



**PLCs, Software,  
Conveyor Controls**

JK Bukston, BL. 10  
1618 Sofia BULGARIA  
**Phone:** (+359 2) 975 11 80  
**Fax:** (+359 2) 975 11 81  
**E-mail:** [indsoft@einet.bg](mailto:indsoft@einet.bg)  
[www.indsoft.bg](http://www.indsoft.bg)

---

# ***Connecting ConveyLinX-Ai/Ai2 modules to Siemens S7 PLCs***

## ***Appendix C***

### ***Zone Status/Tracking***

Rev 1.1

December, 2019

## **I. ZPA Status explanation.**

Each ZPA zone has “Status” – value that represents the “Product” relation with the zone.

The possible values are as follows:

0 – Unused by the zone logic. If the PLC writes 0 to a status field(for example to wake up a zone by writing to its Upstream zone status), it will be ignored by the logic.

- 1- Empty(motor may or may not be running)
- 2- Accepting/Empty motor run
- 3- Not used
- 4- Full and motor is running
- 5- Full and motor is stopped
- 6- Busy (used in various special operations like the Merge functionality)

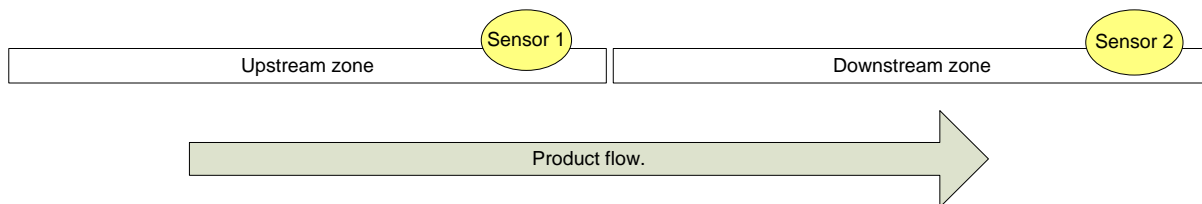


**NOTE: All Status values are decimal numbers.**

To explain the statuses, it is useful to visualize them as follows:

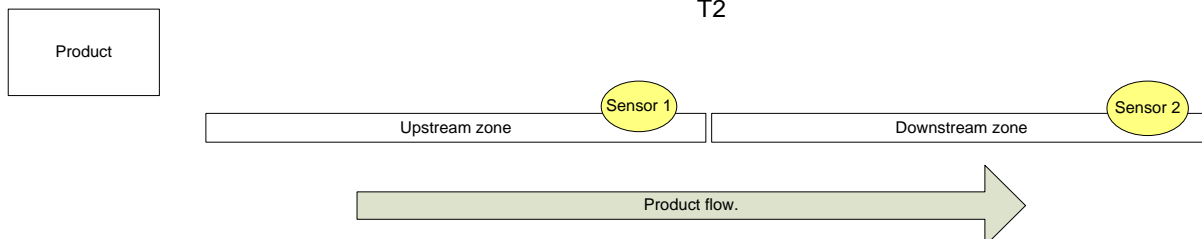
Imagine, that T1, T2, T3 and T4 are time moments of conveyor operation.

T1

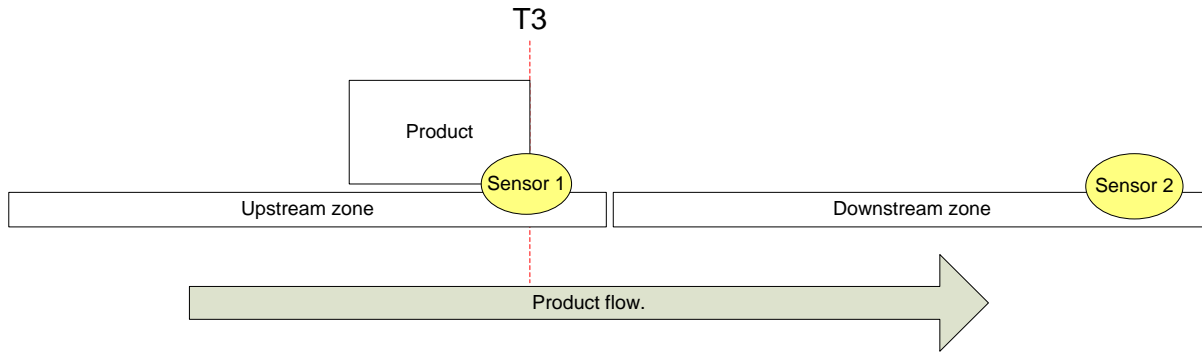


**T1:** There is no “Product” to handle – “Upstream” and “Downstream” zones have “**State = 1**” “**EMPTY**” - No “Product” on the sensor, motor is not running.

T2

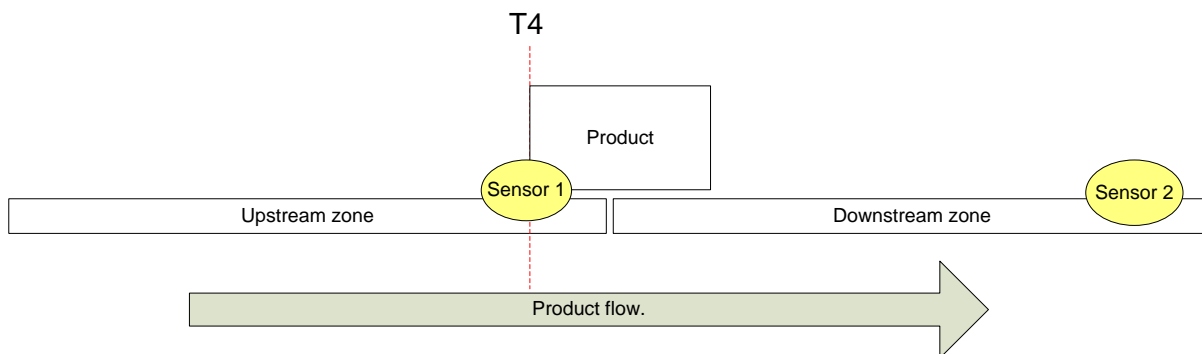


**T2:** “Upstream” zone starts accepting the “Product” – “Upstream” zone now has “**State = 2**” “**ACCEPTING**” - No “Product” on the sensor, motor is running. “Downstream” zone have “**State = 1**” “**EMPTY**”.



**T3:** Case 1 (We have forced accumulation on the “Upstream” zone) – “Upstream” zone changes its state to **“State = 5” “Full and Stopped”** – There is a “Product” on the sensor, the motor is not running. “Downstream” zone still has **“State = 1” “EMPTY”**.

**T3:** Case 2 (“Product” is conveyed downstream) – “Upstream” zone will change its state to **“State = 4” “Full and Running”** – There is a “Product” on the sensor, the motor is running. The “Downstream” zone will now change its state to **“State = 2” “ACCEPTING”**.



**T4:** “Product” unblocks “Sensor 1” – “Upstream” zone will now have the state of **“State = 1” “EMPTY”**. This does not mean that the motor immediately stops and is not rotating. The motor will continue to run for the duration of the “Run After Timer” “Downstream” zone will not change its state to **“State = 2” “ACCEPTING”**.



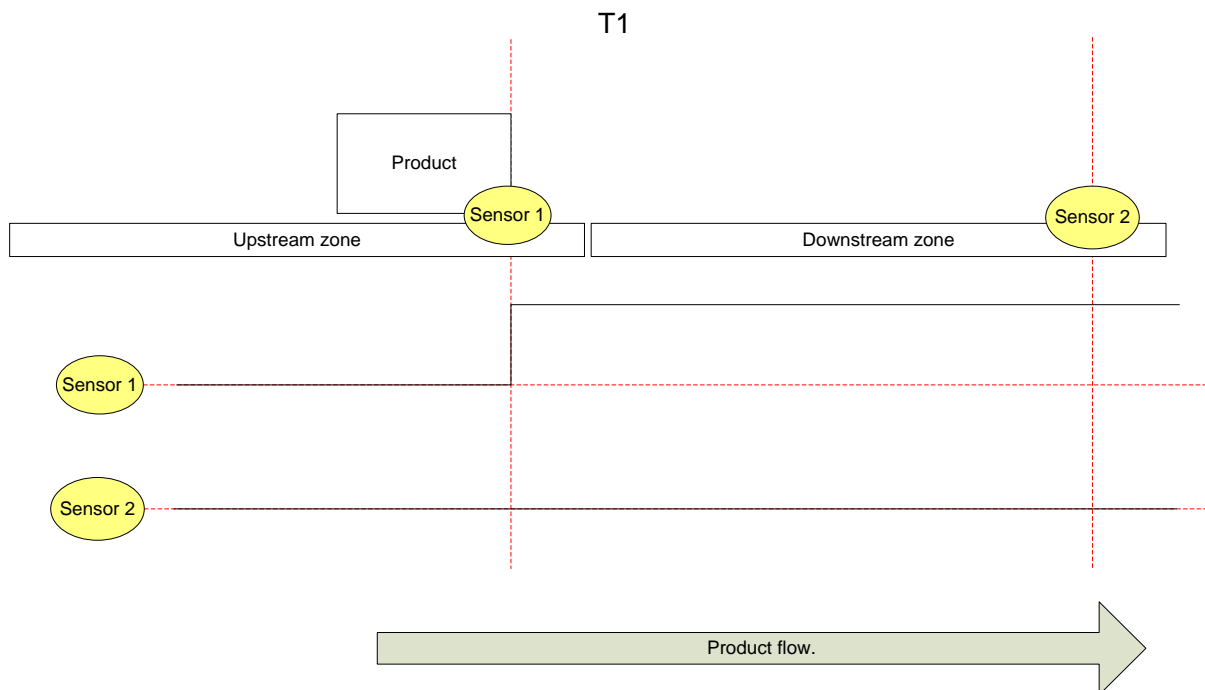
**NOTE:** There is **“State = 6” “BUSY”** - Zone cannot accept “Product”. For example when zone is in “Power up” procedure.

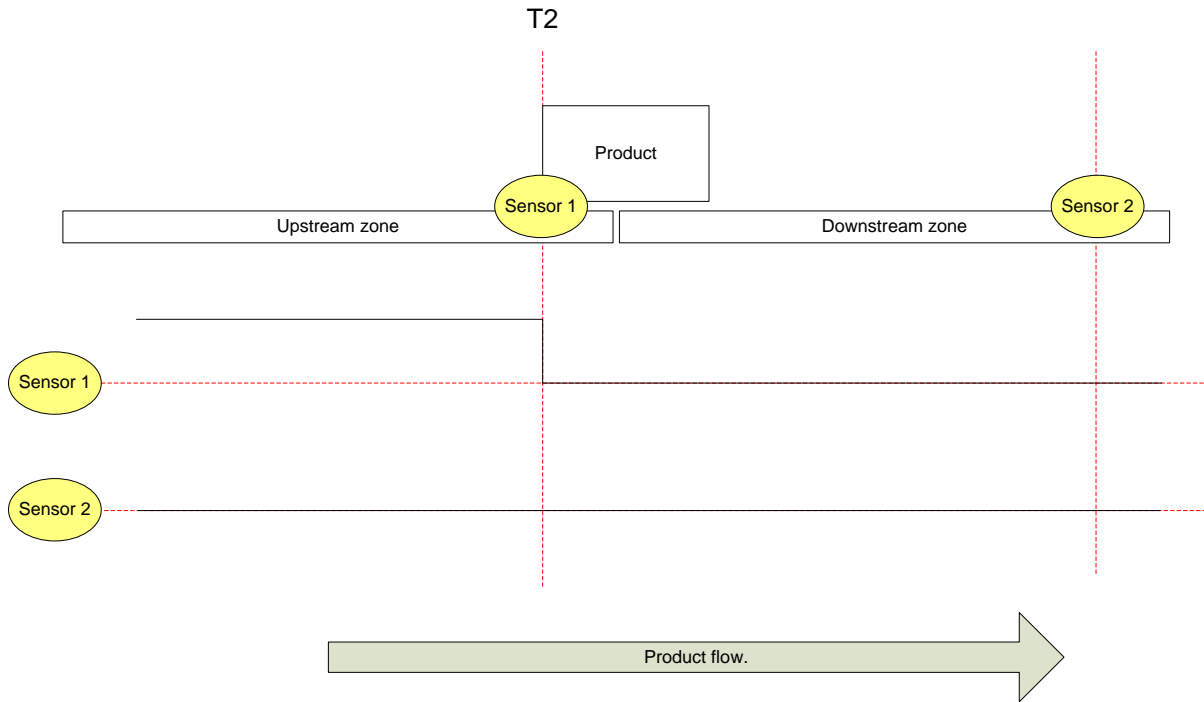
## II. ZPA Tracking.

The tracking number is a 32bit number, which travels with the “Product”. The most secure method to read the Tracking data from a ZPA zone is to follow these steps:

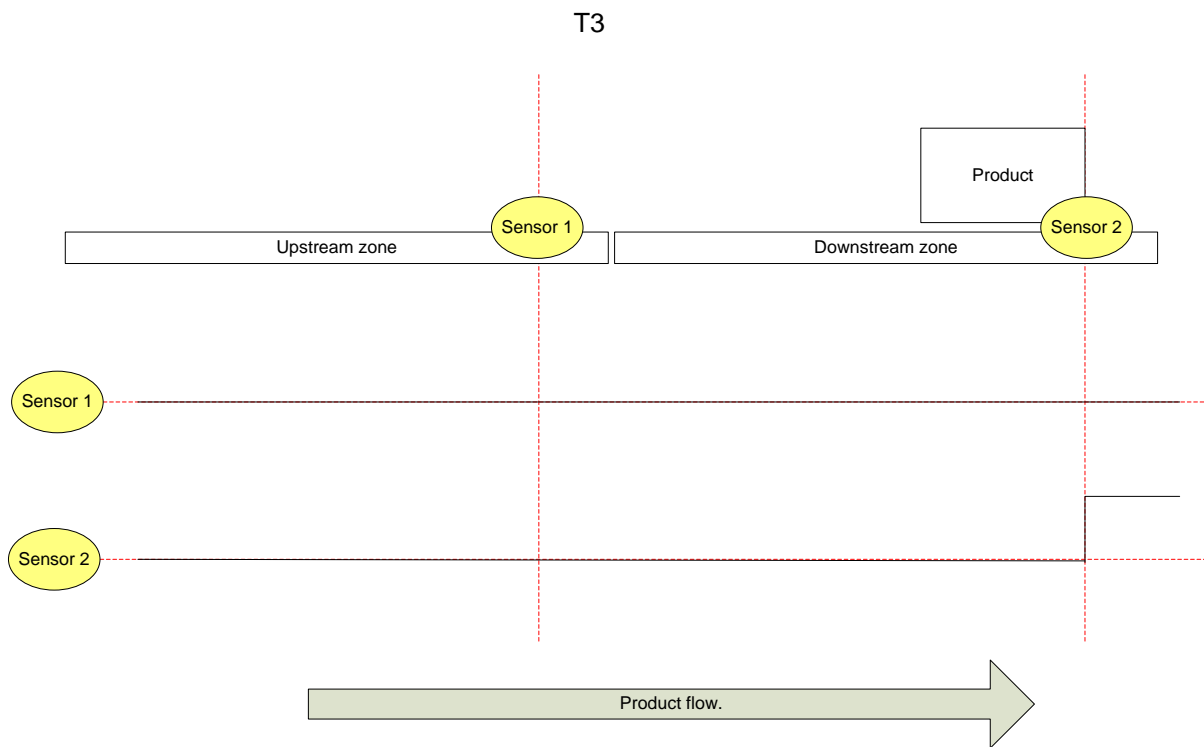
- a. Accumulate, wait for the “Product” to arrive, read the Tracking data, then release the “Product”. Tracking of the zone is in “TrackingUpstreamZone” for the “Upstream” zone, and "TrackingDownstreamZone" for the “Downstream” zone.

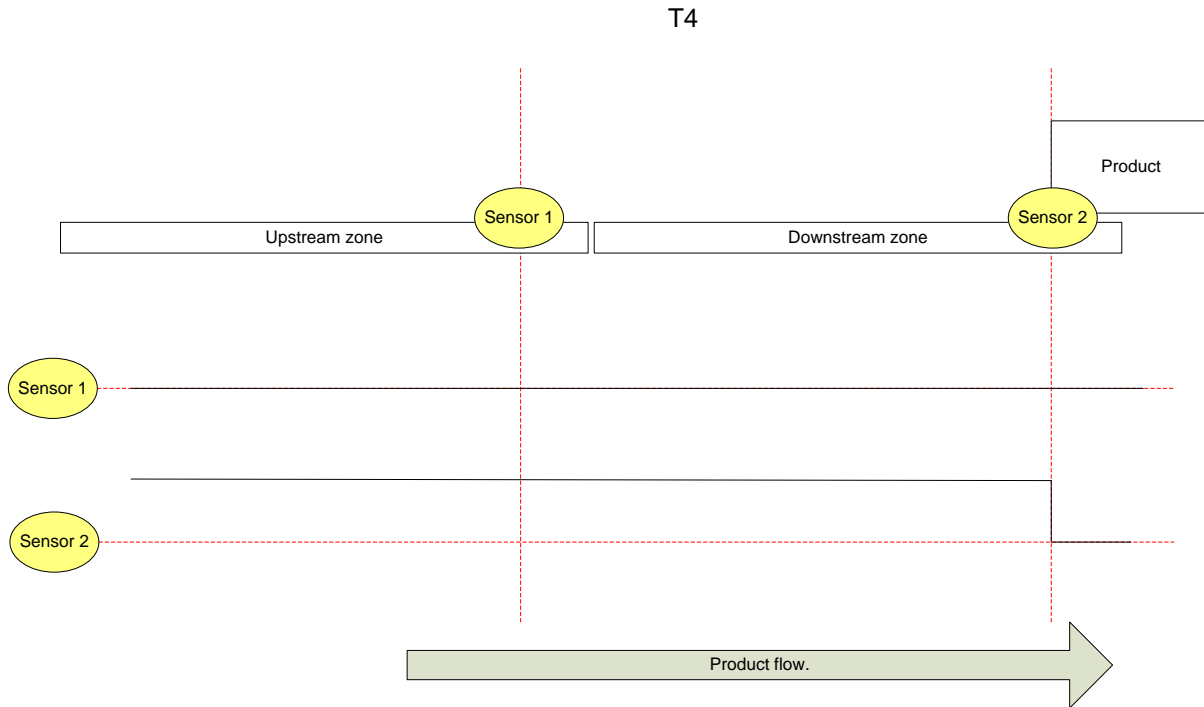
The pictures given below represent four moments (T1, T2, T3 and T4), of the “Product” travel path through the “Upstream” and “Downstream” zones.





The tracking data on the “Upstream” zone is available to be read from moment “T1” to moment “T2” in “TrackingUpstreamZone”.



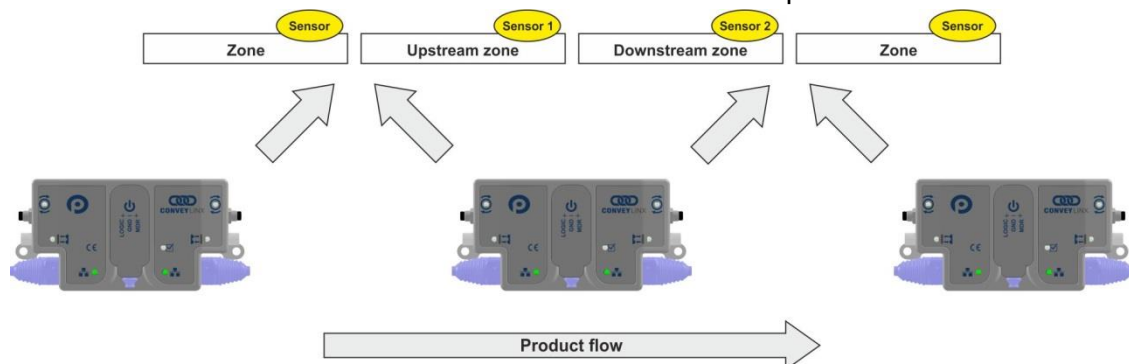


Similarly, the Tracking data on the “Downstream” zone is available from moment “T3” to moment “T4” in “TrackingDownstreamZone”.

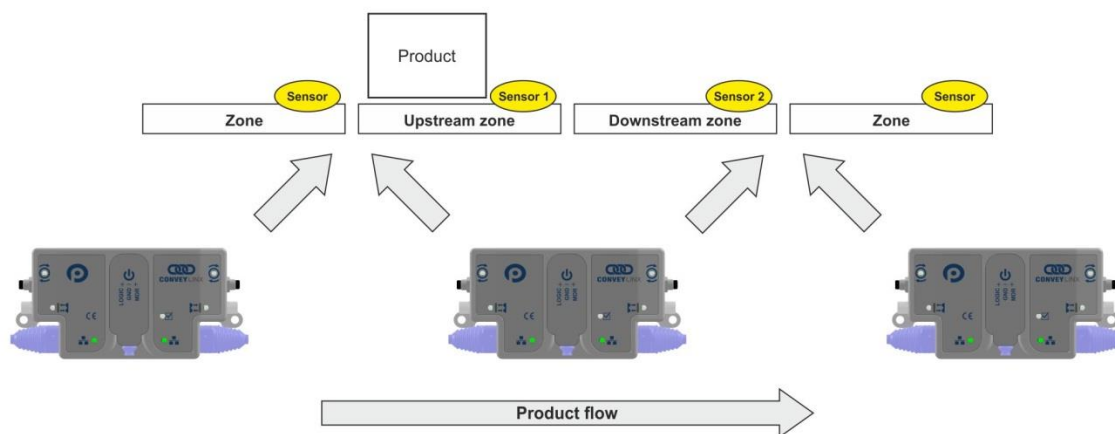
### III. Examples on how to read the “Tracking” from a ZPA zone.

#### Example 1. Read “Tracking” on “Upstream” zone using zone “state”.

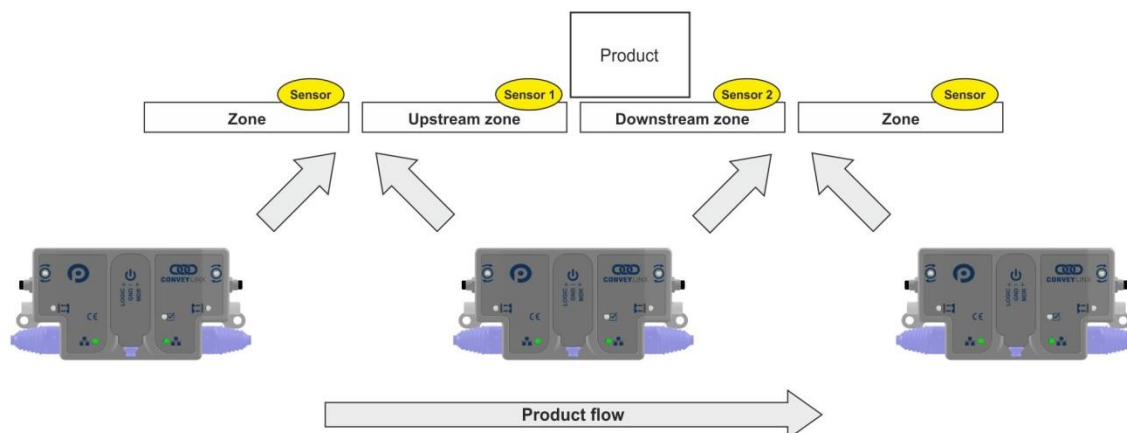
1. Force zone to accumulate – set "AccumulateControlUpstream.Accumulate" to be true.



2. Wait for the “Product” to arrive (T1 moment). T1 moment occurs when the state in “StateUpstreamZone” changes from “ACCEPTING” to “ACCUMULATED PRODUCT(full and stopped)”.



3. Read the "Tracking" from "TrackingUpstreamZone".
4. Release the "Product".



- 4.1. If We do not want to stop the next "Product" – set the "AccumulateControlUpstream.Accumulate" to false.
- 4.2. If We want to stop the next "Product", use "ReleaseCounterUpstreamZone" – read the "ReleaseCounterUpstreamZone" (from the Input data) from the module, increment the value and write it to "ReleaseCounterUpstreamZone" (to the output data). This will release the current "Product", and the next one will be accumulated.

### **Example 2. Read "Tracking" on "Upstream" zone using "ArrivalCounterUpstreamZone".**

1. Force the zone to accumulate – set to 1 "AccumulateControlUpstream.Accumulate".
2. Wait for the "Product" to arrive (T1 moment). T1 moment occurs when "ArrivalCounterUpstreamZone" is incremented by the module.
3. Read the "Tracking" from "TrackingUpstreamZone"
4. Release the "Product".

- 4.1. If We do not want to stop the next “Product” – set the “AccumulateControlUpstream.Accumulate” to false.
- 4.2. If We want to stop the next “Product”, use “ReleaseCounterUpstreamZone” – read the “ReleaseCounterUpstreamZone” (from the Input data) from the module, increment the value and write it to “ReleaseCounterUpstreamZone” (to the output data). This will release the current “Product”, and the next one will be accumulated.



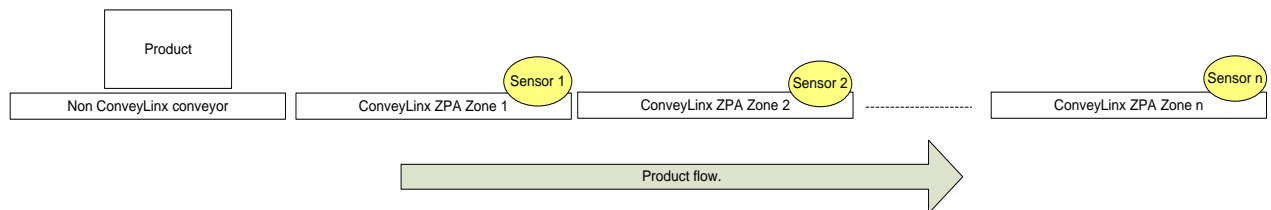
**NOTE: The Accumulate bit in “AccumulateControlUpstream” must be set to 1 when We use “ReleaseCounterUpstreamZone”.**

Major difference between the two examples is that in “Example 2” We are secure from communication loss, and do not care about the “Product” direction flow, when we wait for the “Product” to arrive on the zone.

### Example 3. Write Tracking to Upstream zone.

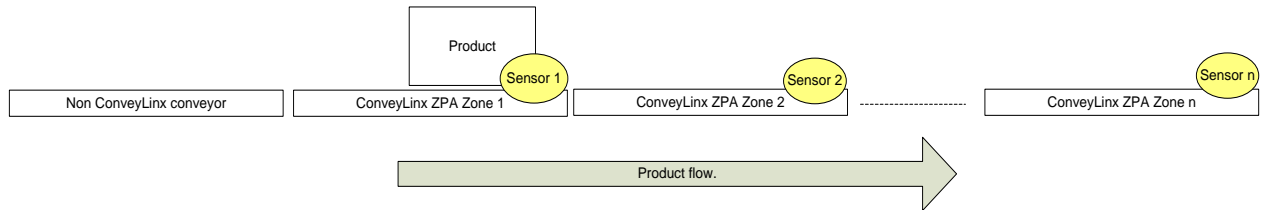
1. Force zone to accumulate – set to 1 “AccumulateControlUpstream.Accumulate”.
2. Wait for the “Product” to arrive (T1 moment). Using “Status” or “ArrivalCounterUpstreamZone”(the arrival counter is better).
3. Write the Tracking data to the “InductTrackingOnUpstreamZone” data field.
4. Release “Product” – read “ReleaseCounterUpstreamZone” (“CLXZPA\_IN”) from module, increment value and write it to “ReleaseCounterUpstreamZone” (“CLXZPA\_OUT”). This will relaes current “Product”, and the next one will be accumulated.

### Example 4. Induct the “Tracking” at the most “Upstream” zone of the conveyor.



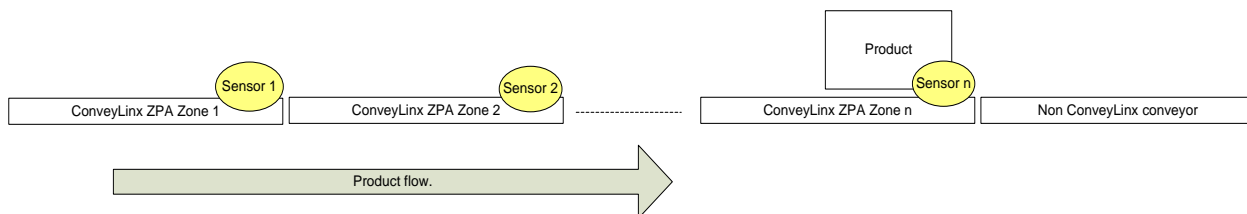
1. Wait “Zone 1” to be ready for accepting the “Product” – “StateUpstreamZone” should be 1(Empty).
2. Write the Tracking data to “ModuleInductTrackingOnInductSide”.
3. Write “State = 4” “SENDING” in “InductControlState” – to wake up “Zone 1”.
4. Run the non ConveyLinX conveyor.
5. Write “State = 1” “EMPTY” in “InductControlState” when the “Product” leaves the non-ConveyLinX conveyor.



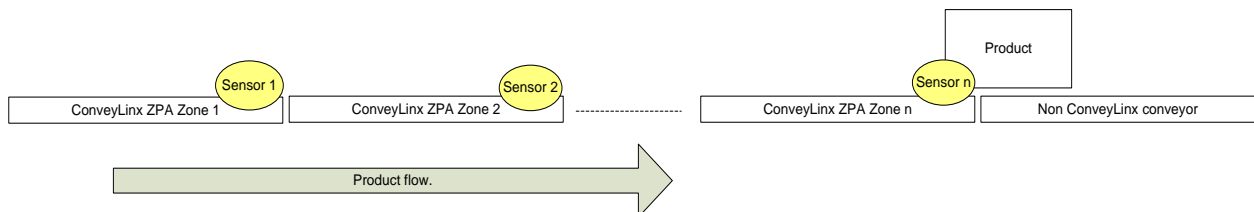


6. Wait for the “Product” to arrive on “Zone 1” – “StateUpstreamZone” == 4 or 5.
7. Set “ModuleInductTrackingOnInductSide” with value zero. This is done, so that the next product is not also inducted with this Tracking. There are protections, but still it is a good practice to reset the procedure.

**Example 5. Get the “Tracking” from the most “Downstream” zone of the conveyor.**



1. Write in the “DischargeControlState = 1” (“EMPTY”).
2. Wait for the “StateDownstreamZone” to become 4 (“SENDING- Full and running”).
3. Write in the “DischargeControlState = 2” (“ACCEPTING”).
4. Run the non-ConveyLinX conveyor.



5. Wait for the “StateDownstreamZone” to become 1 or 2 (“EMPTY or ACCEPTING”).
6. Read the Tracking data from “ModuleDischargeTracking”
7. Wait for the “Product” to arrive on the non-ConveyLinX conveyor.
8. Write “DischargeControlState” to be 5 or 4” to accumulate.

### Example 6. Diverter with ConveyLinx-Ai/Ai2 modules.

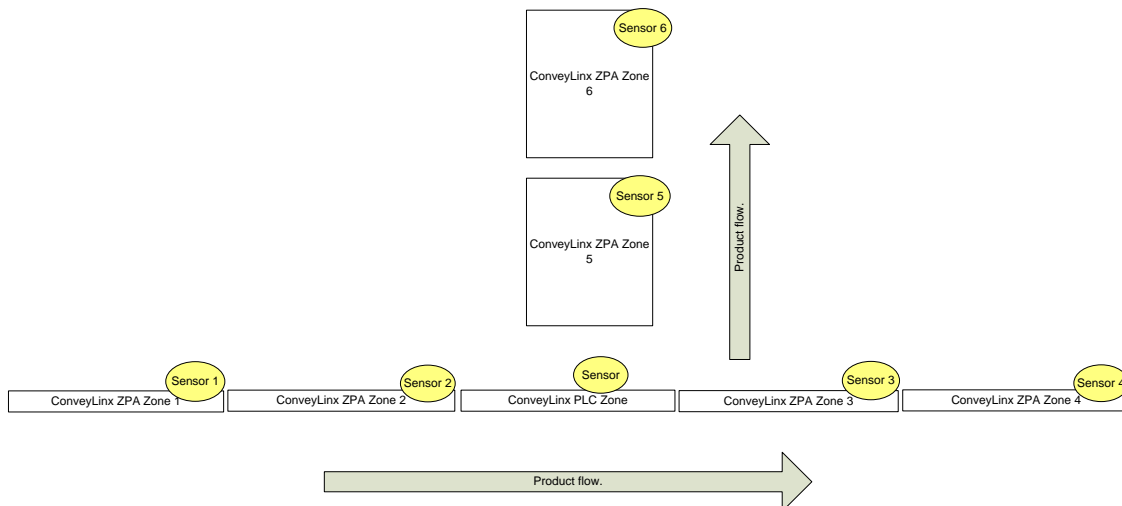
The PLC must be connected to the following modules: “ConveyLinx-Ai/Ai2 containing ZPA zone 2”, “ConveyLinx-Ai/Ai2 containing PLC zone”, “ConveyLinx-Ai/Ai2 containing ZPA zone 3” and “ConveyLinx-Ai/Ai2 containing ZPA zone 5”.

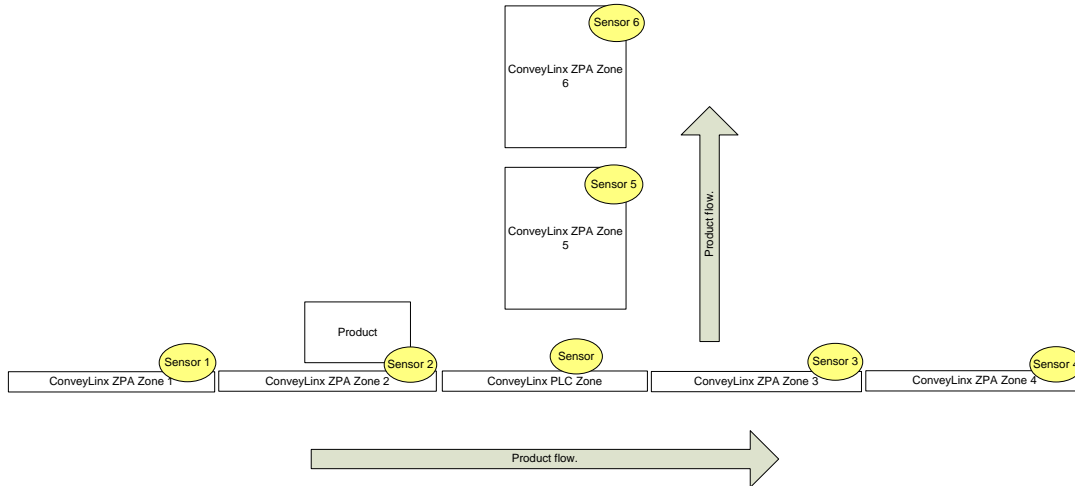


**NOTE:** When making the divert zone ConveyLinx-Ai/Ai2 module in PLC mode, check “Clear Connections” checkbox.

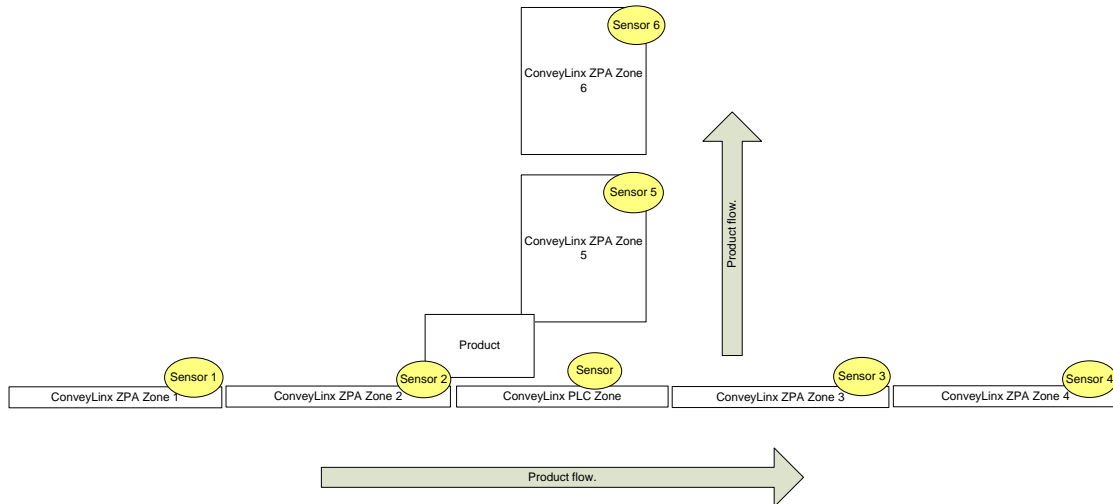
### Accept product.

“ConveyLinx ZPA Zone 2” is a discharge zone from the “PLC zone” point of view, so we can use “Example 5” to accept “Product” with his Tracking data. The only difference is that the PLC must control the mechanics of the diverter zone (“ConveyLinx PLC Zone”) module. In the accepting phase we exchange data only with “ConveyLinx ZPA Zone 2” module, and control “ConveyLinx PLC Zone” (using “CLXPLC\_IN” and “CLXPLC\_OUT”).



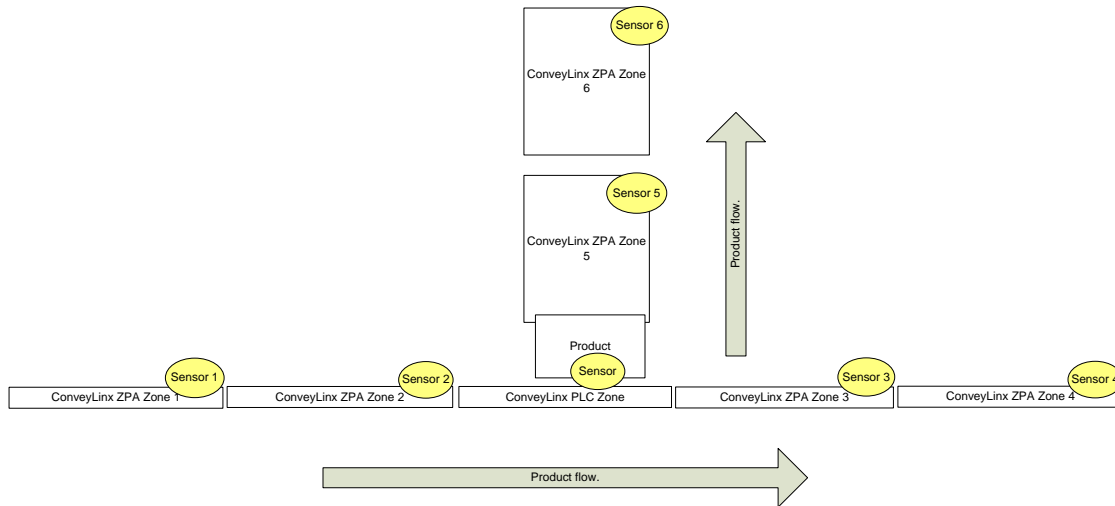


1. Write in the “(ConveyLinx ZPA zone 2.) DischargeControlState = 1” (“EMPTY”).
2. Wait for the “(ConveyLinx ZPA zone 2.) StateDownstreamZone” to become 4 (“SENDING- Full and running”).
3. Write in the “(ConveyLinx ZPA zone 2.) DischargeControlState = 2” (“ACCEPTING”).
4. Run “ConveyLinx PLC Zone” motor.



5. Wait for the “(ConveyLinx ZPA zone 2.) StateDownstreamZone” to become 1 or 2 (“EMPTY or ACCEPTING”).

6. Read the Tracking data from “(ConveyLinX ZPA zone 2.) ModuleDischargeTracking”

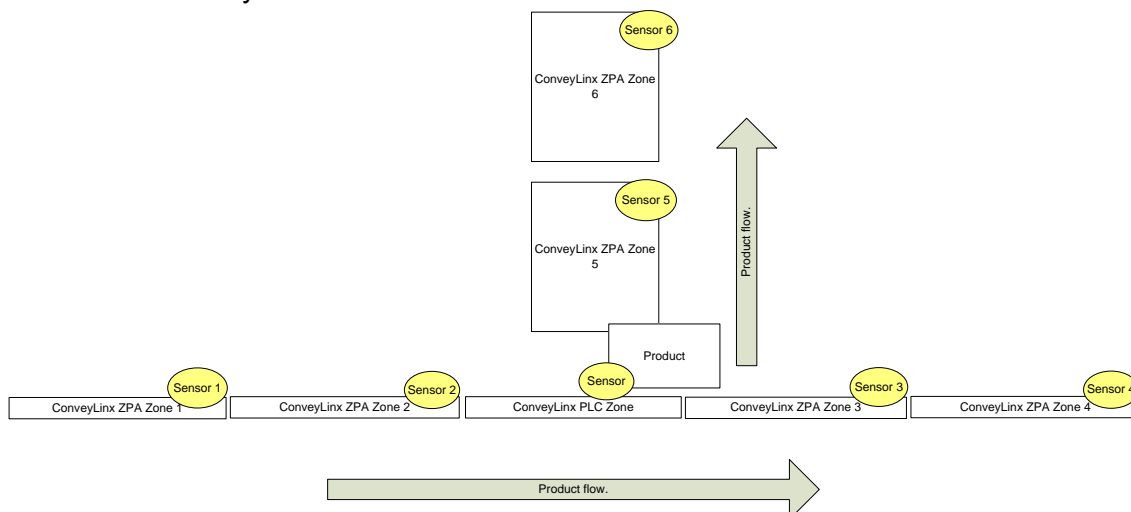


7. Wait for the product to arrive on “ConveyLinX PLC Zone”.
8. Write “(ConveyLinX ZPA zone 2.) DischargeControlState” to be 5 or 4” to accumulate.

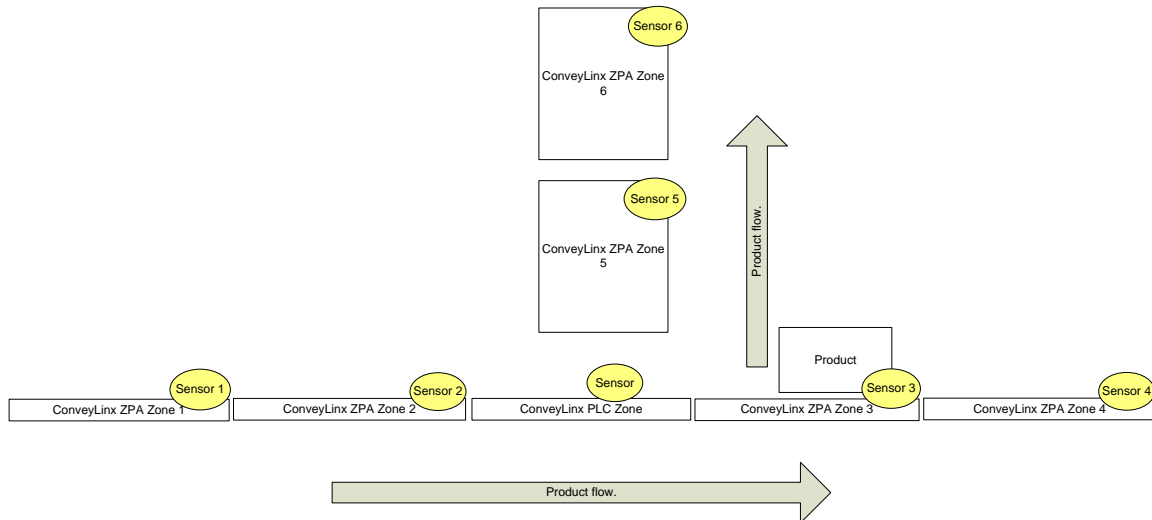
### **Send accepted “Product” down trough main line.**

“ConveyLinX ZPA Zone 3” is an induct zone from the “ConveyLinX PLC Zone” point of view, and like accepting we can use “Example 4” to induct the “Product” with its “Tracking”. In the sending phase we exchange data only with the “ConveyLinX ZPA Zone 3” module, and control “ConveyLinX PLC Zone” module.

1. Wait for the “ConveyLinX ZPA Zone 3” to be ready for accepting – “StateUpstreamZone” = 1.
2. Write Tracking to “(ConveyLinX ZPA Zone 3) ModuleInductTrackingOnInductSide”.
3. Write “State = 4” “Full and Running” in “InductControlState” – to wake up “Zone 3”.
4. Run “ConveyLinX PLC Zone” motor.



5. Write “EMPTY” in “InductControlState” when the product leaves the “ConveyLinx PLC Zone”.

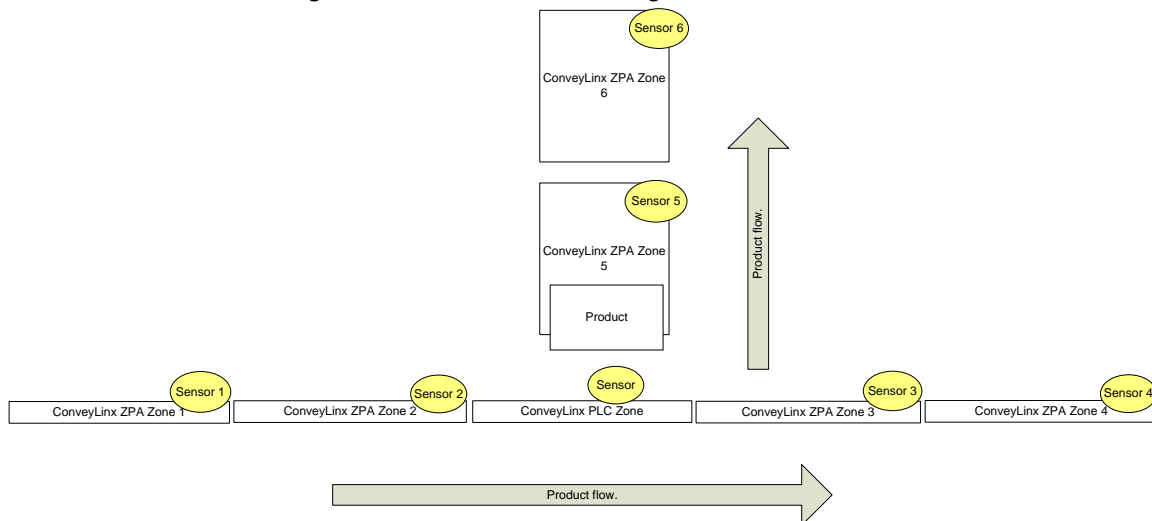


6. Wait for the product to arrive on Zone 3 – “StateUpstreamZone” = 4 or 5.
7. Set “ModuleInductTrackingOnInductSide” with the value of zero.

### Divert accepted “Product”.

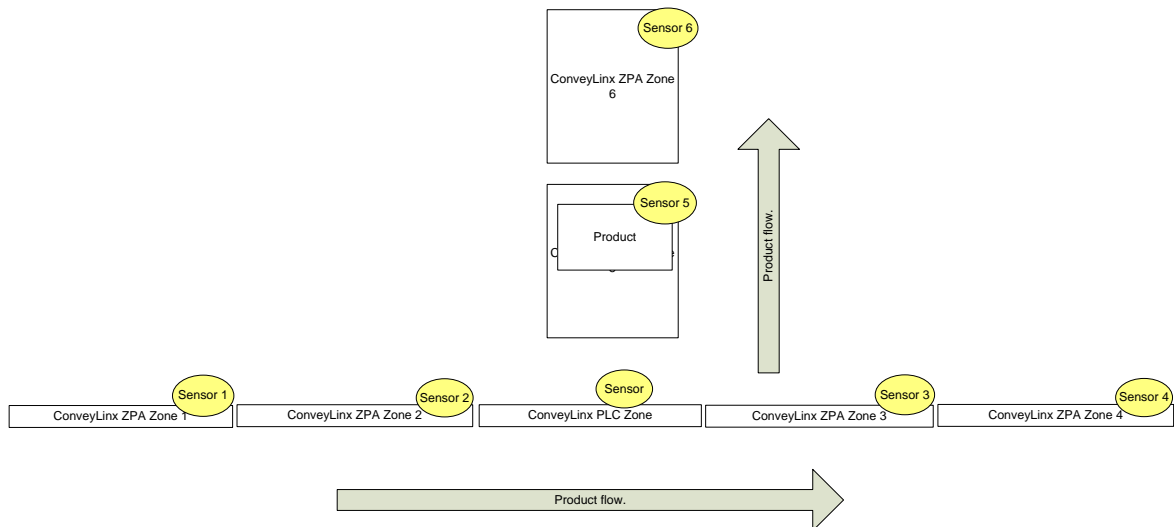
“ConveyLinx ZPA Zone 5” is an induct zone from the “ConveyLinx PLC Zone” point of view, and like the accepting phase we can use “Example 4” to induct the product with its Tracking. In the divert phase we exchange data only with “ConveyLinx ZPA Zone 5” module, and control “ConveyLinx PLC Zone” module.

1. Wait for the “ConveyLinx ZPA Zone 5” to be ready for accepting – “StateUpstreamZone” should be empty.
2. Write the Tracking to “ModuleInductTrackingOnInductSide” of “Zone 5”.



3. Write “**State = 4**” “**Full and Running**” in “InductControlState” – to wake up “Zone 5”.

4. Run “ConveyLinX PLC Zone” motor
5. Write “**EMPTY**” in “InductControlState” when the “Product” leaves the “ConveyLinX PLC Zone” .



6. Wait “Product” to arrive on “Zone 5” – “StateUpstreamZone” = 4 or 5.
7. Set “ModuleInductTrackingOnInductSide” with value zero for zone 5.