

# Make2Pack at GRUNDFOS

- Current typical production and future concept
- Challenge: data types in PackTags
- Own additions to PackTags
- Possible inconsistency in PackTags
- Other wishes/challenges

# Current typical production

- Equipment from different vendors, that doesn't know each other
- Pallets carrying data
  - Order- and product information
  - Product ID
  - Bill of material
  - Return parameters



## Future concept

- Vendors deliver equipment with standard interfaces based on, inter alia, the Make2Pack standards
- No pallets carrying data
- One globally used MES system
  - Supplies and acquire (to and from the equipments) all information carried in the pallets today.



# Challenge – Data types in PackTags

- **7.5.1.9.1 Command.Product[#].ProductID**
  - Data Type: **INT (32bit)**
  - Tag Descriptor: Structured Array of Product ID#
  - This Product ID is used to indicate to the machine which product it is producing (i.e., **SKU or UPC**). The array can be used for machines that run multiple products.
- **7.5.1.9.3.1 Command.Product[#].Ingredients[#].IngredientID**
  - Data Type: **INT (32bit)**
  - Tag Descriptor: Structured Array of Ingredient Ids
  - The Ingredient ID is an arbitrary number associated with the raw material, or ingredient for a particular product number. The user will define the value associated to the ingredient IDs that are used in the operation of the machine for a particular product. Each ingredient should have a distinct ID (SKU or UPC).
- *unitname.Command.Product[0].*
  - *Ingredient[n].Parameter[0].Name = 98765432*
  - *Ingredient[n].Parameter[0].Unit = Pcs*
  - *Ingredient[n].Parameter[0].Value = 4*
  - *Ingredient[n].Parameter[1].Name = TORQUE*
  - *Ingredient[n].Parameter[1].Unit = Nm*
  - *Ingredient[n].Parameter[1].Value = 25*
  - *Ingredient[n].Parameter[2].Name = ANGLE*
  - *Ingredient[n].Parameter[2].Unit = DEGREE*
  - *Ingredient[n].Parameter[2].Value = 90*
- **Wanted: a data type of free choice as is the case with the ISA95 standard**
- (SKU: Stock-keeping unit - a unique identifier for each distinct product and service that can be purchased )
- (UPS: Universal Product Code = barcode)

# Own additions to PackTags

- Grundfos.
  - ProductionInfo[0].
    - HandShake.PackTagDataReady
    - HandShake.OkToProduce
  - ProductData[0].
    - PalletNumber
    - ProductionOrder
    - Product
    - Description
    - OrderQuantity
    - SerialNumber

# Possible inconsistency in PackTags?

- 7.5 Tag Details

Product[#]			UnitName.Command.Product[#]	Product
	ProductID		UnitName.Command.Product[#].ProductID	Int (32bit)
	ProcessVariables[#]		UnitName.Command.Product[#].ProcessVariables[#]	Descriptor
		ID	UnitName.Command.Product[#].ProcessVariables[#].ID	Int (32bit)
		Name	UnitName.Command.Product[#].ProcessVariables[#].Name	String
		Unit	UnitName.Command.Product[#].ProcessVariables[#].Unit	String

- 7.5.3.11.1 Admin.ProdProcessedCount[#].ID

Data Type: INT (32bit)

Tag Descriptor: ID Value of ProdProcessedCount

This is the arbitrary (user defined) ID value of the processed products. This is non-descript value that can be used for any user tag requirements. The ID value can be SKU or a user specific product identifier.

- 7.5.3.11.2 Admin.ProdProcessedCount[#].Name

Data Type: STRING

Tag Descriptor: Structured Array of Names in ProdProcessedCount

The name is used to literally describe the product ID. An example parameter name may be PRODUCT A, ABC PRODUCT, etc. This also could be displayed on HMI screens. The array is typically needed for machines that have quality reporting or PDA (Production Data Acquisition) needs.

- Where does the value for "Name" come from?

## Other wishes/challenges

- Traceability via PackTags

For example:

- Packing products to pallets. Which product ID's on which pallets?
- Which stator (ID) is in which pump?

- Time synchronization via PackTags?

The PLCs must synchronize their internal clocks with each other as well as with the MES system so that e.g. the time of alarms is synchronized.

Further more daylight savings can be centrally managed