HIRSCHMANN
AUTOMOTIVE

Technical Delivery Regulation

MES01 – Manufacturing Execution System

This standard governs the requirements for the documentation and the general regulations for the delivery of systems.

Revision status:
This delivery regulation MES01 replaces all previous regulations.

Version:	Page no.:	Description of change:	Date:
MES01	all	Created on	08.10.2018

Responsibility:	Müller Elias
Department:	DI

Table of contents

1.General 4
1.1. Area of Application 4
1.2. Deviations 4
1.3. Standards/Regulations 4
2.Interface Specification 4
2.1. Interface 4
2.2. Interface Protocol 5
2.3. Machine Information 5
2.4. Counter Information 6
2.5. Packaging Unit Information 6
2.6. Production Order Information 7
3.Programme Structure 8
4.Examples 12

1. General

1.1. Area of Application

This Hirschmann factory standard specifies the delivery regulations for the documentation of machines, systems and production facilities.

1.2. Deviations

Deviations from this delivery specification which may appear necessary or appropriate to the manufacturer, require written approval from Hirschmann Automotive.

1.3. Standards/Regulations

Even if this technical delivery specification does not specify such in detail, the contractor is fully responsible for, in addition to the requirements specified in this technical delivery specification, all requirements applicable to their service arising from regulations (e.g. EC directives, regulations and other applicable laws) as well as from standards and generally accepted rules of technology.

As far as regulations, standards and technical rules are referenced in this technical delivery specification, the contractor himself must check whether they are applicable to his work and whether other regulations, standards and rules are also to be adhered to.

If in doubt, the contractor must immediately contact the client.
In addition, the contractor shall immediately notify the client if the contractor recognises or identifies, on the basis of their knowledge, that the service to be rendered by the contractor is not suitable for the intended purpose or suited only to a limited extent.

2. Interface Specification

2.1. Interface

General
The worldwide standardized OPC-UA (Open Platform Communications Unified Architecture) interface is used as the Hirschmann Automotive standard interface in the shop floor environment. OPC-UA is a requirement of Hirschmann-Automotive. In the event of non-fulfilment, the contractor must contact the client.

This interface specifies the transport of data, security mechanisms and the semantic structure of the data. OPC UA describes both the transport of machine-to-machine data and also the interfaces and semantics of data.

Injection Moulding Machines
The standard interface protocol Euromap 77 must be used for injection moulding. This interface protocol is a requirement of Hirschmann Automotive. In the event of non-fulfilment, the contractor must contact the client.

2.2. Interface Protocol

The Hirschmann Automotive Interface Protocol describes the interface between the machines/systems and the SAP Manufacturing Execution (SAP-ME). The machines and systems generate information that the SAP-ME collects, processes and forwards. This interface can be expanded as required.

The following functions are covered by this interface:

- Machine Information
- Counter Information
- Packaging Unit Information
- Production Order Information

2.3. Machine Information

"OperatingMode" indicates in which mode the machine/system currently is. The options are "Manual", "Auto", "Standby", "Producing", "Error", and "Setup". These modes have the "BOOL" data type. If the machine/system changes to "Error" mode, the error description is sent via "ErrorDescription" as data type "DINT".
A downtime code (DTC) must be generated for every error condition. Only the first DTC has relevance for SAP-ME. All subsequent errors/warnings, etc. are ignored. The DTC table must be provided as a .csv files separated by ";". It is mapped by the MES system. A .csv file must be provided by each individual computer.

Example DTC: DTC 55 is written as a value, the MES system must map the plain text from the .csv file.

											Element Name
Machine	STRUCT										Datatype
	OperatingMode	STRUCT									Example
		Hand	BOOL	TRUE / FALSE							
		Auto	BOOL	TRUE / FALSE							
		Standby	BOOL	TRUE / FALSE							
		Producing	BOOL	TRUE / FALSE							
		Error	BOOL	TRUE / FALSE							
		ErrorDescription	DINT	TRUE / FALSE							
		Setup	BOOL	TRUE / FALSE							
	MESControlling	STRUCT									
		InterfaceOut	STRUCT								
			InterfaceVersion	STRING[5]	'v2.0'						
			BlockVersion	STRING[5]	'v2.0'						
			HeartbeatOut	INT	1..32000						
		InterfaceIn	STRUCT								
			HeartbeatCycleTime	INT	2000ms						
			HeartbeatTimeout	INT	2500ms						
			HeartbeatIn	INT	1..32000						

2.4. Counter Information

The counter information is used to log the produced quantity and the reject parts, and to transfer them to SAP-ME. Date and time are transferred via TimeStamp. Handshake confirms the transfer of the data from the target system. If a "good" component is handed over, this is communicated to the SAP-ME by "QualityOK".
The element "Part" can be expanded, depending on the machine type. This must be clarified with Hirschmann Automotive.

Counter	STRUCT						Element Name	
	InterfaceOut	STRUCT					Datatype	
		Handshake	BOOL	TRUE / FALSE			Example	
		TimeStamp	Date_and_Time	2018-09-14-08:26:00:000				
		User	STRING[20]	Operator				
		Part	ARRAY[n] of STRUCT					
			Part[1]	STRUCT				
				Serial	STRING[40]	'567890-0001'		
				QualityOK	BOOL	TRUE / FALSE		
				FailureDescription	STRING[11]	'NC001002003'		
				Measurement	ARRAY[n] of STRUCT			
					Measurement[1]	STRUCT		
						Name	STRING[20]	'+ST10-BG03'
						Value	REAL	21,234567
						Unit	STRING[5]	'°C'
					Measurement[2]	STRUCT		
					.			
					.			
					.			
					Measurement[n]	STRUCT		
			Part[2]	STRUCT				
			.					
			.					
			.					
			Part[n]	STRUCT				
	InterfaceIn	STRUCT						
		HandshakeAcknowledge	BOOL					

2.5. Packaging Unit Information

If the last part of a packaging unit was produced by the machine, the variable PackingUnit.CurrentPackingUnit.InterfaceOut.PackingUnitFinished is set by SPS to TRUE.
The PackingUnit.CurrentPackingUnit.InterfaceOut.PartsInPackage variable stores the number of parts in the current package.
If the label is printed via the ERP system, the variable PackingUnit.InterfaceOut.CurrentPackingUnit.PrintLabel is also set to TRUE. Subsequently, a handshake loop is performed.
The data for the next packaging unit is transferred via the variables of the interface PackingUnit.NextPackingUnit.InterfaceIn. Subsequently, a handshake loop is performed.

The data for the next packaging unit can also be transferred during the filling of a packaging unit. The data is stored by the SPS. (For example, if the job begins when two or more empty packages are inserted,
the data for the first and then the data for the second package is transferred)
In the case of an MES error, an error message is to be printed on the label, i.e. it has to be replaced.

											Element Name
PackingUnit	STRUCT										Datatype
	CurrentPackingUnit	STRUCT									Example
		InterfaceOut	STRUCT								
			Handshake	BOOL							
			PackingUnitFinished	BOOL							
			PackageNumber	STRING[40]							
			PartsInPackage	DINT							
			PrintLabel	BOOL							
			LastPackingUnitOfOrder	BOOL							
		InterfaceIn	STRUCT								
			HandshakeAcknowledge	BOOL							
	NextPackingUnit	STRUCT									
		InterfaceOut	STRUCT								
			HandshakeAcknowledge	BOOL							
		InterfaceIn	STRUCT								
			Handshake	BOOL							
			PackageNumber	STRING[40]							
			PackageSize	DINT							

2.6. Production Order Information

After the order quantity has been reached, it may be the case that an incomplete packaging unit has been produced as a result of the machine idling.
This packaging unit is recorded as described in Point 6 (see attachment). The actual number is transferred as the quantity.
For the last packing unit of a job, the variable LastPackingUnitOfOrder is set to TRUE, which means that no new NextPackingUnit data is transferred for the next packing unit.

											Element Name
Order	STRUCT										Datatype
	InterfaceOut	STRUCT									Example
		HandshakeAcknowledge	BOOL	TRUE / FALSE							
	InterfaceIn	STRUCT									
		Handshake	BOOL	TRUE / FALSE							
		MESOnly	BOOL	TRUE / FALSE							
		OrderNumber	STRING[40]	'1234567890'							
		Operation	STRING[40]	?							
		OrderTarget	DINT	100.000							
		SerialBase	STRING[40]	'567890'							
		SerialStartIndex	DINT	0							

3. Programme Structure

Machine Interface

	Element Name				**Syntax:**	
	Datatype				**UpperCamelCase**	
	Example					
1	**Machine**	STRUCT				
2		**OperatingMode**	STRUCT			
3			**Hand**	BOOL	TRUE / FALSE	
4			**Auto**	BOOL	TRUE / FALSE	
5			**Standby**	BOOL	TRUE / FALSE	
6			**Producing**	BOOL	TRUE / FALSE	
7			**Error**	BOOL	TRUE / FALSE	
8			**ErrorDescription**	DINT	55	
9			**Setup**	BOOL	TRUE / FALSE	
10		**MESControlling**	STRUCT			
11			**InterfaceOut**	STRUCT		
12				**InterfaceVersion**	STRING[5]	'v2.1'
13				**BlockVersion**	STRING[5]	'v2.1'
14				**HeartbeatOut**	INT	1..32000
15			**InterfaceIn**	STRUCT		
16				**HeartbeatCycleTime**	INT	2000ms
17				**HeartbeatTimeout**	INT	2500ms
18				**HeartbeatIn**	INT	1..32000

1	Machine interface
2	Machine operating mode
3	TRUE: Machine is in manual mode
4	TRUE: Machine is in automatic mode
5	TRUE: Machine is in standby mode
6	TRUE: Machine produces without automatic stop
7	TRUE: Machine is in fault condition
8	Error description if machine is in error state, DTC (Down Time Code) has to be sent. See remark 9
9	TRUE: Machine is in setup mode
10	Information about MES interface
11	
12	Implemented interface/block version, Structure: 'vX.Y', X = MajorVersion, Y = MinorVersion
13	Minor versions are backward compatible, Majors aren't
14	PLC writes a value into HeartbearOut, least once a second, Range (0 - 32000), see remark 1
15	
16	Heartbeat cycle time im ms (default: 1000), see remark 1
17	Heartbeat timeout in ms (default: 500), see remark 1
18	PCo copies HeartbeatOut value into HeartbeatIn, see remark 1

Counter Interface

1	**Counter**	STRUCT						
2		**InterfaceOut**	STRUCT					
3			**Handshake**	BOOL	TRUE / FALSE			
4			**TimeStamp**	Date_and_Time	2018-09-14-08:26:00:000			
5			**User**	STRING[20]	Operator			
6			**TotalPartsSUM**	DINT	123000			
7			**TotalPartsOK**	DINT	120000			
8			**TotalPartsNOK**	DINT	3000			
9			**Part**	ARRAY[n] of STRUCT				
10				**Part[1]**	STRUCT			
11					**Serial**	STRING[40]	'567890-0001'	
12					**QualityOK**	BOOL	TRUE / FALSE	
13					**FailureDescription**	STRING[11]	'NC001002003'	
14					**Measurement**	ARRAY[n] of STRUCT		
15						**Measurement[1]**	STRUCT	
16							**Name**	STRING[20] '+ST10-BG03'
17							**Value**	REAL 21,234567
18							**Unit**	STRING[5] '°C'
19						**Measurement[2]**	STRUCT	
20								
21						**Measurement[n]**	STRUCT	
22				**Part[2]**	STRUCT			
23								
24				**Part[n]**	STRUCT			
25		**InterfaceIn**	STRUCT					
26			**HandshakeAcknowledge**	BOOL				

1	Counter interface
2	Output interface
3	TRUE: Signals new data
4	UTC Timestamp of data
5	User identification
6	Sum of all parts ever produced on this machine
7	Sum of all good parts ever produced on this machine
8	Sum of all bad parts ever produced on this machine
9	Array of part descriptions, more than one if in one step there could be produced more than one part
10	
11	Serial number of first part, comprised of Order Number and Part Counter
12	OK, NOK (see example 5)
13	Failure description if Quality is NOK

	Packing Unit Interface				
1	**PackingUnit**	STRUCT			
2		**CurrentPackingUnit**	STRUCT		
3			**InterfaceOut**	STRUCT	
4				**Handshake**	BOOL
5				**PackingUnitFinished**	BOOL
6				**PackageNumber**	STRING[40]
7				**PartsInPackage**	DINT
8				**PrintLabel**	BOOL
9				**LastPackingUnitOfOrder**	BOOL
10			**InterfaceIn**	STRUCT	
11				**HandshakeAcknowledge**	BOOL
12		**NextPackingUnit**	STRUCT		
13			**InterfaceOut**	STRUCT	
14				**HandshakeAcknowledge**	BOOL
15			**InterfaceIn**	STRUCT	
16				**Handshake**	BOOL
17				**PackageNumber**	STRING[40]
18				**PackageSize**	DINT

1	Packing Unit interface
2	Data of current packing unit
3	Output interface
4	TRUE: Signals new data
5	TRUE: Signals that the Packint Unit ist finished
6	Package Number passed from NextPackingUnit interface
7	Nb. of parts in this package
8	TRUE: Print packing label if Labelprinter is attached to ERP system
9	TRUE: Last packing unit of current order (may not be filled to PackageSize)
10	Input Interface
11	TRUE: Data transfer OK
12	Data of next packing unit
13	Output interface
14	TRUE: Data transfer OK
15	Input Interface
16	TRUE: Signals new data
17	Packing unit identification
18	Package Size

			Order Interface		
1	**Order**	STRUCT			
2		**InterfaceOut**	STRUCT		
3			**HandshakeAcknowledge**	BOOL	TRUE / FALSE
4		**InterfaceIn**	STRUCT		
5			**Handshake**	BOOL	TRUE / FALSE
6			**MESOnly**	BOOL	TRUE / FALSE
7			**OrderNumber**	STRING[40]	'1234567890'
8			**Operation**	STRING[40]	?
9			**OrderTarget**	DINT	100.000
10			**SerialBase**	STRING[40]	'567890'
11			**SerialStartIndex**	DINT	0

1	
2	Output interface
3	TRUE: Data transfer OK
4	Input Interface
5	TRUE: Signals new data
6	TRUE: Produce only if MES connection is active, see remark 2
7	Next order number
8	ERP description of operation (should be displayed in HMI)
9	Number of parts to be produced
10	Serial number basis, see remark 3
11	Number to start with, see remark 3

Remarks

- 1. The value written every HeartbeatCycleTime by the PLC to the HeartbeatOut variable must be copied from the PCo to the HeartbeatIn variable at the latest HeartbeatTimeout. After this time, a connection break (see remark 2) is assumed.

- 2. If an interruption of the MES connection (timeout at Heartbeat) is detected, a fault message must appear on the HMI. If MESOnly is set to TRUE, the error message should not be acknowledged since production can only be carried out with active MES. If MESOnly is set to FALSE, the handshake signals of the Counter, PackingUnit and Logging interface are ignored after acknowledging the error message. This allows the currently active job to be finished.

- 3. The product serial number consists of the SerialBase and an incremented number beginning with the value in SerialStartIndex

4. Examples

1. Start of a new Order

Initial situation:
The machine has just been switched on and is in manual mode. This is signaled by Machine.OperatingMode.Hand = TRUE.

In the ERP system, a new job is created for processing with the following data:
MESOnly = FALSE
OrderNumber = 123456
OrderTarget = 2000
SerialBase = 05092017.123456.
SerialStartIndex = 00000 (tbd)
Recipe.DataPoint [1] = "Kundentype ##"
Recipe.DataPoint [2] = "99.5"

The MES sends the data to the machine via the PCo and sets Order.InterfaceIn.Handshake to TRUE.
The data is transferred from the PLC and the handshake (HS) is acknowledged (Order.InterfaceOut.HandshakeAcknowledge = TRUE),
The acknowledgment of the HS is set to FALSE by the PCo (Order.InterfaceIn.Handshake) and then by the PLC (Order.InterfaceOut.HandshakeAcknowledge). This sequence is referred to hereinafter as a handshake cycle.
With this procedure, a new job was transferred to the machine.
The machine then signals the operator that a new job is to be produced and displays the details in the HMI.
All necessary preparation is made by the operator and the machine is set to automatic mode:
Machine.OperatingMode.Hand = FALSE; Machine.OperatingMode.Produce = TRUE

2. Order Processing with logging

Initial situation:

a order was started as described in Example 1.

Measurement values are determined and assessed at two test stations, a measured value within the tolerance at the first test station,
at the second station two measured values, the second NOK being:
Logging.Point.Point[1].InterfaceOut.Timestamp : fill with actual UTC timestamp
Logging.Point.Point[1].InterfaceOut.Serial = "05092017.123456.00011"
Logging.Point.Point[1].InterfaceOut.Measurement.Measurement[1].Minimum = 9.98
Logging.Point.Point[1].InterfaceOut.Measurement.Measurement[1].Value = 9.99
Logging.Point.Point[1].InterfaceOut.Measurement.Measurement[1].Maximum = 10.02
Logging.Point.Point[1].InterfaceOut.Handshake = TRUE
Logging.Point.Point[2].InterfaceOut.Timestamp : fill with actual UTC timestamp
Logging.Point.Point[2].InterfaceOut.Serial = "05092017.123456.00010"
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[1].Minimum = 19.98
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[1].Value = 19.99
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[1].Maximum = 20.02
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[2].Minimum = 5.98
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[2].Value = 5.96 **<- NOK**
Logging.Point.Point[2].InterfaceOut.Measurement.Measurement[2].Maximum = 6.02
Logging.Point.Point[2].InterfaceOut.Handshake = TRUE

Setting the Logging.Point.Point [#]. InterfaceOut.Handshake variables to TRUE, the handshake cycle is started.

3. MES Problem during Order processing with MESOnly = FALSE

Initial situation:
The value #### is written by the PLC in Machine.MESInformation.InterfaceOut.HeartbeatOut.
After HeartbeatTimeout, the value has not yet been copied from PCo to Machine.MESInformation.InterfaceIn.HeartbeatIn.
The PLC assumes a fault in the MES connection and stops the processing of the job, changes to the error state (Machine.OperatingMode.Error = TRUE, Machine.OperatingMode.ErrorDescription = "MES communication fault") and displays a message on the HMI.
Since order processing was also allowed without MES (Order.InterfaceIn.MESOnly = FALSE) the order processing is continued after acknowledgment of the error message on the HMI. The handshake cycles of the counter, PackingUnit and logging interface are ignored. (See example 6). In the case of an MES error, an error message is to be printed on the packing unit label so that it is recognizable that it should be replaced.
If the job is completed, however, no new job can be started since the necessary information via the order interface is missing.

4. MES Problem during order processing with MESOnly = TRUE

After the connection error has occurred, the machine enters the error state and a fault message is displayed,
but this can not be acknowledged because execution of the order without MES connection is not allowed (Order.InterfaceIn.MESOnly = TRUE).
If a correct heartbeat cycle is carried out again, the error messages can be acknowledged and the processing of the order is continued after acknowledgment.

5. Product finished

When a product is finished, this is displayed via the counter interface. In the variable Counter.InterfaceOut.User, the user identification of the operator is passed, in the Counter.InterfaceOut.Part.Part [1] .Serial the serial number.
The variable Counter.InterfaceOut.Part.Part [1].Quality is used to specify the classification. Possible values are OK for good parts and NOK for bad parts, the error reason is to be specified in the variable Counter.InterfaceOut.Part.Part [1] .FailureDescription.
Parts produced for machine setting, laser tests or similar are also reported as NOK parts, the error description must start with "INITIALIZE:" followed by the reason. These parts are not good parts, but they must not be included in the fault statistics as bad parts.
A number of standard error descriptions are defined during the test with the pilot line.

6. Packing unit finished

When the last part of a packaging unit was produced by the machine the variable PackingUnit.CurrentPackingUnit.InterfaceOut.PackingUnitFinished is set to TRUE by the PLC.
In the PackingUnit.CurrentPackingUnit.InterfaceOut.PartsInPackage variable the number of parts in the current package is stored.
If the label is to be printed via the ERP system, the variable PackingUnit.InterfaceOut.CurrentPackingUnit.PrintLabel is also set to TRUE. Subsequently, a handshake cycle is performed.
The data for the next packaging unit are transferred via the variables of the PackingUnit.NextPackingUnit.InterfaceIn interface. Subsequently, a handshake cycle is performed.
The data for the next packaging unit can also be transferred during the filling of a packaging unit. The data is stored by the PLC. (For example, when the job starts when two or more empty packages are inserted, the data for the first and then the data for the second package are transferred)
In the case of an MES fault, an error message is to be printed on the label, which means that it must be replaced.

7. Order finished

After reaching the order number, it may be the case that an incomplete packaging unit has been produced due to the machine empty run.
This packing unit is booked as described in Example 6. The actual piece number is transferred as the quantity.
For the last packaging unit of a job, the LastPackingUnitOfOrder variable is set to TRUE, which means that no new NextPackingUnit data is transferred for the next packaging unit.

8. Logging of machine values (for example energy consumption values)

Measured energy values (power, air consumption, etc.) can also be stored via the logging interface.
A logging point not used for product measurements is used for this purpose. The machine serial number is used as serial number (e.g. Logging.Point.Point[3].InterfaceOut.Serial = "HIRSCHMANN 500"), the individual measured values are transferred in the measuring points. If limit values are known, these can be transferred with the measured value.
If the limit value is exceeded, an action can be carried out by the MES. For example, create a maintenance order.

9. DTC Generation and Execution

A Downtime Code has to be generated for every fault state.
The DTC has to be the first reason why the machine is stopped. All following errors/warnings etc. will be ignored.
The DTC Table has to be provided as .csv file seperated by ";".
It will be mapped by the MES system. The .csv file has to be provided by every single machine.
For example: DTC 55 is written as a value, the MES system has to map the clear text out of the .csv file.