

This function block makes it easy to implement a Random Infeed conveyor application with a minimum of programming.

Application Description

Random Infeed conveyors, sometimes called SmartBelt conveyors, take product randomly spaced on an Infeed Conveyor and evenly space the products on an Outfeed Conveyor. Often, though not always, there will be an additional conveyor with lugs between which the products must be placed.

In the example above:

F is a Lugged Conveyor (Not used in all applications). If present, the Lugged Conveyor may or may not be servo driven.

E is the Outfeed Conveyor, which is electronically or mechanically geared to run at the same speed as the Lugged Conveyor (if present).

C and **D** are servo driven Phasing Belts. (Depending on the line speed, product spacing and product handling characteristics, some applications will require only one Phasing Belt while some will require more than one).

B is the Spacer belt, which is normally electronically or mechanically geared to run at the same *product rate* as the Outfeed Conveyor.

A is the Infeed Conveyor which delivers randomly spaced products to the system.

The system will have an encoder on the Outfeed Conveyor, which allows the Phasing Belts to synchronize to it and measure the amount of spacing correction needed.

Sensors are placed at the intersection of each belt to detect the exact position of the products.

Your application program will use one instance of the function block for each Phasing Belt. The actual phasing corrections are made using standard MC_GearRelativeInDist function blocks.

Function Block Inputs:

Enable – Self-explanatory.

Flight Pitch – is the desired spacing from the leading edge of one product to the leading edge of the next.

ProductLength – Self-explanatory.

TargetPosition – The position on the Outfeed Conveyor, relative to the sensor where the leading edge of the product should be placed.

CorrectionPct – The amount of the total phasing correction this Phasing Belt should make.

MaxNegPct – The maximum negative correction the belt should make. (Prevents the product backing up into the next product).

CorrectionZone – The available distance to make the correction.

Sensor – The drive input to which the product sensor is connected.

Tag – A text string used by the Error Log to aid in locating the source of any error caused by this instance of the Function Block.

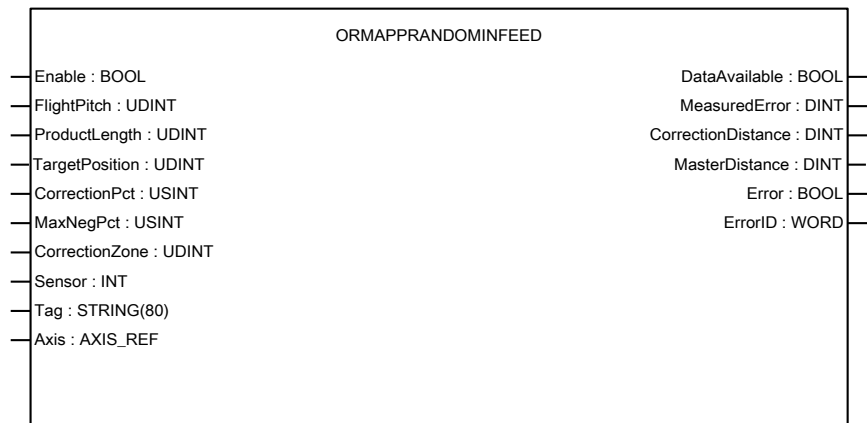
Function Block Outputs:

DataAvailable – Indicates a product has been detected and output data is available.

CorrectionDistance – The amount of phase correction required.

MasterDistance – The line distance available to make the correction.

Error and **ErrorID** – Indicate a function block error.



The OrmAppRandomInfeed function block showing its inputs and outputs.