Process Machines use *TopBarcode*, *BottomBarcode*, *FlippedBoard*, *ProductTypeId*, *WorkOrderId*, *BatchId* or a combination of these attributes to identify the required program and load this program. The loaded program includes conveyor width and machine may adjust its conveyor width accordingly. Instead, machine may use Hermes *Width* to adjust its conveyor width.
  Program may be selected, for example, by
  - Complete barcode (*TopBarcode* or *BottomBarcode*) is used to select the program, or
  - Barcode (*TopBarcode* or *BottomBarcode*) with a pattern applied to cut out the relevant part of the barcode to select the program, or
  - *ProductTypeId* to select the program, or
  - *WorkOrderId* and *BatchId* to select the program, or
  - Barcode (*TopBarcode* or *BottomBarcode*) with a pattern applied to cut out the relevant part of the barcode and this relevant part of the barcode can be combined with *ProductTypeId* with another pattern applied to select the program.
  - *FlippedBoard* should be included in these selection criteria to distinguish between top and bottom side of the board

  Combinations of the above-mentioned Hermes attributes make up a product identification to identify the program. To facilitate configuring the selection of programs, a machine should provide a mapping table that maps such product identifications to programs.
  Note: Different product identifications may map to the same program.
- In case of a program change, machines without stoppers at the end will send a *BoardForecast* with the new *Width* to allow a downstream conveyor to adjust its width accordingly
- In case of a required setup change, machine stops and notifies the operator on its GUI

Limitations: none
Hermes version: 1.1 or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features:

- *FeatureBoardForecast* for machines without stopper at the end and machines downstream of it

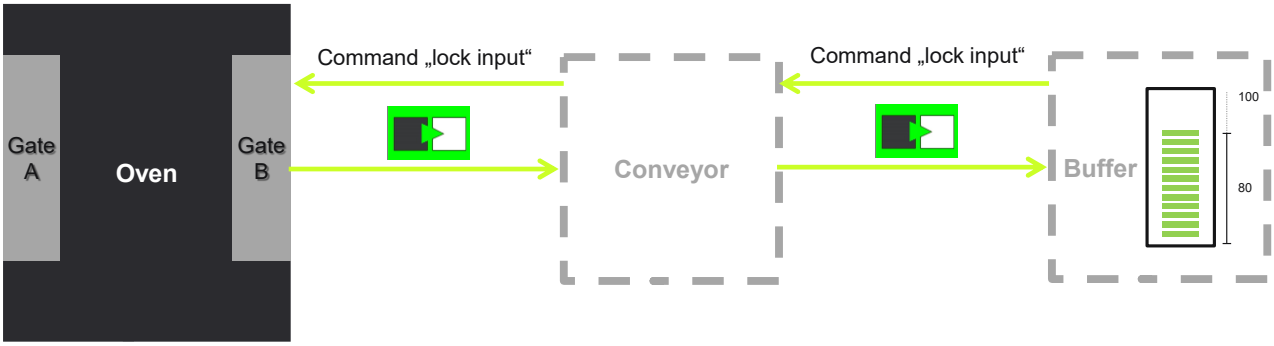
Data requirements:

- *TopBarcode, BottomBarcode*, *FlippedBoard, ProductTypeId*, *WorkOrderId*, *BatchId*: Board Handling Machines **Pass on**, Process Machines **React**

Error Condition: In case of no matching program to download or failure while downloading the machine shall raise an error.

3.2.3 Oven Error Loop

Description: In the case of an oven, in which the transport and process cannot be stopped, it is necessary to block the inlet if the downstream machines are not able to accept any additional boards.



Workflow:

- A machine behind an oven is not able to accept boards, irrespective of the machine being ready or the buffer becoming critical or full.
- A machine shall communicate this situation to the upstream machines using the Hermes *Command* message.
- If a machine receives such a Hermes *Command* and is configured not to react to it, it shall forward this *Command* upstream.
- As soon as the oven receives this *Command*, it blocks its inlet for new boards.
- If the downstream machine is ready again or the buffer space is big enough, the downstream machine communicates this situation again to the upstream machine.
- The oven receives this information and unblocks its inlet.

Limitations: The setting of the prewarning Buffer level needs to be configured according to oven capacity

Hermes version: 1.4 or higher

Required setup: see 3.1 Hermes Basic Board Transfer

Required supported features:

- *FeatureCommand*

Data requirements:

- *Command*:
  - Oven **React** to “Lock input conveyor” and “Unlock input conveyor”
  - Machines between Oven and Buffer **Pass on**
  - Buffer **Update**

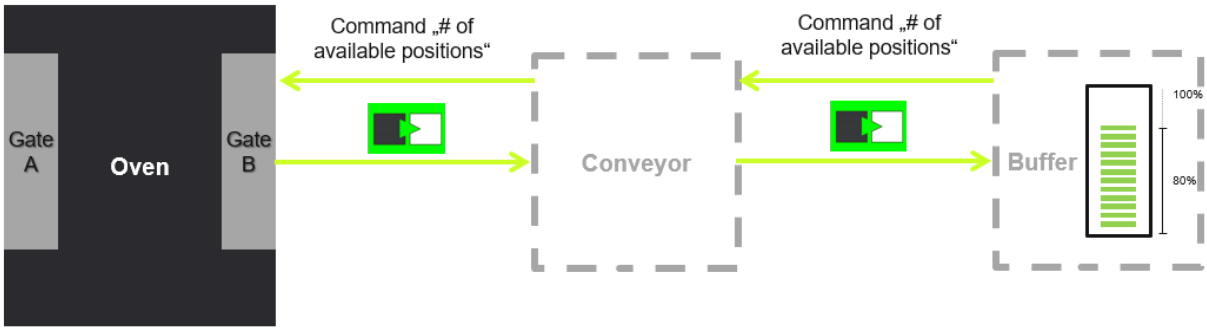
Error Condition: none

3.2.4 Capacity-based oven infeed control by downstream buffer

Oven knows number of boards inside and uses Command attribute to decide whether to lock inlet or not.

Conveyor needs to be configured to
1) don't react
2) add positions
3) deduct positions

Buffer sends a *Command* message with "number of available positions"



Workflow:

- A downstream buffer has a limited number of available positions.
- The buffer uses the Command message to communicate its available capacity upstream to the oven. The Command attribute is an integer. Values from 1000 onwards are available for customer-defined commands. The range 1000 to 1999 is used to represent the number of available positions.
- Machines located between the oven and the buffer forward the information upstream. Depending on their function and current board occupancy, they may forward the value unchanged, add available positions, or deduct occupied positions.
- The oven compares the received number of available downstream positions with the number of boards currently inside the oven.
- Based on this comparison, the oven determines whether additional boards may enter or whether the oven inlet must be blocked.

Limitations: none
Hermes version: 1.4 or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features:

- *FeatureCommand*

Data requirements:

- *Command*:
  - Oven: **React** to values 1000–1999
  - Machines between oven and buffer need to be **Configured** and should **React** in one of three modes:
    1. No change: Forward value unchanged (default)
    2. Add positions: Add available intermediate buffer positions
    3. Deduct positions: Deduct boards that must be transferred to the buffer
  - Buffer **Update**

Condition: none

3.2.5 Reverse Transportation

Description: A board is loaded to a machine and unloaded on the same interface in the opposite direction.
Workflow:

- Board is loaded e.g., from a magazine station to a process machine
- Board gets processed
- Reverse transportation on the same conveyor back to the machine where the board came from.

Limitations: Requirements apply to the affected machines and transport machines in between
Hermes version: 1.1 or higher
Required setup: For reverse transportation all involved machines need to setup one upstream and one downstream connection per gate with reverse transportation.
Required supported features: none
Data requirements: *ServiceDescription*: *InterfaceId* must be set
Error Condition: none

3.2.6 Upstream Reverse and Flipping

Description: A board shall be processed both sides and therefore turned over between process steps. The flipping unit is located upstream to the process machine regarding to the direction of the production line. The flipping unit processes only when instructed by the process machine by using Hermes data.
Workflow:

- Flipping unit loads a board from upstream
- It passes the board through to the process machine unturned
- Process machine sends *BoardForecast* message to the flipping unit to indicate that a board will be reverse transferred soon and space for the board has to be reserved; no new board loading from upstream.
- Protocol and board handling to process machine
- Process machine processes top side of the board
- Protocol and board handling back to flipping unit and indication that the board shall be turned over
- Flipping unit turns over the board as requested
- To the process machine it sends: *BoardAvailable*, *FlippedBoard*=2 (Bottom side is up)
- Protocol and board handling to process machine
- Process machine processes bottom side of the board; flipping unit now can load a new board from upstream if available
- Process machine unloads the board to the downstream Output

Limitations: Requirements apply to the affected machines and transport units in between.
Hermes version: 1.4 or higher
Required setup: For reverse transportation all involved machines need to setup one upstream and one downstream connection per gate with reverse transportation
Required supported features:

- *FeatureBoardForecast*

Data requirements:

- *FlippedBoard*: Process Machines **React**, Flipping Unit **Update,** Board Handling Machines **Pass on**
- *Route:* Process Machines **Update**, Flipping Unit **React,** Board Handling Machine **Pass on**
- *Action:* Process Machines **Update**, Flipping Unit **React,** Board Handling Machine **Pass on**

Error Condition: none

3.2.7 Downstream Reverse and Flipping

Description: A board shall be processed both sides and therefore turned over between process steps. The flipping unit is located downstream to the process machine regarding to the direction of the production line. The flipping unit processes only when instructed by the process machine by using Hermes data.

Workflow:

- Process machine processes top side of a board
- Indication to the flipping unit that the next coming board shall be turned over and returned
- Flipping unit sends BoardForecast message to the process machine to confirm the upcoming reversal transport
- Protocol and board handling to flipping unit
- Process machine must ensure that board can be returned
- Flipping unit turns over the board as requested
- Protocol and board handling back to the process machine
- Process machine processes bottom side of the board
- Protocol and board handling to flipping unit and indication that the board now can be passes through unturned
- Board passes through the flipping unit unturned

Limitations: Requirements apply to the affected machines and transport units in between.

Hermes version: 1.4 or higher

Required setup: For reverse transportation all involved machines need to setup one upstream and one downstream connection per gate with reverse transportation

Required supported features:

- *FeatureBoardForecast*

Data requirements:

- *FlippedBoard*: Process Machines **React**, Flipping Unit **Update,** Board Handling Machines **Pass on**
- *Route:* Process Machines **Update**, Flipping Unit **React,** Board Handling Machines **Pass on**
- *Action:* Process Machines **Update**, Flipping Unit **React,** Board Handling Machines **Pass on**

Error Condition: none

3.2.8 Manual Removal of Boards

3.2.8.1 Monitored Removal

Description: A manual removal of boards might happen on purpose. In order to provide documentation about the event of removal the machine shall inform the Supervisory System. For this purpose, the Hermes message *BoardDeparture* can be used.

Workflow:

- The machine control recognizes the sudden absence of the board (planned or unplanned).
- Optionally it requests the user for a manual confirmation.
- Hermes message *BoardDeparture* is sent to vertical channel including all board attributes.

Limitations: none

Hermes version: 1.2 or higher

Required setup: Hermes vertical channel setup to connect to Supervisory System.

Required supported features:

- *FeatureBoardTracking*

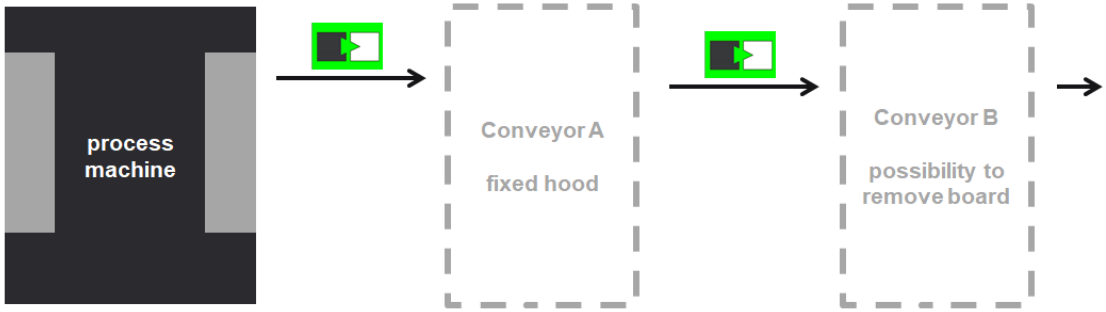
Data requirements:

- *BoardDeparture*/*BoardTransfer*: All machines **Update**
- All other data as specified in *BoardDeparture* message must **Pass on**.

Error Condition: none

3.2.8.2 Removal at next possible withdrawal Point

Description: A process machine instructs the next possible downstream conveyor to stop the board for its manual removal. In this example only conveyor B provides the possibility that the board can be manually removed.



Workflow:

- Process machine sends *BoardAvailable* with *Route* = 999 to conveyor A to indicate removal of the next coming board
- Protocol and board handling to conveyor A
- Conveyor A does not offer the option of removing the board manually
- Board passes through to conveyor B
- Conveyor B requests the user to remove the board
- Removal is monitored (see 3.2.8.1 Monitored Removal),
- Hermes data is discarded for further downstream transportation.

Limitations: Requirements apply to the affected machines and transport units in between.
Hermes version: 1.4 or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features: none
Data requirements:

- *Route*: Machines, in which the call for removal is decided **Update**, machines without the possibility for manual removal **Pass on**, machines with the possibility for manual removal **React**

Error Condition: none

3.2.9 Transfer Subboard

3.2.9.1 Info

Description: Subboard information may be required in case a Board is used with multiple subboards or a carrier which contains multiple subboards. The subboard information can be used to transfer the barcode,

position and the state of each subboard within this panel/carrier. Equipment can then use this subboard information and react to it or just pass it on.
For example, an Inspection System may use the subboard status to decide whether to inspect or skip a certain subboard. Based on the inspection result it may update the subboard status, so that downstream machines know which subboards to process and which to skip.

Workflow:

- First capable machine, e.g., Printer, Inspection or Placer or a supervisory system needs to update the property *Subboards* in *BoardAvailable*
- All downstream machines until the last capable machine at least need to pass on *Subboards*, all capable machines at least need to react, if applicable, react & update

Limitations:

- Functionality is available only downstream
- Number of subboards and barcode size must not exceed the machines message size capabilities

Hermes version: 1.4 or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features: none
Data requirements:

- *Subboards*: **Pass on** or **React** or **React & Update**

Error Condition: none

3.2.9.2 FailedBoard vs SubBoards State Attribute

The FailedBoard attribute in the BoardAvailable message carries information about the board to be handed over. The term „board“ is used in a generic way, standing for any object an upstream machine can hand over (single PCB, multi PCB, tray, carrier, etc.).
For compound objects, like multi-PCB’s, state information for each individual object (sub board) can be conveyed using the State attribute (St), which is part of the SubBoards node.

1) Machines that handle entire objects and do not deal with subboards, like PCB conveyors, buffers or magazine loaders, can make use of the FailedBoard attribute to handle boards according to its state (like sorting Failed/Good boards into designated magazines or pausing transport of boards with „unknown quality“ for manual inspection). These machines will neither make use nor alter available SubBoard data but may forward it along with other BoardAvailable data.
2) Machines that handle subboards individually - like laser marking or routing machines – should make use of the Subboards State attribute to process subboards according to its state (like „bad-marking“ Failed boards or depanelling Good boards only). Additionally, they should maintain the SubBoards State to reflect changes to the product (like setting a subboard position to „Missing“ after depanelling).

SubBoards State data should not be used for handling/process decisions of the entire object. These decisions should be based on the FailedBoard, Route or Action attributes.

3.2.10 Automatic Stacker Level Height Adjustment

Description: The Stacker can calculate its necessary level pitch by using the Hermes *TopClearance* and *BottomClearance* attributes when receiving a board and adjust automatically its level pitch accordingly.
Workflow:

- Stacker receives *TopClearance* and *BottomClearance* in *BoardAvailable* or *BoardForecast* message
- Based on received *TopClearance* and *BottomClearance* Stacker calculates necessary level pitch

Limitations: none
Hermes version: 1.0 (1.1 if using *BoardForecast*) or higher
Required setup: see 3.1 Hermes Basic Board Transfer
Required supported features: *FeatureBoardForecast* if used
Data requirements:

- *TopClearance* and *BottomClearance*: Pick & Place / Mounter **Update**, Stacker **React**, all other machines **Pass on**

Error Condition: max. clearance height of Stacker exceeded

3.2.11 Expediting Machine Response to Upcoming Changes

It is recommended to use *BoardForecast* messages to inform downstream machines as soon as possible about upcoming changes. Receiving such information early, i.e., before a board is arriving, will allow a machine to do necessary adjustments, so that the machine is already prepared for the arriving board, e.g., conveyor width is adjusted, new program is loaded, and setup is changed.
When a machine starts processing a board, this machine should send a BoardForecast message – either forward a previously received BoardForecast message or otherwise create a BoardForecast message with the HERMES data of the board that is currently being processed.

Timing when to send *BoardForecast* and which type of *BoardForecast* to use depends on how the machine controls its output conveyor:

1) Machine that can stop the board at the output conveyor: These machines can wait until the downstream machine is ready for the new product and then transfer the board. Hence, machines that could stop the board at output conveyor need not control the downstream machine concerning preparation for product change.
2) Machines that cannot stop the board at the output conveyor: These machines cannot wait and must transfer the board immediately when they have finished processing it. For example, when the reflow process for a board is finished a reflow oven cannot wait for transferring this board. If such a board continues to stay in the reflow oven, it will be burnt and become unusable. Hence, machines that cannot stop the board at the output conveyor need to control the downstream machine concerning status of preparation for product change.

3.2.11.1 Program Change without controlling of Downstream Machine

When a machine that can stop the board at the output conveyor receives a *BoardAvailable* or *BoardForecast* message, it should react as follows:

- If the machine is empty, i.e., no board is inside this machine, it should send a *BoardForecast* message without *ForecastId*.
  If the machine is not empty, i.e., one or more boards are inside the machine, it should wait until it is empty and then send a *BoardForecast* message without *ForecastId*.

- If the data in the received *BoardAvailable* or *BoardForecast* message indicates a product change, the machine should send a *BoardForecast* message without *ForecastId* before starting the preparation for product change. In case an operator acknowledgement is required for product change, the *BoardForecast* message should be sent after operator acknowledgement.
- In case the machine cannot accommodate the requested changes, it should not send a *BoardForecast* message downstream.

3.2.11.2 Program Change with controlling of Downstream Machine

A machine that cannot stop the board at the output conveyor needs to ensure that the downstream machine is prepared and ready for a board before letting in such a board. This can be accomplished by sending a *BoardForecast* message with *ForecastId*. When a machine that cannot stop the board at the output conveyor receives a *BoardAvailable* or *BoardForecast* message, it should react as follows:

- If the machine is empty, i.e., no board is inside this machine, it should send a *BoardForecast* message with *ForecastId*. It waits until it receives from the downstream machine a *MachineReady* message with the same *ForecastId*. This indicates that the downstream machine is prepared to take the board. If the machine is not empty, i.e., one or more boards are inside the machine, it should wait until it is empty and then send a *BoardForecast* message with *ForecastId* and then wait for a *MachineReady* message from the downstream machine with the same *ForecastId*.
  Upon receiving the *MachineReady* message with same *ForecastId* from the downstream machine, the machine can let the board in.
- A machine that receives a *BoardForecast* message with *ForecastId* should include this *ForecastId* when sending a *MachineReady*.
- If the data in the received *BoardAvailable* or *BoardForecast* message indicates a product change, the machine should send a *BoardForecast* message without *ForecastId* before starting the preparation for product change. In case an operator acknowledgement is required for product change, the *BoardForecast* message should be sent after operator acknowledgement.
- In case the machine cannot accommodate the requested changes, it should not send a *BoardForecast* message downstream.

Note: It is up to each individual machine to send *BoardForecast* with or without a *ForecastId*.

3.2.11.3 *BoardForecast* with ID and cascading the *BoardForecast* downstream

In case that a machine needs to ensure readiness of more than one machine downstream, *BoardForecast* messages with *ForecastId* may be cascaded. Such a cascaded *BoardForecast* will ensure that more than one machine further down the line will be ready to accept a new product, e.g., an oven followed by a cool-down conveyor followed by a buffer.

The workflow for cascaded *BoardForecast* is similar to section 3.2.11.2 “Program Change with controlling of Downstream Machine”. In addition, each machine needs to be configurable to enable/disable forwarding of a received *BoardForecast* with *ForecastId*. In case of cascading *BoardForecast* the machine needs to be set up to cascade a received *BoardForecast* with *ForecastId* before responding to the upstream machine.

Note: For forwarding lines without loops it is recommended not to use cascaded *BoardForecast* messages with *ForecastId*.

3.2.12 Support of Gate Conveyor (Telescopic Conveyor)

A Gate Conveyor or Telescopic Conveyor can be opened to let an operator pass from one side of an SMT line to another:

- Gate open: Operator can pass
- Gate closed: Board can pass

There are two scenarios to consider:

1) Gate Conveyor normally open
2) Gate Conveyor normally closed

3.2.12.1 Fast Preparation for normally open Gate Conveyor (Telescopic Conveyor)

In this scenario, the gate is in default state normally open. The gate will be closed in time to meet the next board's estimated time in the received *BoardForecast* message.

Recommendations:

In this scenario it is recommended to use a *BoardForecast* message without *ForecastId*. It is important to update the expected time of arrival of the board. As soon as an estimate is available for the timespan until the board is ready for downstream, a *BoardForecast* message should be sent. If there is an updated estimate of the arrival time, a new *BoardForecast* message can be sent to optimize the line. It is recommended to have above 1 second resolution of the estimated time.

If the received *BoardForecast* does not contain estimated time information, the Gate Conveyor should not react to this *BoardForecast* message and instead wait for the *BoardAvailable* message.

3.2.12.2 Controlling a normally closed Gate Conveyor (Telescopic Conveyor)

In this scenario, the gate is in default state normally closed. The gate will be opened to let an operator pass through to the other side of the SMT line. It should be avoided opening a normally closed Gate Conveyor when a board is eminent to be transferred.

The recommended handling of this scenario is similar to section 3.2.12.1 "Fast Preparation for normally open Gate Conveyor (Telescopic Conveyor)".

3.2.13 Handling of *BoardForecast* by a Shuttle

Two kinds of Shuttles need to be considered:

- Shuttle 1-to-N lanes
- Shuttle N-to-1 lanes

3.2.13.1 Shuttle 1 Lane to N Lanes

If the Shuttle knows the downstream lane with high probability, then the Shuttle should send *BoardForecast* to that downstream lane with updated estimated time. It is recommended to make it a configuration option whether *BoardForecast* should be sent downstream to all lanes or only specific lanes. In the case of different products *BoardForecast* should not be sent downstream. In the case of similar products on all lanes the *BoardForecast* can be sent.

Note: Only one lane will receive a real board even if all lanes receive a *BoardForecast*.

Please see also the recommendations in section 3.2.12.1 "Fast Preparation for normally open Gate Conveyor (Telescopic Conveyor)".

3.2.13.2 Shuttle N Lanes to 1 Lane

It is recommended to send the *BoardForecast* of the board that will arrive first. For example: Lane 1 has a *BoardForecast* of 10s and lane 2 has a *BoardForecast* of 12s, then *BoardForecast* from lane 1 should be sent first to downstream.

Please see also the recommendations in section 3.2.12.1 “Fast Preparation for normally open Gate Conveyor (Telescopic Conveyor)”.

4 Appendix

4.1 Glossary / Abbreviations

AOI	Automatic Optical Inspection
GUI	Graphical User Interface
ID	Identifier
M2M	Machine-to-Machine
SMT	Surface-Mount Technology
SPI	Solder Paste Inspection

4.2 References

[IPC_HERMES_9852] IPC-HERMES-9852 The global standard for machine-to-machine communication in SMT assembly, V1.7, www.the-hermes-standard.info

4.3 History

Version	Date	Author	Change
0.1	05-Oct-2021	Workgroup Hermes Use Cases	Initial Version
0.2	20-Oct-2021	Workgroup Hermes Use Cases	Updated after Review of Width Adjustment
0.3	26-Oct-2021	Workgroup Hermes Use Cases	Major rework
0.4	27-Oct-2021	Workgroup Hermes Use Cases	Updated after Review
0.5	04-Nov-2021	Workgroup Hermes Use Cases	Updated after Review, pending approval
1.0	14-Mar-2022	Tom Marktscheffel, Peter Sundström	Approved
1.1	13-Sep-2023	Tom Marktscheffel	Added BoardForecast handling, Approved
1.2	11-Nov-2024	Markus Mittermair	SubBoards vs FailedBoard State Attribute
1.3	13-Aug-2026	Markus Mittermair	Added Capacity-based oven infeed control by downstream buffer