



SIEMENS software function blocks

PRORUNNER mk5 with encoder

Printing date : 29 oktober 2018
Version : 4.0
Author : B. Jansen

1 Version control

Tabel 01: Version control

Revision	Date	Author	Order number	Reason for revision
1.0	19-10-2017	BJA	Q17150	Initial preparation of description
1.1	19-3-2018	BJA	Q18150	Amendment of description of operation & sensors <ul style="list-style-type: none"> - Prorunner mk5 Merging belt conveyors <ul style="list-style-type: none"> - Infeed conveyor RDC2 - Infeed conveyor RDC4 - Outfeed conveyor RDC2 - Outfeed conveyor MBC4
1.2	27-7-2018	BJA	Q18150	Update with newest electrical drawings
4.0	29-10-2018	BJA	Q18150	Update encoder functionality and trace functions Update to V4.0 after internal discussions.

2 Table of contents

1	Version control.....	2
2	Table of contents	2
3	Introduction.....	6
3.1	Function of the document	6
3.2	Target audience	6
3.3	Configuration sheet.....	7
4	System description	8
4.1	Basic operation of the Prorunner mk5.....	8
4.2	Module description Prorunner mk5	10
4.3	Level determination modules Prorunner mk5.....	12
4.4	Positioning of carriers Prorunner mk5	13
4.4.1	Encoder functionality	13
4.4.2	Carrier reference position	14
4.4.3	Carrier in area of infeed module	14
4.4.4	Carrier in area of outfeed module	15
4.4.5	Carrier in the area stop in the event of faults/blockages.....	15
4.4.6	Carrier in data transfer area	16
4.5	Starting conveyor run Prorunner mk5	17
4.5.1	Transport by infeed module	17
4.5.2	Transport by outfeed module	19
4.6	Initialisation Prorunner mk5	21
4.6.1	Flowchart initialisation Prorunner mk5	22
4.7	Communication modules Prorunner mk5.....	23
4.7.1	Flow chart communication Prorunner mk5	24

4.8	Chain stretch Prorunner mk5	25
5	Hardware configuration.....	26
5.1	Supported PLC types	26
5.2	Cycle time	26
5.3	Program memory size	26
5.4	TIA Portal version	26
6	Function block main module Prorunner mk5.....	27
6.1	Function block specifications	28
6.2	Flow chart Functionality main module	29
6.3	Components main module Prorunner mk5	30
6.4	Configuration PLC program	31
6.5	Configuration main module Prorunner mk5.....	32
6.6	Inputs main module Prorunner mk5.....	33
6.7	Outputs main module Prorunner mk5.....	34
6.8	Motor interface main module Prorunner mk5	34
6.9	Status main module Prorunner mk5	35
6.10	Faults main module Prorunner mk5	36
7	Function block encoder Prorunner mk5.....	38
7.1	Function block specifications	38
7.2	Flowchart diagram encoder	39
7.3	Inputs encoder	40
7.4	Interfaces encoder	41
8	Function block infeed module RDC2	42
8.1	Function block specifications infeed module RDC2	42
8.2	Components infeed module RDC2.....	43
8.3	Flow chart functionality infeed module RDC2	44
8.4	Configuration infeed module RDC2	45
8.5	Inputs infeed module RDC2	46
8.6	Outputs infeed module RDC2	46
8.7	Motor interface infeed module RDC2.....	47
8.8	Status infeed module RDC2.....	47
8.9	Faults infeed module RDC2	48
9	Function block infeed module RDC4	50
9.1	Function block specifications infeed module RDC4.....	51
9.2	Components infeed module RDC4.....	52
9.4	Flow chart functionality infeed module RDC4	53
9.5	Configuration infeed module RDC4	54
9.6	Inputs infeed module RDC4	55
9.7	Outputs infeed module RDC4	56
9.8	Motor interface infeed module RDC4.....	56
9.9	Status infeed module RDC4.....	57
9.10	Faults infeed module RDC4	58
10	Function block outfeed module RDC2.....	60
10.1	Function block specifications outfeed module RDC2	61
10.2	Components outfeed module RDC2	62
10.3	Flow chart functionality outfeed module RDC2.....	63

10.4	Configuration outfeed module RDC2	64
10.5	Inputs outfeed module RDC2	65
10.6	Outputs outfeed module RDC2	65
10.7	Motor interface outfeed module RDC2	66
10.8	Status outfeed module RDC2	66
10.9	Faults outfeed module RDC2	67
11	Function block outfeed module MBC4	68
11.1	Function block specifications outfeed module MBC4	69
11.2	Components outfeed module MBC4	70
11.4	Flow chart functionality outfeed module MBC4	71
11.5	Configuration outfeed module MBC4	72
11.6	Inputs outfeed module MBC4	73
11.7	Outputs outfeed module MBC4	74
11.8	Motor interface outfeed module MBC4	74
11.9	Status outfeed module MBC4	75
11.10	Faults outfeed module MBC4	76
12	Motor interface	78
Figuur 01:	Main module Prorunner mk5	10
Figuur 02:	Sub-module(s) Prorunner mk5	11
Figuur 03:	Level determination Prorunner mk5	12
Figuur 04:	Carrier reference position	14
Figuur 05:	Carrier in area of infeed module	14
Figuur 06:	Program memory size	26
Figuur 07:	Function block main module	27
Figuur 08:	Components Prorunner mk5 (Main module)	30
Figuur 09:	Function block infeed module RDC2	42
Figuur 10:	Components infeed module RDC2	43
Figuur 11:	Function block infeed module RDC4	50
Figuur 12:	Components infeed module RDC4	52
Figuur 13:	Function block outfeed module RDC2	60
Figuur 14:	Components outfeed module RDC2	62
Figuur 15:	Function block outfeed module MBC4	68
Figuur 16:	Components outfeed module MBC4	70
Tabel 01:	Version control	2
Tabel 02:	PLC types	26
Tabel 03:	Components main module	30
Tabel 04:	Configuration main module	32
Tabel 05:	Inputs main module Prorunner mk5	33
Tabel 06:	Outputs main module Prorunner mk5	34
Tabel 07:	Motor interface main module Prorunner mk5	34
Tabel 08:	Status main module Prorunner mk5	35
Tabel 09:	Faults list main module Prorunner mk5	36
Tabel 10:	Inputs encoder Prorunner mk5	40
Tabel 11:	Interface encoder Prorunner mk5	41
Tabel 12:	Components infeed module RDC2	43
Tabel 13:	Configuration infeed module RDC2	45

Tabel 14:	Inputs infeed module RDC2	46
Tabel 15:	Outputs infeed module RDC2	46
Tabel 16:	Motor interface infeed module RDC2	47
Tabel 17:	Status infeed module RDC2	47
Tabel 18:	Faults list infeed module RDC2	48
Tabel 19:	Components infeed module RDC4	52
Tabel 20:	Configuration infeed module RDC4	54
Tabel 21:	Inputs infeed module RDC4	55
Tabel 22:	Outputs infeed module RDC4	56
Tabel 23:	Motor interface infeed module RDC4	56
Tabel 24:	Status infeed module RDC4	57
Tabel 25:	Faults list infeed module RDC4	58
Tabel 26:	Components outfeed module RDC2	62
Tabel 27:	Configuration outfeed module RDC2	64
Tabel 28:	Inputs outfeed module RDC2	65
Tabel 29:	Outputs outfeed module RDC2	65
Tabel 30:	Motor interface outfeed module RDC2	66
Tabel 31:	Status outfeed module RDC2	66
Tabel 32:	Faults list outfeed module RDC2	67
Tabel 33:	Components outfeed module MBC4	70
Tabel 34:	Configuration outfeed module MBC4	72
Tabel 35:	Inputs outfeed module MBC4	73
Tabel 36:	Outputs outfeed module MBC4	74
Tabel 37:	Motor interface outfeed module MBC4	74
Tabel 38:	Status outfeed module MBC4	75
Tabel 39:	Faults list outfeed module MBC4	76
Tabel 40:	Motor control interface incoming commands → UDT_Motorcontrol_Status	78
Tabel 41:	Motor control interface outgoing commands → UDT_Motorcontrol_Command	78

3 Introduction

Read this chapter carefully before commissioning the software for the Prorunner mk5.

3.1 Function of the document

This document attempts to clarify the functionality, operation and components of the Prorunner mk5 for the purpose of commissioning the software function blocks.

3.2 Target audience

This document is intended for technical staff/engineers with basic knowledge of SIEMENS PLC programming.

3.3 Configuration sheet

The Qimarox configuration sheet has an option to produce a PDF document containing all required settings (Example is showed below)

Prorunner mk5

v4.0

Qimarox

mk5 18XXXXX rev -

ENCODER SOFTWARE SETTINGS

Variable name	Value	Unit	Type
Prorunner.Model	False	-	Bool
Prorunner.Direction	False	-	Bool
Prorunner.ColomnHeight	4700	mm	Real
Prorunner.SupportHeight	72	mm	Real
Prorunner.GearRatio	120,63	-	Real
Prorunner.Speed	0,354	m/s	Real
Prorunner.Carriers	3	-	Int
Prorunner.InfeedLevels	1	-	Int
Prorunner.OutfeedLevels	1	-	Int
Encoder.PulsesPerRevolution	4	-	Int
Options.CarrierObjectDetection	False	-	Bool
Options.CarrierCheckPosition	0	mm	Real
Chain.MinimumLength	3023	mm	Real
Chain.MaximumLength	3073	mm	Real
Chain.TotalLength	8966	mm	Real
ObjectDimension.MaxHeight	400	mm	Real
ObjectDimension.MaxLength	600	mm	Real
ObjectDimension.MinLength	600	mm	Real
ObjectDimension.MaxWidth	400	mm	Real
Automatic.Speed	50,0	Hz	Int
Automatic.Acceleration	1500	ms	Int
Automatic.Deceleration	500	ms	Int
Infeed 1			
Module.Level	1	-	Int
Module.Height	800	mm	Real
Options.LowspeedPickup	False	-	Bool
Conveyor.Speed	0,52	m/s	Real
Conveyor.Length	750	mm	Int
Transport.CycleTime	2,18	s	Real
Transport.Speed	50	Hz	Int
Transport.Acceleration	260	ms	Int
Transport.Deceleration	260	ms	Int
Movement.CycleTime	0	s	Real
Movement.InfeedPosition	0	mm	Int
Movement.WaitingPosition	0	mm	Int
Movement.Speed	0	m/s	Real
Movement.HighSpeed	0	Hz	Int
Movement.LowSpeed	0	Hz	Int
Movement.Acceleration	0	ms	Int
Movement.Deceleration	0	ms	Int

If you don't have this document, please contact Qimarox using the following email address Sales@qimarox.com; Please also mention the Qimarox serial number in the mail

4 System description

This chapter describes the system description of the Prorunner mk5.

Please read this carefully to get a clear picture of what the software function block supports.

4.1 Basic operation of the Prorunner mk5

The position of the carriers of the Prorunner mk5 are tracked precisely by means of an encoder. With each pulse of the encoder, the positions of the carriers are updated with the calculated distance travelled per pulse.

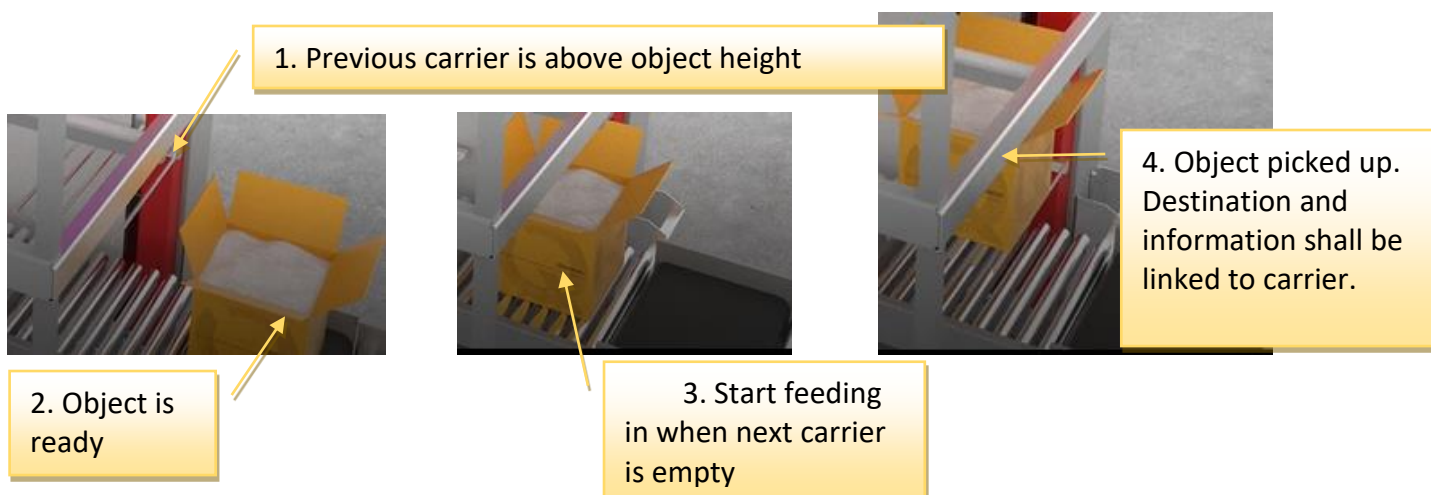
Before you can work with the Prorunner, you must complete the initialisation procedure. This puts all infeed/outfeed modules in a safe/base position. After that, it is checked whether all carriers are present, and their position is determined again.

In automatic mode, Prorunner runs continuously, provided no input/output conveyor blocks the process.

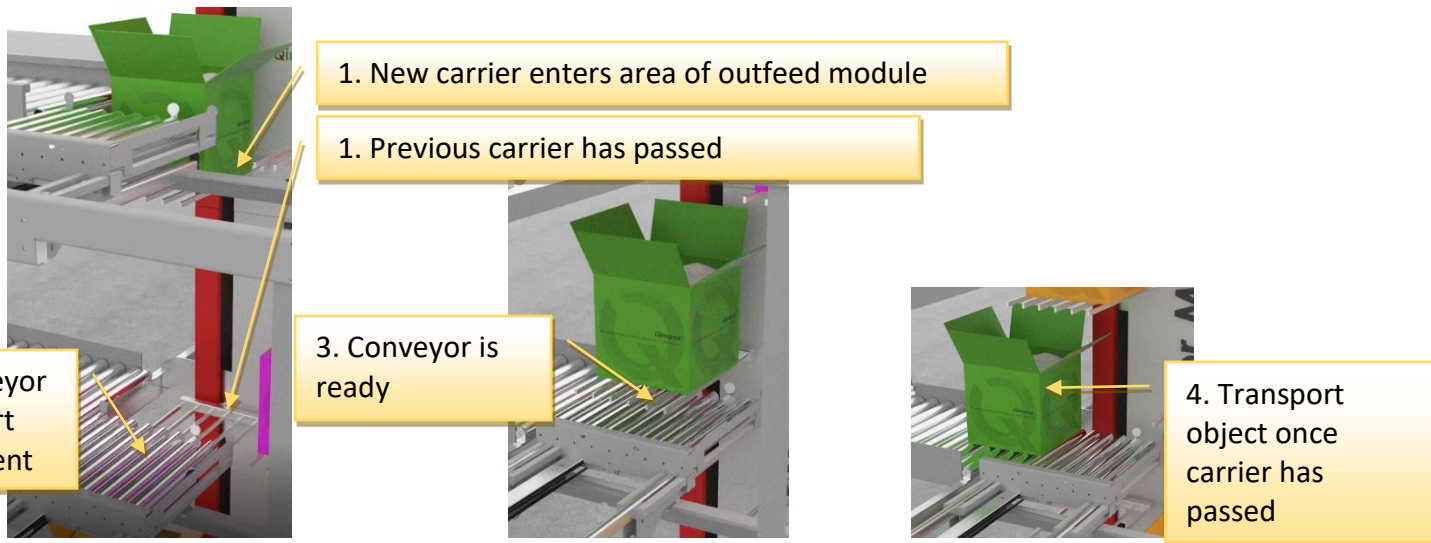
As soon as a carrier enters the area of an infeed conveyor and the carrier is empty, the feeding in of an object can start; this is only done if there is an object present on the infeed conveyor. The Prorunner tells the infeed conveyor how much time is still available to start and complete the infeed process. If the available time is greater than the time required, the infeed may start. As soon as the infeed cycle starts the infeed conveyor will inform the Prorunner, this is done by handshake commands.

When the infeed run is completed, information is made available by the infeed conveyor. This information includes information relating to the desired discharge/outfeed level and other customer-specific information.

If the infeed cycle takes too long, the Prorunner will stop rotating.



This information is pushed along by the Prorunner until the desired discharge level is reached and there, too, this information is made available by the Prorunner to the outfeed transport module.



As soon as a carrier enters the area of an outfeed level, and the object on the carrier is also destined for this level, the available time will be shared by the Prorunner.

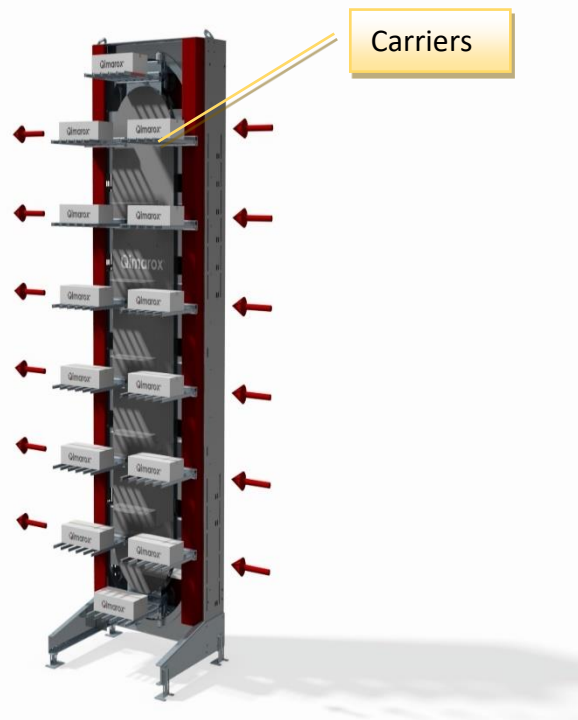
The outfeed transport module can start its cycle if the available time is greater than the required time. As soon as the outfeed transport module detects the object, it will make the information available to the downstream transport. The outfeed module will start feeding the object out. This must be completed before the next carrier enters the hazardous area, if its not completed the Prorunner will stop rotating.

4.2 Module description Prorunner mk5

The software is made up of modules.

There is the main module and at least 2 sub-modules, of which, 1 infeed module and 1 outfeed module. The main module can work with multiple infeed/outfeed modules.

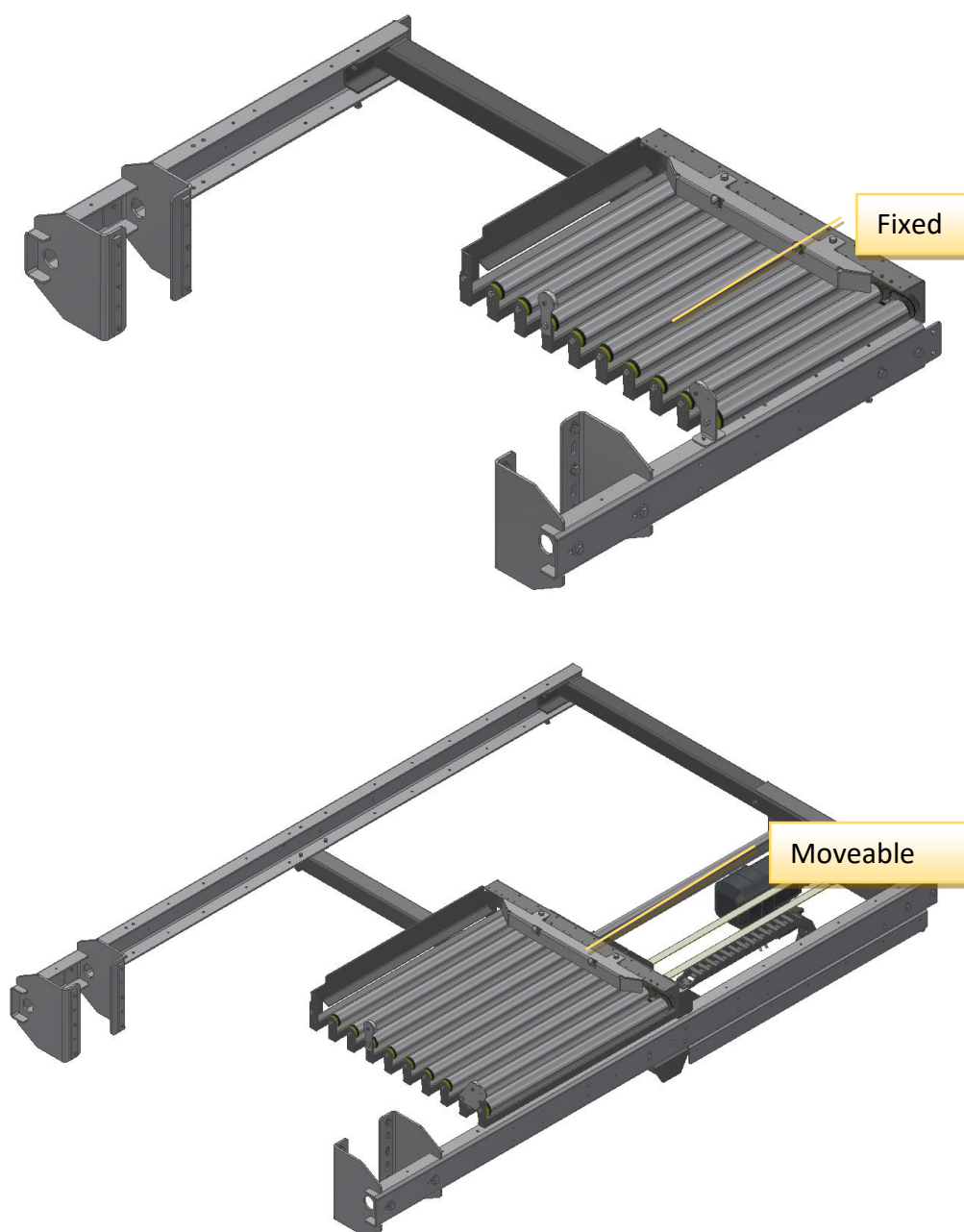
Figuur 01: Main module Prorunner mk5



The function of the main module (Prorunner mk5) is:

- Moving the carriers
- Keeping track of current position of carriers by means of an encoder
- Handling communication with the sub-modules (infeed/outfeed conveyors)
- Protecting the elevator drive Prorunner mk5 + carriers against serious damage

Figuur 02: Sub-module(s) Prorunner mk5

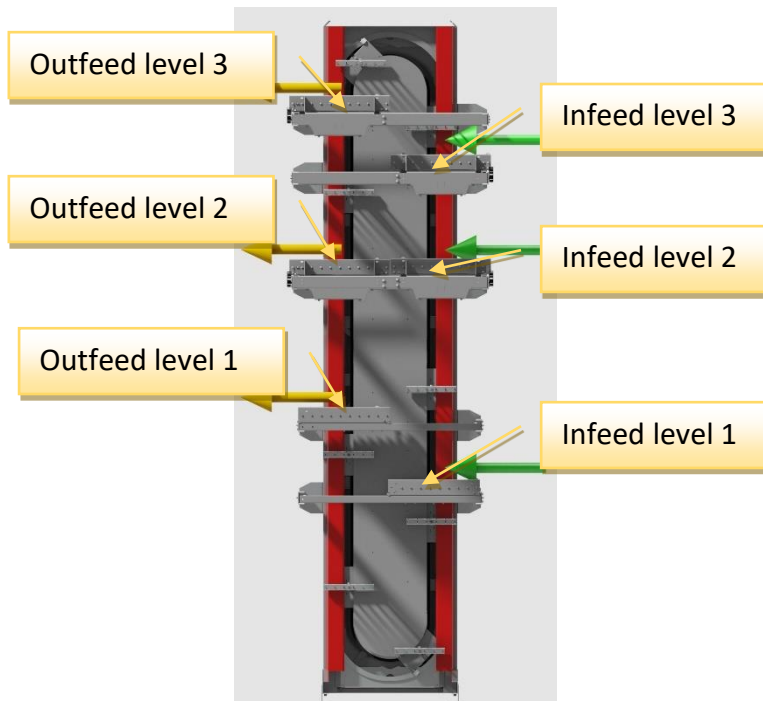


The function of the sub-module(s) (infeed/outfeed conveyors) is:

- Transporting object to desired position
- Handling communication with the main module
- Securing and monitoring its movements and conveyor runs
- Handling communication with upstream/downstream control systems / installations

4.3 Level determination modules Prorunner mk5

Figuur 03: Level determination Prorunner mk5



Infeed/outfeed modules must always be numbered from bottom to top.

Infeed modules are always on one side of the elevator and outfeed modules on the other side.

The transport side can be set for each module. In this case, the following can be set, as shown in the following figure:

4.4 Positioning of carriers Prorunner mk5

4.4.1 Encoder functionality

To track the position of all carriers Qimarox installs by default a 24VDC encoder from SEW type name; EI72.

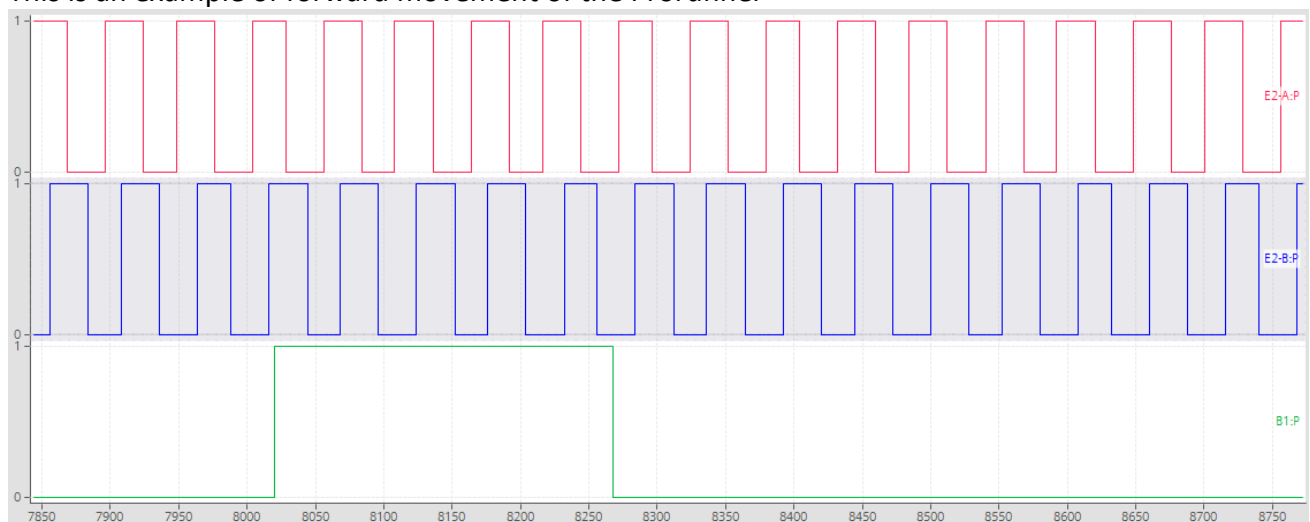
This encoder gives 2 pulses per channel per motor rotation, so in total 4 pulses per motor rotation. We also use the rising/falling edge detection of the encoder pulses, so we have **8 pulses per motor rotation**.

We choose this encoder because these pulses can still be detected by a normal PLC input without the need of a special encoder / highspeed counter hardware. This frequency of the encoder is below 50Hz (20ms) on 1500 rpm.

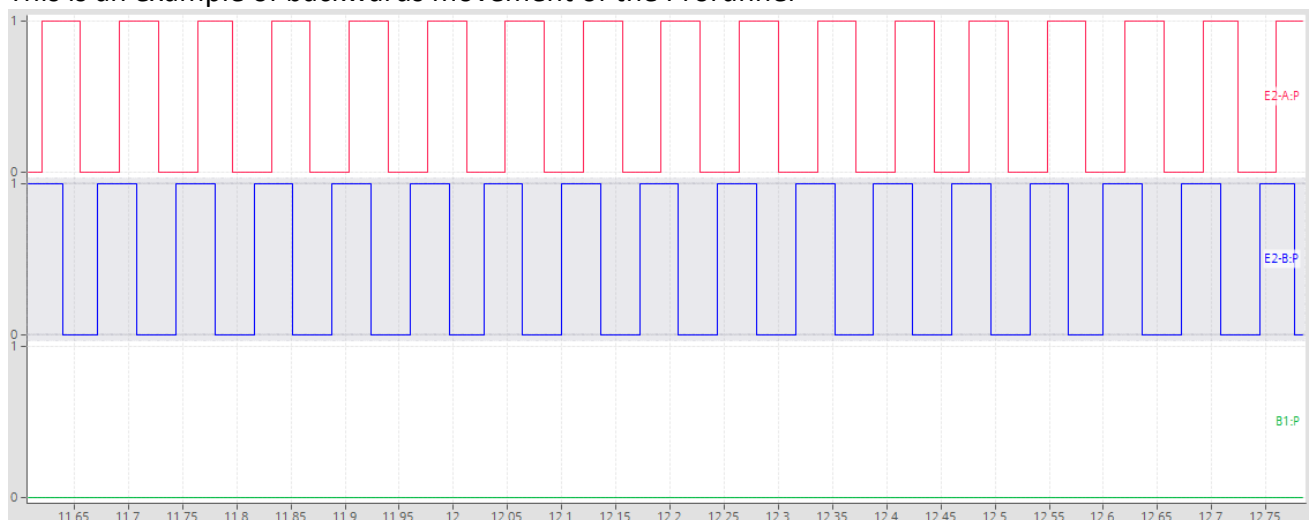
If you like to use an encoder with higher resolution specifications, you are responsible for the position tracking and distance per pulse calculation!!

A normal pulse train look like listed below;

This is an example of forward movement of the Prorunner



This is an example of backwards movement of the Prorunner

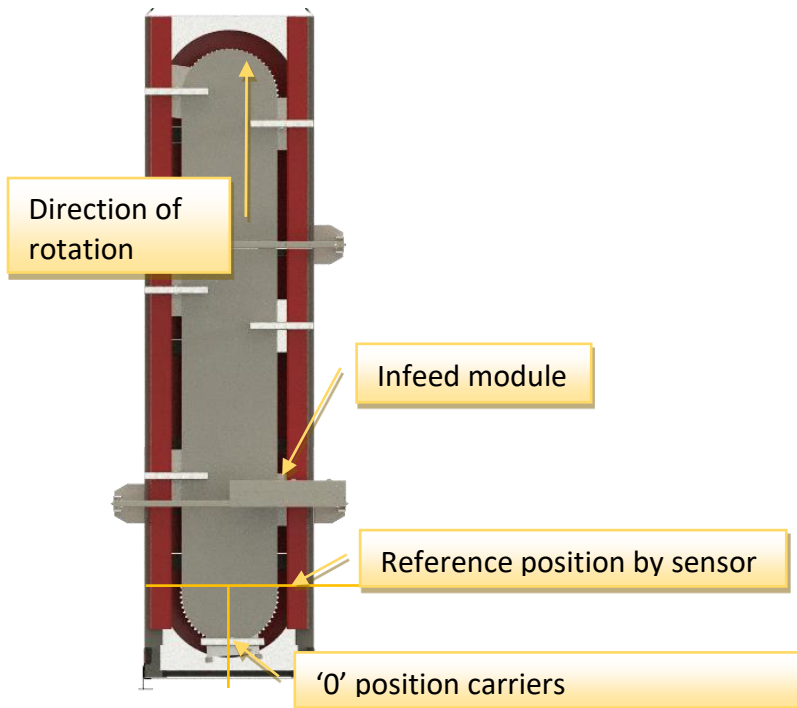


4.4.2 Carrier reference position

The following figure shows the position at which the carrier is aligned with a fixed (calculated) reference. The '0' position is at the heart of the lower chain wheel; it will move slightly as soon as the chain starts to be tightened.

Each carrier is aligned once again after a complete cycle, so the position remains reliable.

Figuur 04: Carrier reference position

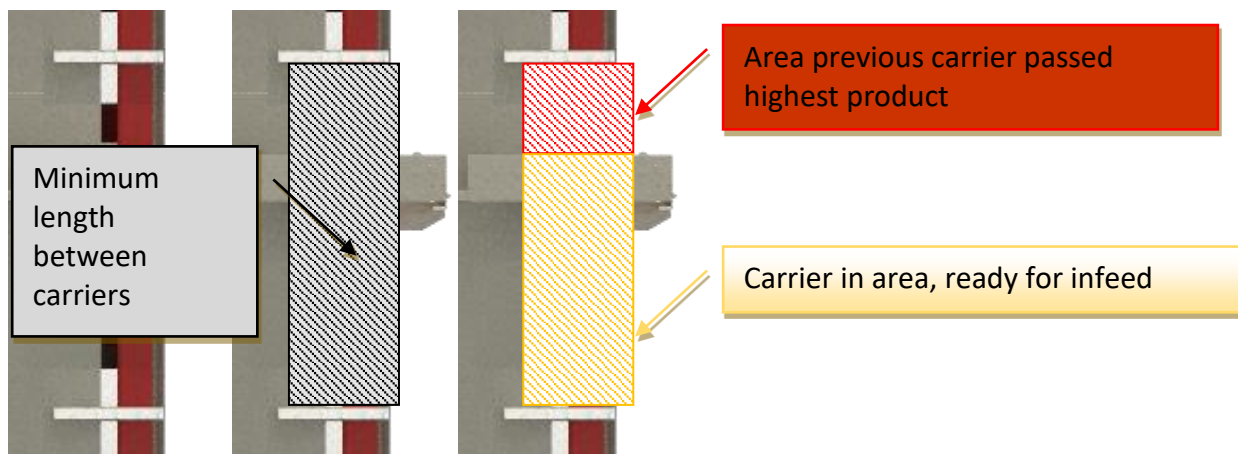


4.4.3 Carrier in area of infeed module

The following figure can be used to determine whether a carrier is within the area of the infeed module.

If a previous carrier is past the highest product point, the next carrier is made available to the infeed module, only when it is empty.

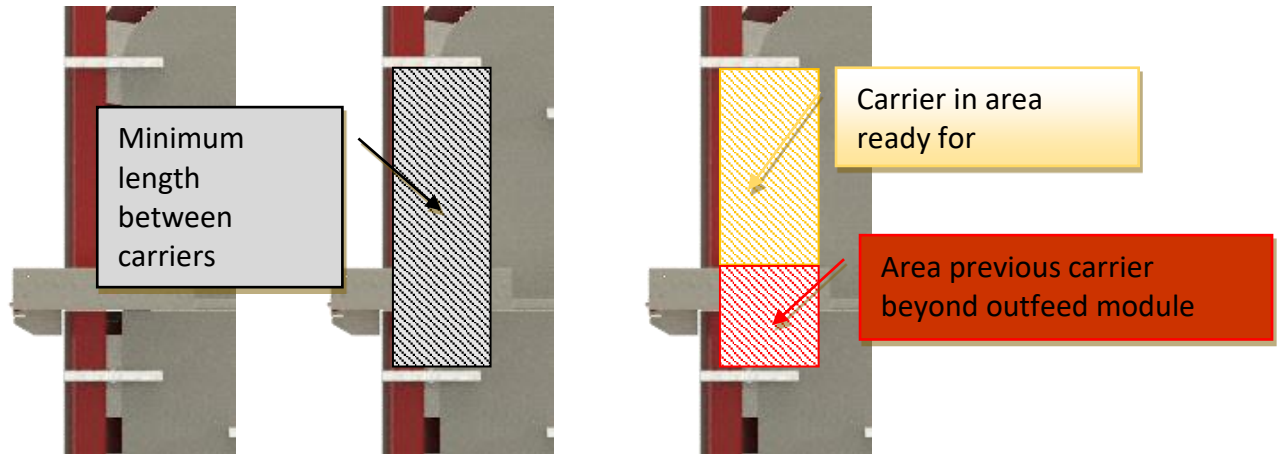
Figuur 05: Carrier in area of infeed module



4.4.4 Carrier in area of outfeed module

The following figure can be used to determine whether a carrier is within the area of the outfeed module.

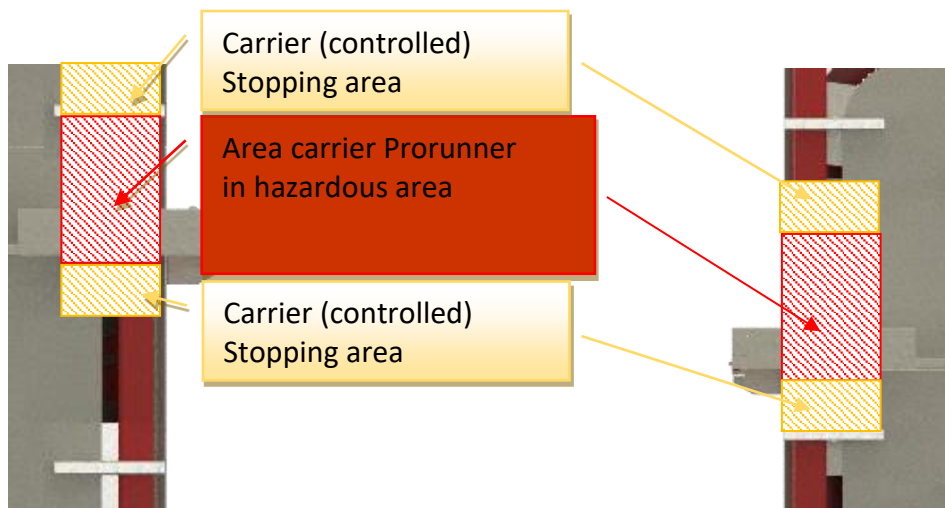
If a carrier is past the outfeed module then information has been transferred



4.4.5 Carrier in the area stop in the event of faults/blockages

If a carrier enters the hazardous area at an infeed/outfeed module and there is a blockage/fault, the carrier must stop. Stopping already happens in the yellow area so that the red area remains free. This allows the Prorunner mk5 to make a controlled stop.

These areas apply to both manual and automatic mode.



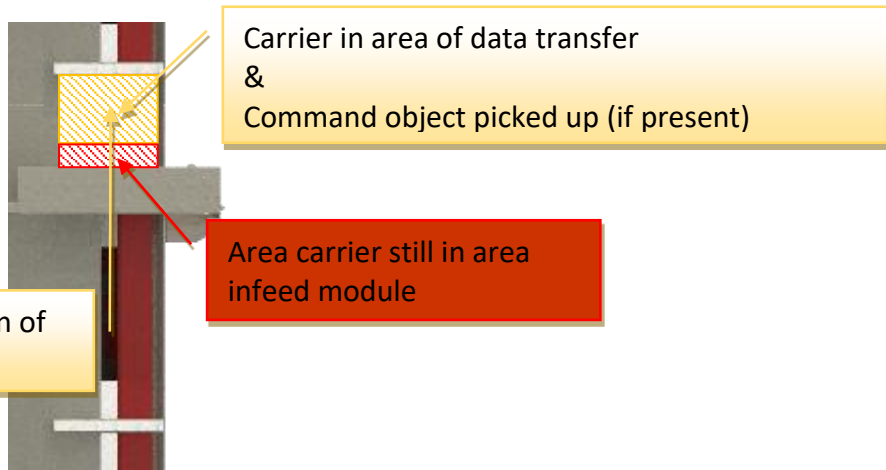
If the PRORUNNER mk5 is blocked by a sub-module, Feedback of the following information is shown; The level number, carrier number is and whether this has been done in manual or automatic mode.

4.4.6 Carrier in data transfer area

As soon as the carrier passes the infeed module (yellow area) and the infeed module has an object ready, the Prorunner will take over information from the infeed module

Information:

- Destination level number
- Source level number
- Installation-dependent information (Optional, to be determined per installation)

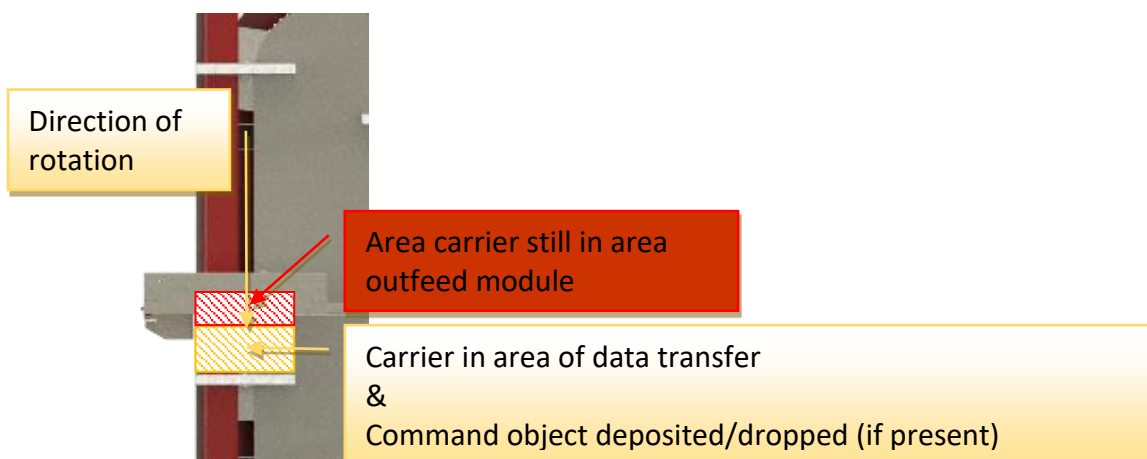


As soon as the carrier has deposited/dropped an object onto the outfeed module, information is transferred when it has passed the outfeed module.

As soon as the outfeed module has copied the information, which is indicated by the communication interface, the Prorunner will delete the information.

Data transfer:

- Destination level number
- Source level number
- Installation-dependent information (Optional, to be determined per installation)



These areas work in both automatic mode and manual mode!

4.5 Starting conveyor run Prorunner mk5

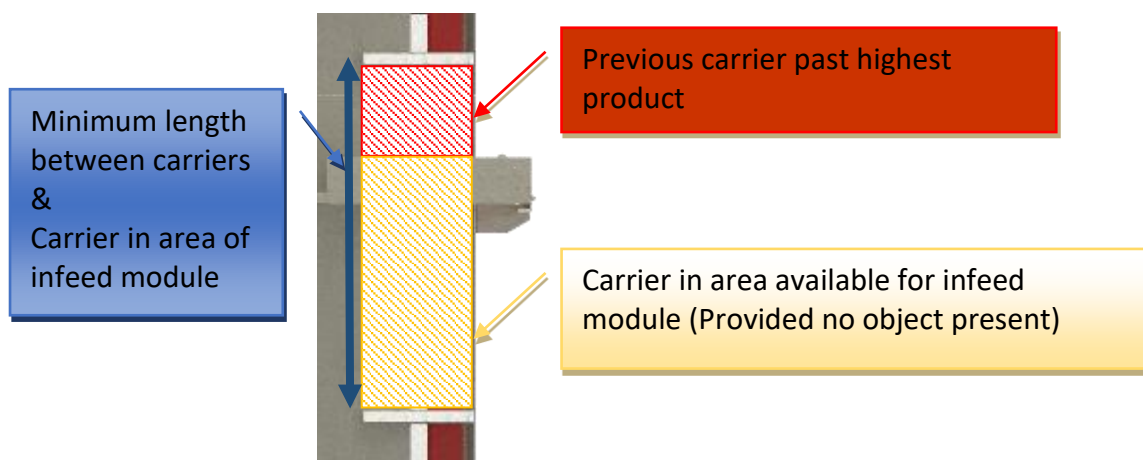
Starting a conveyor run, through the infeed/outfeed sub-modules, depends on the carrier availability of both the Prorunner mk5 and the status of the sub-module. Below it is explained for each module how the availability is determined.

4.5.1 Transport by infeed module

If the infeed module has an object available, it notifies the PRORUNNER mk5 of this by means of the communication interface. Once the previous carrier is past the point of highest object, the remaining length/area of the carrier in the area of the infeed module is determined.

An available time is calculated using this remaining length.

If the time required for the entire infeed procedure is less than the available time, the infeed procedure may be started.



Example:

Minimum length between carriers:	1500mm
Maximum height objects:	400mm
Height of infeed module:	2000mm
Carrier position:	900mm
Speed Prorunner:	0.5m/s

Carrier Prorunner in infeed module area:

Carrier position: < Height of infeed module + Maximum height objects

&

Carrier position > Height of infeed module – (Maximum height objects + Minimum length between carriers)

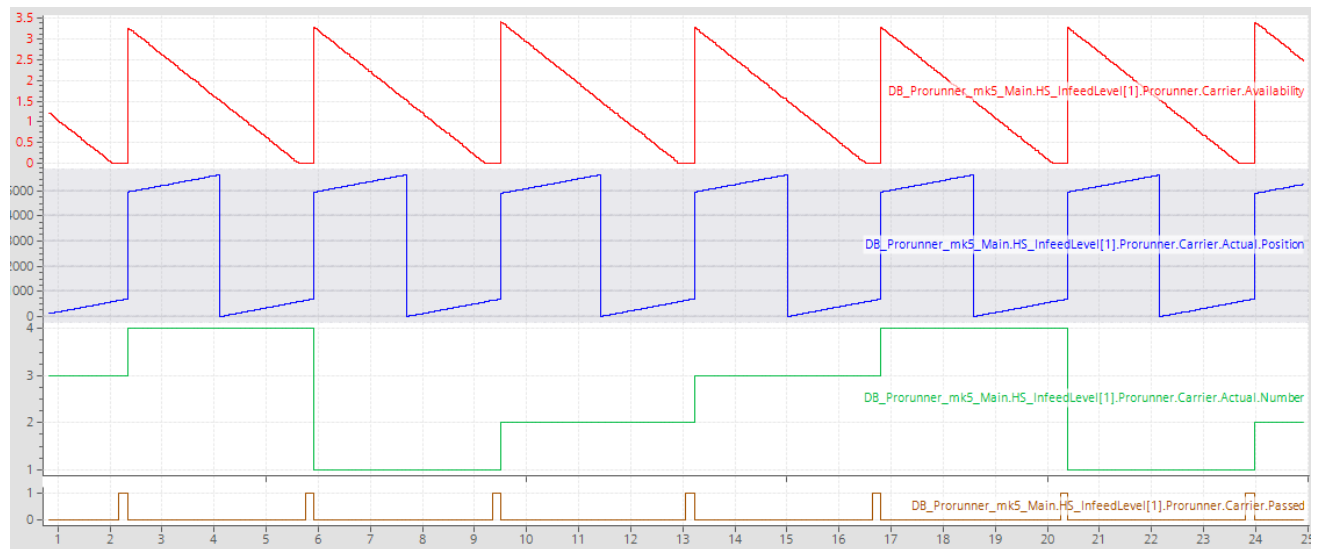
Available time:

$(\text{Height of infeed module} - \text{Carrier position}) / \text{Speed Prorunner}$

$2000 - 900 = 1100 / 0,5 = \underline{2.2s}$

The available time decreases when the Prorunner moves, until it falls below the required time.

On the next page you can find a trace overview of the availability time recorded on a testing module.



- Red line = Available time
- Blue line = Position of carriers
- Green line = Carrier number
- Brown line = Carrier passed infeed module

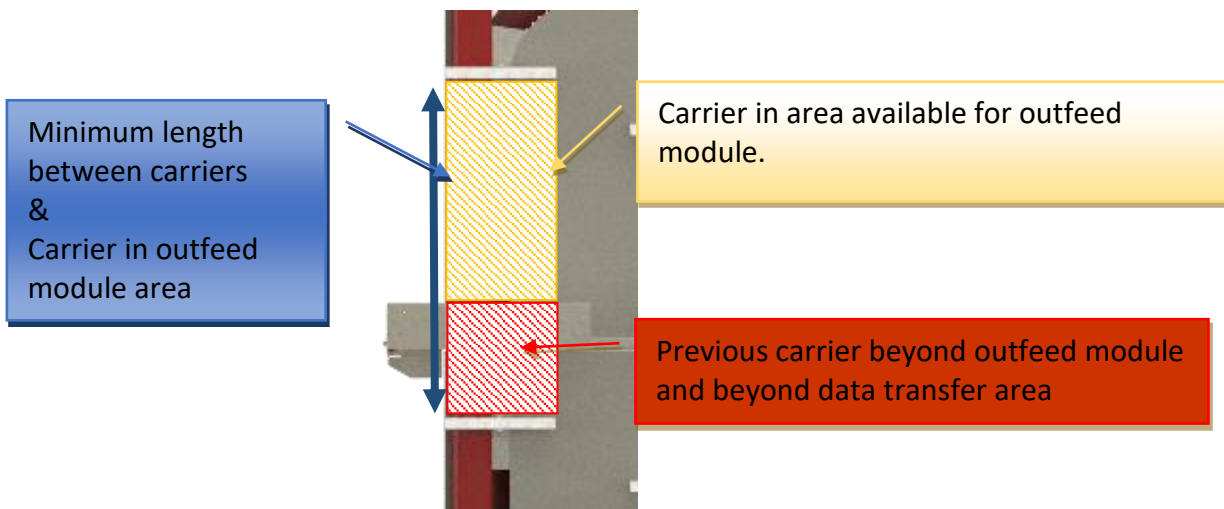
4.5.2 Transport by outfeed module

If the carrier has an object available for the relevant outfeed module, the Prorunner communicates this through the communication interface to the outfeed module. Once the previous carrier is past the point of the highest object, the remaining length/area of the carrier within the area of the outfeed module is determined.

An available time is calculated using this remaining length.

This time does not limit the outfeed module. If an object on the carrier must go to this level, the carrier will wait for the safe position above the outfeed module if it is not yet ready.

The available time only shows whether the outfeed module can complete its procedure/cycle without stopping the Prorunner.



Example:

Minimum length between carriers:	1500mm
Data transfer area:	300mm
Height of infeed module:	8000mm
Carrier position:	6800mm
Speed Prorunner:	0.5m/s

Carrier Prorunner in infeed module area:

Carrier position > Height of infeed module – (Data transfer area – Minimum length between carriers)

&

Carrier position < Height of infeed module + Data transfer area

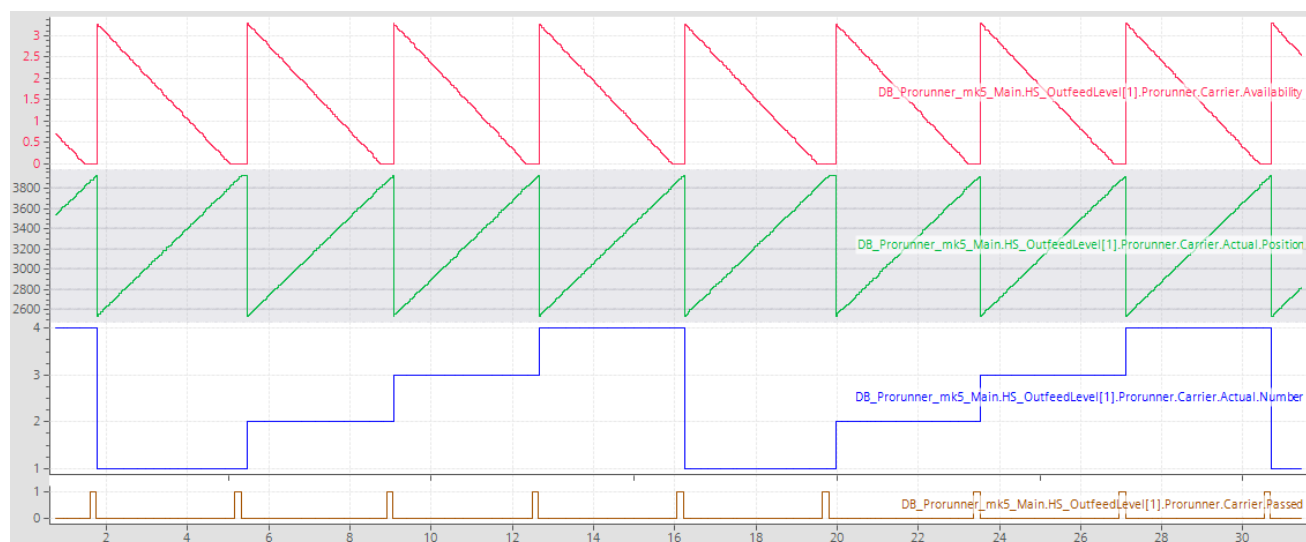
Available time:

$(\text{Height outfeed module} - \text{Carrier position}) / \text{Speed Prorunner}$

$8000 - 6800 = 1200 / 0.5 = \underline{2.4s}$

The available time decreases when the Prorunner moves.

On the next page you can find a trace overview of the availability time recorded on a testing module.



Red line = Available time
Green line = Position of carriers
Blue line = Carrier number
Brown line = Carrier passed infeed module

4.6 Initialisation Prorunner mk5

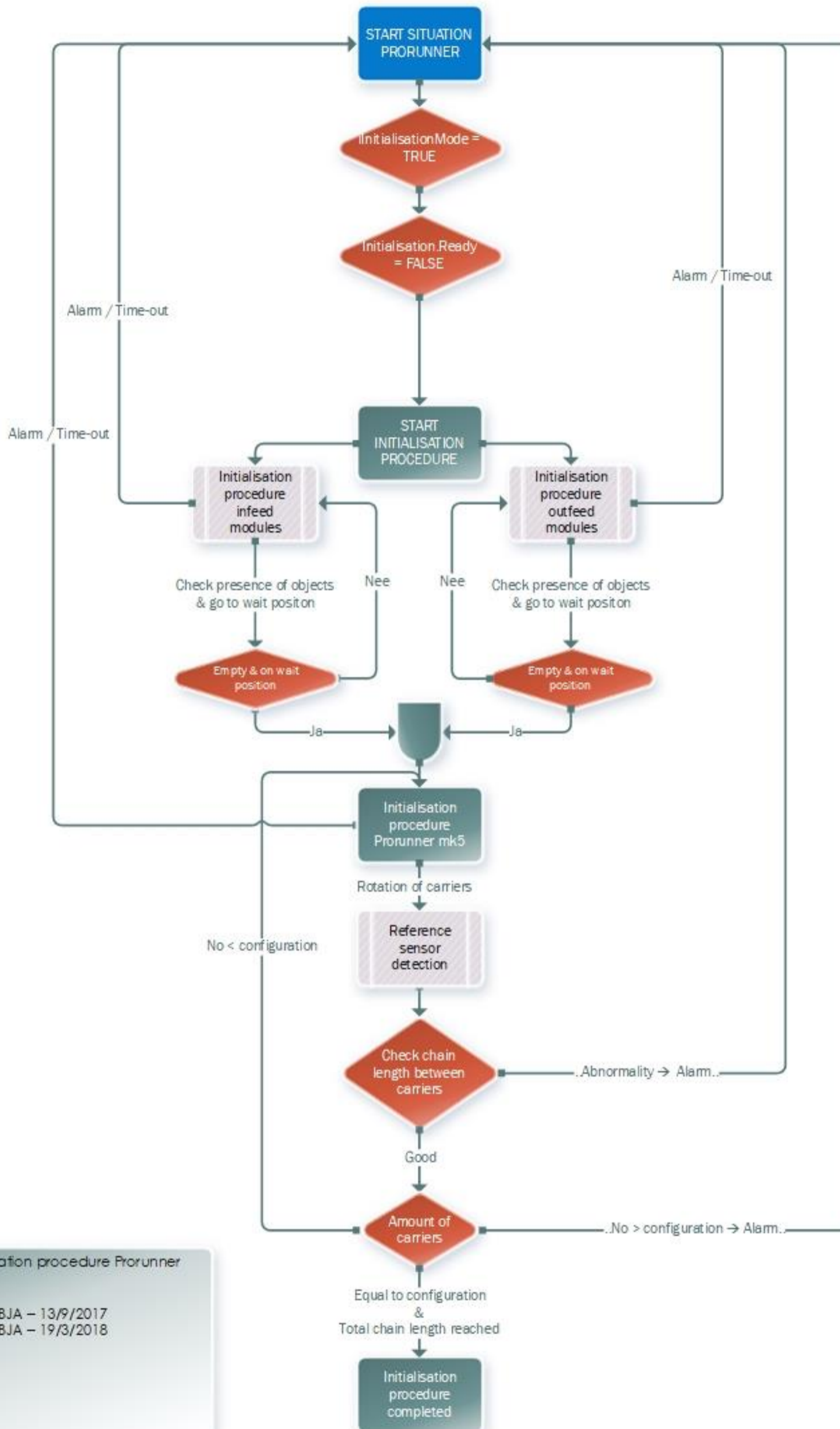
To bring the Prorunner mk5 to the basic position and to ensure that the position of all carriers is reliable, an initialisation procedure must be started.

The initialisation must be carried out:

- Prorunner is switched on.
- Prorunner encoder has proven to be unreliable. Possibilities: Mechanical obstructions, pulses missed, incorrect follow-up of encoder pulses, distance between carriers does not correspond to configuration.

During initialisation, all transport conveyors must be empty and in a safe position. This is checked before the Prorunner can start to run.

4.6.1 Flowchart initialisation Prorunner mk5



4.7 Communication modules Prorunner mk5

To achieve a smooth operation, the main module and sub-modules must communicate with each other. This is done by means of a communication interface.

See the below for the communication overview.

The main module communicates with the encoder and sub-modules.

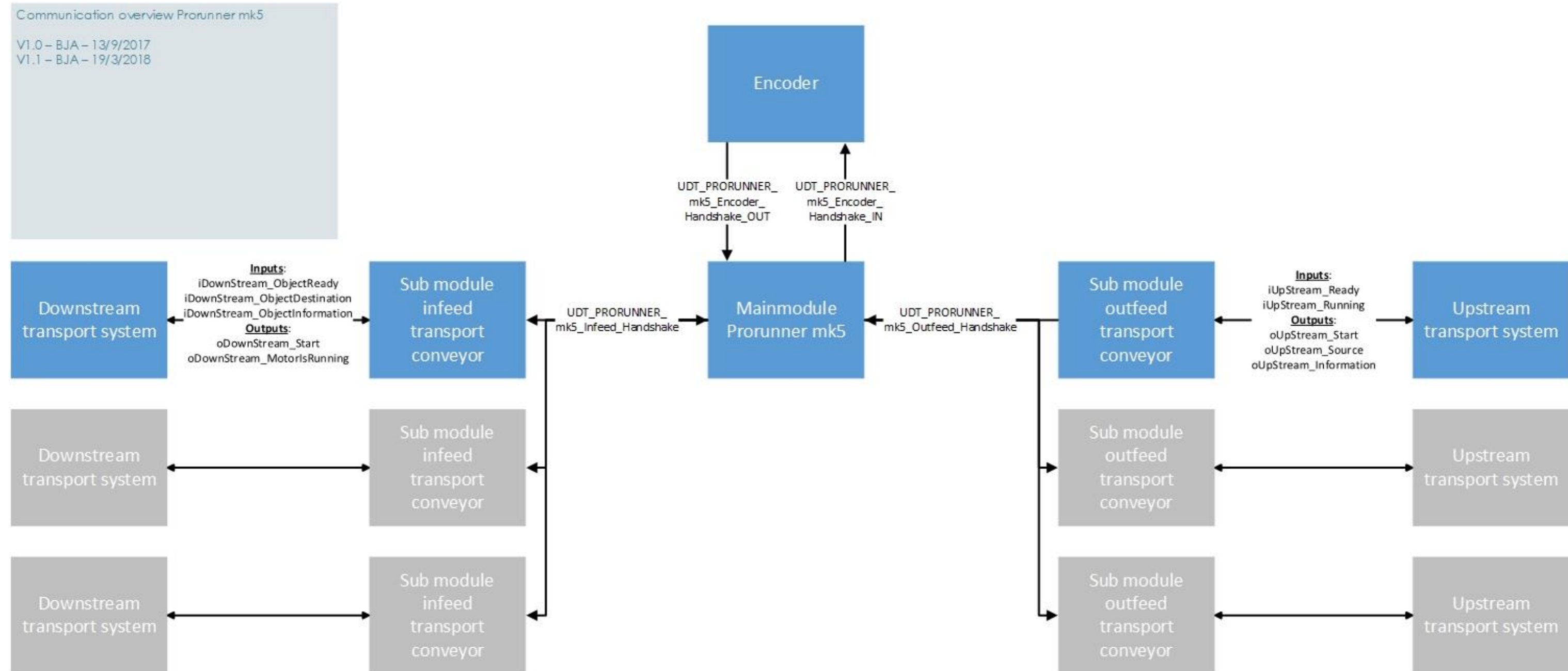
There is a separate software block for the encoder. For the encoder it is important that this is called within certain time limits, otherwise there is a possibility that pulses will be missed. A separate time task is reserved for this, which ensures that the encoder is called every 2 ms.

Communication runs in both directions between the main module and sub-modules in order to determine when objects can be fed in and whether outfeed modules are ready to be received.

Different commands are used to monitor whether sub-modules complete everything within time limits; if not, the main module is stopped.

A sub-module also communicates with the downstream transport system in order to infeed/outfeed products in a controlled way. This communication also determines which destination an object has, and it is possible to send information with it.

4.7.1 Flow chart communication Prorunner mk5



4.8 Chain stretch Prorunner mk5

The chain type used for the mk5 is subject to a chain stretch. This is caused by wear on the chain bearings causing play which results in chain elongation.

To compensate for this, it is determined what the actual distance of a complete cycle is and what the initial total chain length was.

This will produce a chain stretch as a percentage.

Example:

Chain length is originally 6 carriers x 1500mm chain length in between = 9000mm

The encoder counts the actual distance of a complete cycle = 9100mm

Percentage = $(9100/9000 - 1) * 100 = 1.11\%$ stretch

This percentage is offset against the distance travelled per encoder pulse.

With this, the complete round that is made will again correspond to the initial total chain length.

The chain stretch percentage is only calculated when 10 complete rounds have been run. The calculated percentage is the average of these 10 rounds.

5 Hardware configuration

This chapter describes the hardware used to program/test the software function block.

5.1 Supported PLC types

Tabel 02: PLC types

Type	PLC	Article number	Firmware version	Required cycle time
S7-1200	CPU 1212C DC/DC/DC	6ES7 212-1AE40-0XB0	V4.2	7-8 ms. (10 Max.)
S7-1500	CPU 1513-1 PN	6ES7 513-1AL01-0AB0	V2.1	1-2 ms. (6 Max.)

We recommend a standalone 1200 PLC, where the PLC is assigned to the Prorunner installation. This to make sure no fluctuations in cycle time can cause

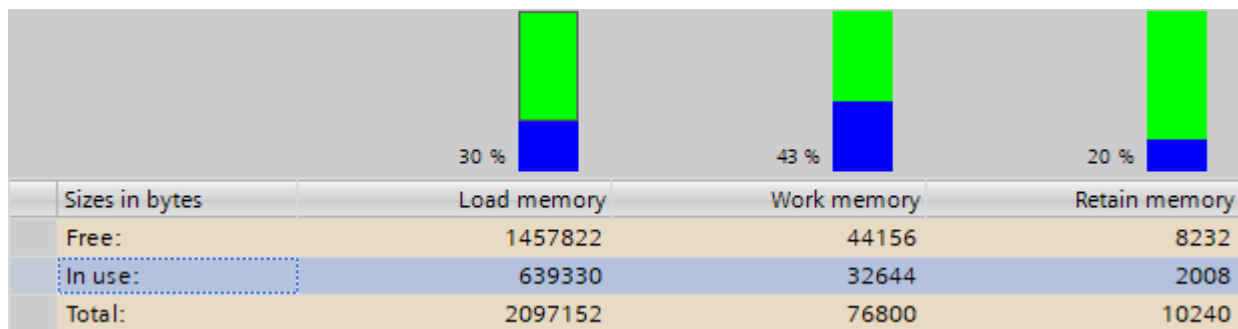
5.2 Cycle time

The required cycle time (see table above) was tested with a setup of 6 racks with 3 infeed modules and 3 outfeed modules. This time will increase with a configuration of more carriers and sub-modules and decrease with less.

5.3 Program memory size

The required memory space for the Prorunner mk5 software is shown below. This is in bytes.

Figuur 06: Program memory size

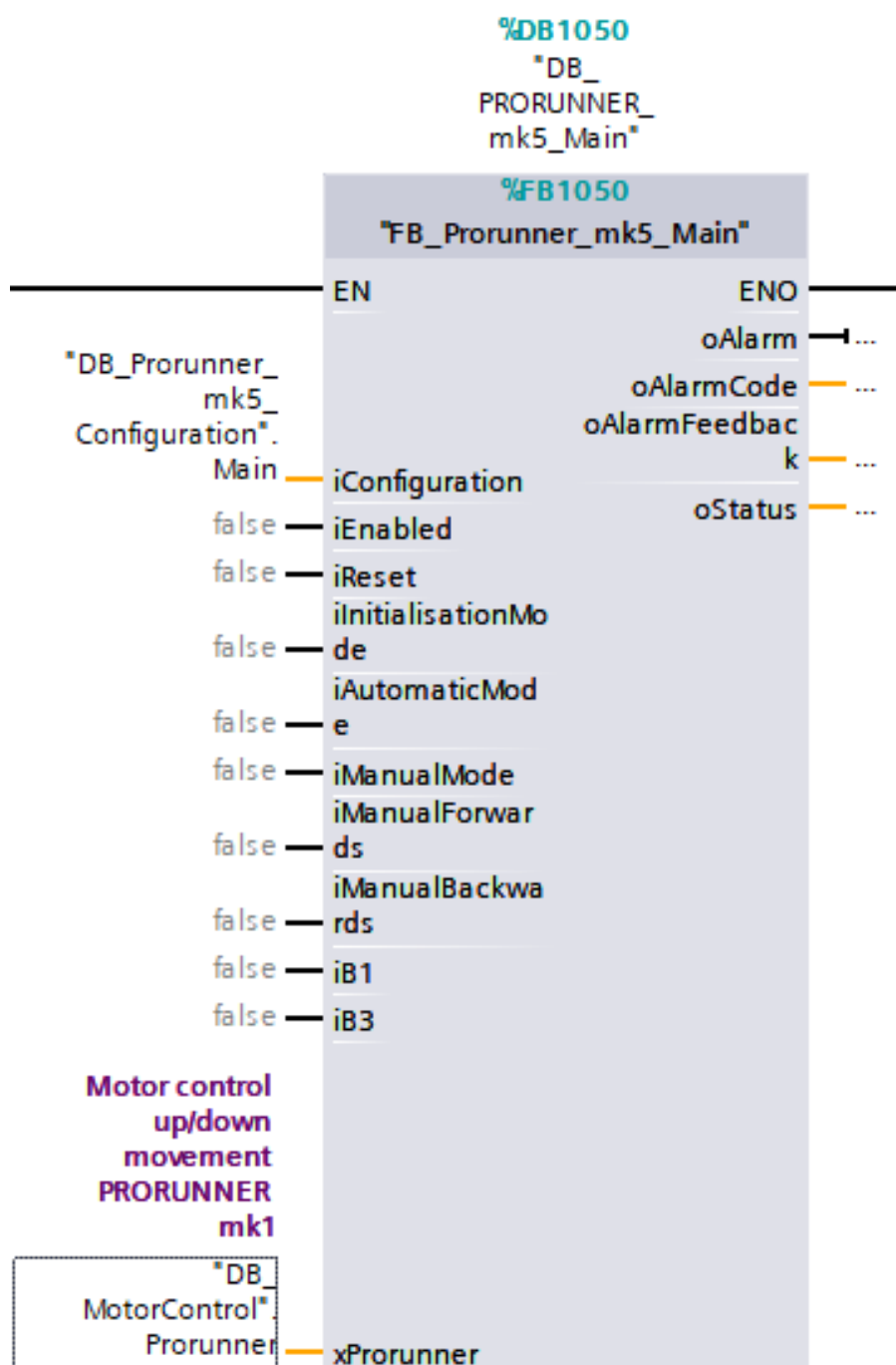


5.4 TIA Portal version

The software is made in TIA Portal V14 + SP1. This is the minimum version that is needed. V14 + SP1 supports the use of SCL networks in a LAD function block. This has been used for the clarity of the software.

6 Function block main module Prorunner mk5

Figuur 07: Function block main module

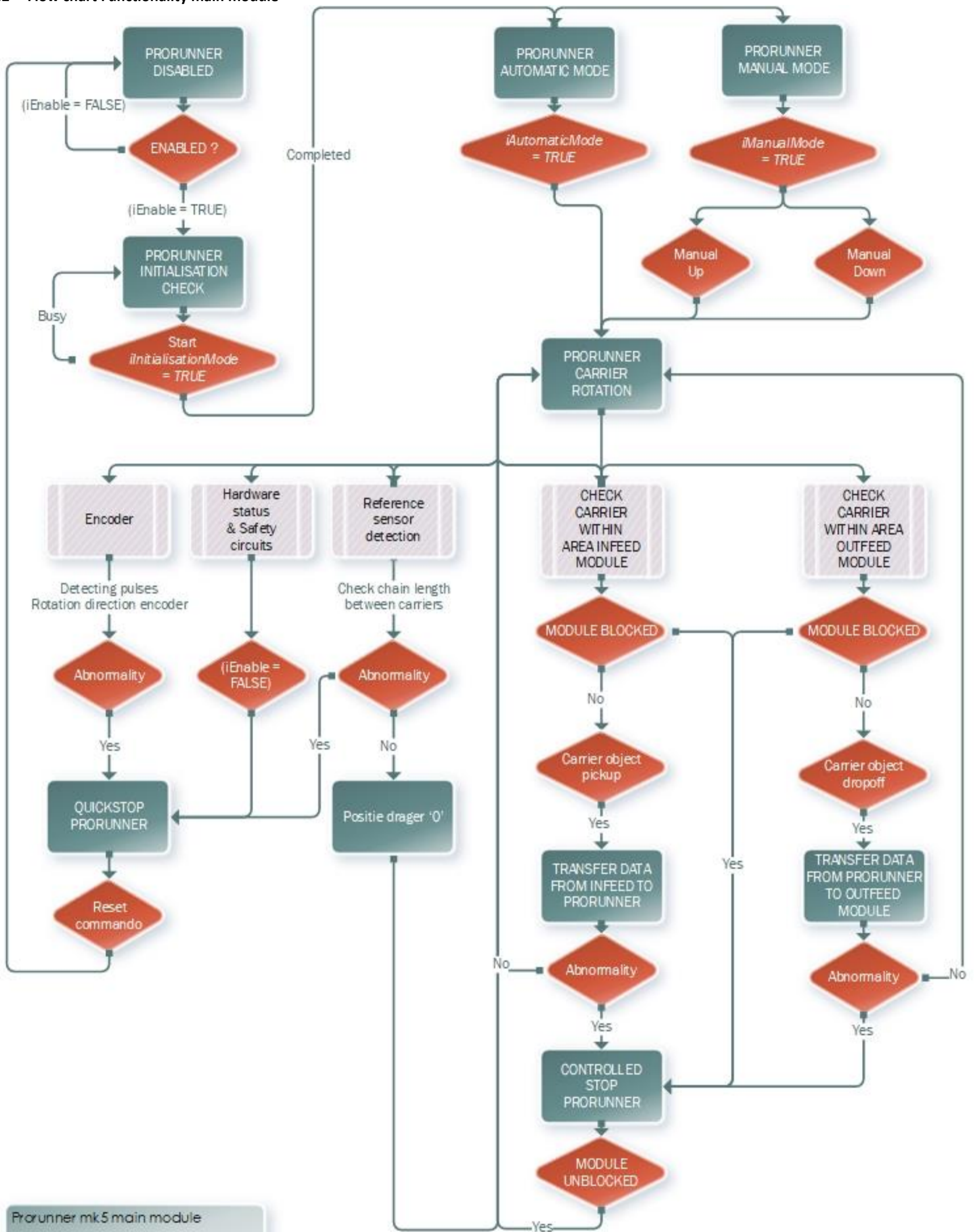


With this software block the main module is called that works as described in Chapter 2.

6.1 Function block specifications

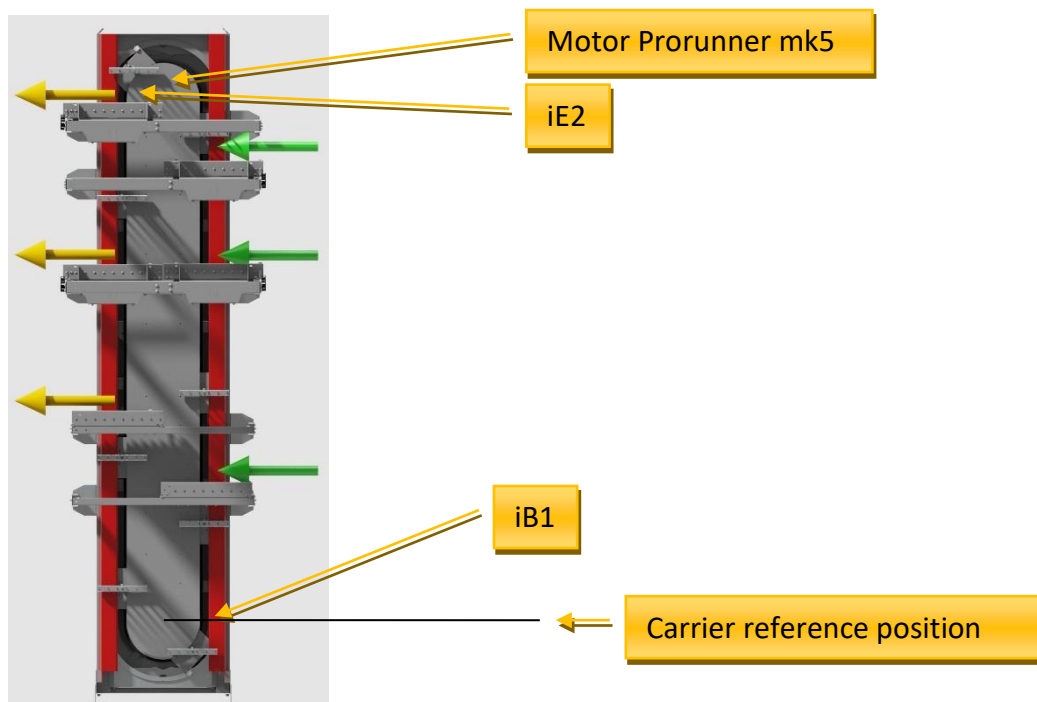
Block name:	FB_PRORUNNER_mk5_Main
Block number:	FB1050
Version:	V1.2
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_MotorControl UDT_Prорunner_mk5_Carrier UDT_Prорunner_mk5_Customer_Information UDT_Prорunner_mk5_Encoder_Configuration UDT_Prорunner_mk5_Encoder_Handshake_IN UDT_Prорunner_mk5_Encoder_Handshake_OUT UDT_Prорunner_mk5_Infeed_Handshake UDT_Prорunner_mk5_Outfeed_Handshake
Function block call	Cyclical (OB1)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

6.2 Flow chart Functionality main module



6.3 Components main module Prorunner mk5

Figuur 08: Components Prorunner mk5 (Main module)



Tabel 03: Components main module

Code		Type
iB1	Sensor Zero / basic position carrier (Position dependent on primary direction of rotation)	IME18-08BPSZC0K
E2	Encoder main module Prorunner	EI72 build in from SEW
iB3	Sensor/ photocell: Product position monitoring. This photocell monitors whether the product has completely entered the infeed roller conveyor.	GL6-P4211 C42-2

6.4 Configuration PLC program

...arox_Library_V14 ▸ Prorunner MK5 ▸ PLC - mk5-XL [CPU 1212C DC/DC/DC] ▸ PLC tags ▸ MK5 Configuration [5]

Tags User constants

MK5 Configuration

	Name	Data type	Value	Comment
1	cCarriers	Int	20	Maximum number of carriers for the eleva...
2	cInfeedLevels	Int	10	Maximum number of infeedlevels for the ...
3	cOutfeedLevels	Int	10	Maximum number of outfeed levels for th...
4	cPassed_Area	Real	50.0	Distance window of carrier after passing le.
5	cOB_1050_Cyclic_Interrupt	Int	2	Cyclic interrupt of OB1050 (default = 2ms...
6	<Add new>			

cCarriers: Maximum number of carriers (Standard 20)
 cInfeedlevels: Maximum number of infeed levels (Standard 5)
 cOutfeedlevels: Maximum number of infeed levels (Standard 5)
 cPassed_Area: Area carrier beyond module
 cOB_1050_Cyclic_Interrupt: Setting time interruption PLC OB block (Standard 2ms.)

6.5 Configuration main module Prorunner mk5

The configuration consists of one UDT → UDT_Prorunner_mk5_Encoder_Configuration.

Everything can/will be determined from the Qimarox configuration sheet.

We strongly recommend using the PDF configuration sheets provided by Qimarox for this.

Tabel 04: Configuration main module

Naming	Data type	Unit	Description
Prorunner.Type	Bool	0/1	0=Standard, 1=XL
Prorunner.Direction	Bool	0/1	Direction of rotation of lift, seen from front 0=CW, 1=CCW
Prorunner.ColomnHeight	Real	mm.	Total height of the Prorunner
Prorunner.SupportHeight	Real	mm.	Height between floor and bottom of Prorunner
Prorunner.GearRation	Real	i	Transmission ratio motor/ reducer
Prorunner.Speed	Real	m/s.	Speed of conveyor track at rated actuation (metre per second)
Prorunner.Carriers	Int	0..99	Number of carriers present on Prorunner mk5
Prorunner.InfeedLevels	Int	0..99	Number of infeed sub-modules / levels
Prorunner.OutfeedLevels	Int	0..99	Number of outfeed sub-modules / levels
Encoder.PulsesPerRevolution	Int	1..1024	Number of pulses per motor revolution (EI72 = 4)
Options.CarrierObjectDetection	Bool	0/1	Monitoring of empty carriers at checking position.
Options.CarrierCheckPosition	Real	mm.	Checking position object detection on carrier (checking by means of sensor iB3)
Chain.MinimumLength	Real	mm.	Minimum chain length between carriers
Chain.MaximumLength	Real	mm.	Maximum chain length between carriers
Chain.TotalLength	Real	mm.	Total chain length all carriers
ObjectDimension.MaxHeight	Real	mm.	Maximum height of objects
ObjectDimension.MaxLength	Real	mm.	Maximum length of objects
ObjectDimension.MaxWidth	Real	mm.	Maximum width of objects
Automatic.Speed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Automatic.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Automatic.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill

6.6 Inputs main module Prorunner mk5

Inputs are required for the system to function properly. Here the required sensors are declared, and the status of the installation.

Tabel 05: Inputs main module Prorunner mk5

Naming	Data type	Unit	Description
iConfiguration	UDT		UDT_Prorunner_mk5_Encoder_Configuration UDT as discussed in section below. This contains all requirements/settings for the installation to function properly
iEnabled	Bool	0/1	System is switched on. Safety functions are operational, hardware has no problems, and power supplies to any controllers are switched on. 1 = Installation switched on
iReset	Bool	0/1	Reset command to reset faults.
iInitialisationMode	Bool	0/1	Start of initialisation procedure. Must remain present until the initialisation procedure is completed. Automatic / Manual mode is being ignored. 1=Initialisation is active
iAutomaticMode	Bool	0/1	Installation is in automatic mode. The direction of rotation of the PRORUNNER is only in the primary direction.
iManualMode	Bool	0/1	Installation is in manual mode. Manual commands influence the direction of rotation.
iManualUp	Bool	0/1	Command for upward movement (seen from input modules) in manual mode
iManualDown	Bool	0/1	Command for downward movement (seen from input modules) in manual mode
iB1	Bool	0/1	Carrier reference position
iB3	Bool	0/1	Empty carriers message by means of photocell detection (optional)

6.7 Outputs main module Prorunner mk5

Outputs must be declared by the customer.

Outputs provide feedback such as faults, status and communication.

Tabel 06: Outputs main module Prorunner mk5

Naming	Data type	Unit	Description
oAlarm	Bool	0/1	Fault active
oAlarmCode	Int	0	Active fault code (See chapter 'Faults' for the correct overview)
oAlarmFeedback	Word	0	Active fault feedback for HMI. One bit is reserved for each fault. Fault 1 = bit 0, fault 2 = bit 1, etc.
oStatus	Int	0	See chapter 'Status Prorunner mk5' for further information.

6.8 Motor interface main module Prorunner mk5

One motor is used for the main module. All commands are prepared via this motor interface to handle different forms of motor actuation.

For more information, please refer to chapter '12 - Motor interface'

We recommend carrying out the motor in conjunction with a frequency controller because there are different requirements for actuation:

- Mechanism is less loaded if engine starts slower by means of acceleration
- Quickstop must be possible $\leq 100\text{ms}$.
Quickstop = quickest possible stop with pre-defined deceleration in controller
- Energize must be possible (This is not a requirement, but more an extra function)
Energize = Motor must be capable of being energized without moving; this is used to keep the lift in place if the brake is not functioning properly.

The software block has been tested with a SEW frequency controller: → Movitrac B → MC07B0015-4-5A3-S0

This one used a 1.5kW motor. Communication can be done through S-Bus or I/O

Tabel 07: Motor interface main module Prorunner mk5

Naming	Data type	Unit	Description
xProrunner	UDT		UDT_MotorControl Motor interface Prorunner

6.9 Status main module Prorunner mk5

The status of the Prorunner is output, giving the user a clear picture of the status/movements of the Prorunner mk5.

Tabel 08: Status main module Prorunner mk5

Code	Description
0	NO ACTION / SWITCHED OFF: Prorunner switched off / no mode active
1	FAULT: Prorunner blocked by fault
2	WAITING: Prorunner ready for use / start
3	MANUAL OPERATION / MANUAL MODE: Prorunner in manual mode
4	TURNING / AUTOMATIC MODE: Prorunner moving and ready
5	EMPTY SIGNAL: PRORUNNER waits at infeed level until infeed conveyor indicates that an object is ready for infeeding.
6	STOPPED DUE TO INFEED TIME PERIOD OR BLOCKING OF INFEED MODULE: Prorunner waits for safe position <u>under or above</u> the relevant infeed module until the infeed module indicates it has an object ready for picking up or is no longer blocked.
7	STOPPED DUE TO OUTFEED MODULE REMAINING BUSY OR BLOCKAGE OUTFEED MODULE: Prorunner waits for safe position <u>above or under</u> the relevant outfeed module until outfeed module indicates it no longer has an object or is no longer blocked.
10	INITIALISATION PROCEDURE ACTIVE Prorunner busy with initialisation procedure.
11	INITIALISATION PROCEDURE READY Prorunner ready with initialisation procedure.

6.10 Faults main module Prorunner mk5

The software function block issues an outfeed signal and a fault code in the event of a fault.

With the outfeed signal 'oAlarm' it is possible to indicate a fault by, for example, switching on a red lamp. With the fault code 'oAlarmCode' you can indicate this better.

With the fault word 'oAlarmFeedback' you can generate a clearly defined fault message by means of an HMI. The faults are divided over 16 bits. Code 1 = bit 0 ... Code 16 = bit 15.

Below is the list of faults that may be active with explanation/result and solution.

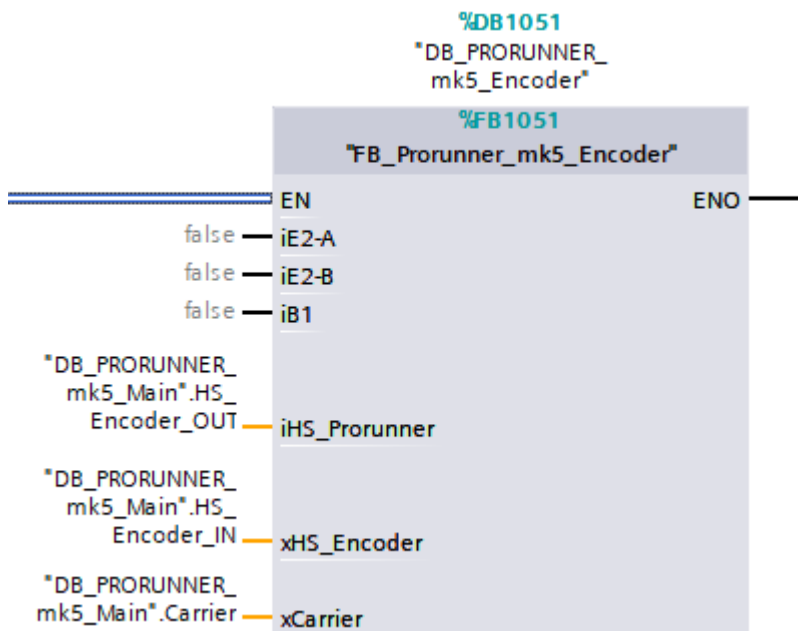
Some faults are only possible when selecting the relevant options.

Tabel 09: Faults list main module Prorunner mk5

Code	Description / Cause	Effect	Solution
5	Movement of Prorunner blocked. Encoder detects no pulses within maximum time limits.	Prorunner is blocked. Quickstop functionality is activated (if possible) Initialisation must be carried out again.	Check Prorunner for blockages, if no blockages can be determined: Check the operation/adjustment encoder. Check whether encoder function block is called fast enough on the PLC; this must be $\leq 2\text{ms}$. Give the reset command to reset the fault (iReset).
6	Prorunner moving in wrong direction; the encoder detects incorrect direction of rotation.	Prorunner is blocked. Quickstop functionality is activated (if possible) Initialisation must be carried out again.	Check direction of rotation encoder Prorunner. Give the reset command to reset the fault (iReset).
7	Movement of PRORUNNER seen while motor not being controlled.	Prorunner is blocked. Energize functionality is activated (if possible) Initialisation must be carried out again.	Check mechanical adjustment (Rem) PRORUNNER. Give the reset command to reset the fault (iReset).
8	Configuration is not entered correctly	Prorunner is blocked. Initialisation is not possible.	Check configuration settings. Make sure that everything is entered correctly and not below the minimum value.
10	Carrier not seen within maximum distance margin between carriers.	Prorunner is blocked. Quickstop functionality is activated (if possible) Initialisation must be carried out again.	Check settings chain lengths in the configuration; check this against the actual distance between the carriers. Check mechanical tolerances of the carrier re the reference sensor. Give the reset command to reset the fault.

Code	Description / Cause	Effect	Solution
11	Unknown carrier reported. Distance between carriers below the minimum distance margin between carriers	Prorunner is blocked. Quickstop functionality is activated (if possible) Initialisation must be carried out again.	Check settings chain lengths in the configuration; check this against the actual distance between the carriers. Check sensor i = MM-B1, since sensor was triggered earlier than usual. Give the reset command to reset the fault.
12	Carrier not empty, during inspection in combination with sensor iB3.	Automatic operation is stopped.	Remove object from carrier. Give the reset command to reset the fault.
13	Fault controller/inverter PRORUNNER.	Automatic operation is stopped.	Check operation/fault controller/drive. Make sure that it is ready for operation again. Give the reset command to reset drive fault. Wait until the controller is ready again.
14	Initialisation necessary	Prorunner is blocked.	Perform initialisation
15	Initialisation procedure TIMEOUT: Procedure takes longer than the total chain length with +50% margin!	Initialisation has not been completed. Automatic / Manual mode is not possible!	Check status sub infeed/outfeed modules, and restart procedure
16	Initialisation procedure ERROR: <ul style="list-style-type: none"> Not all carriers seen by sensor iB1 Chain length between carriers is too short Chain length between carriers is too long Total chain length is incorrect 	Initialisation has not been completed. Automatic / Manual mode is not possible!	Check adjustment sensor iB1 PRORUNNER. Check configuration PRORUNNER. Restart procedure by giving initialisation command again.

7 Function block encoder Prorunner mk5

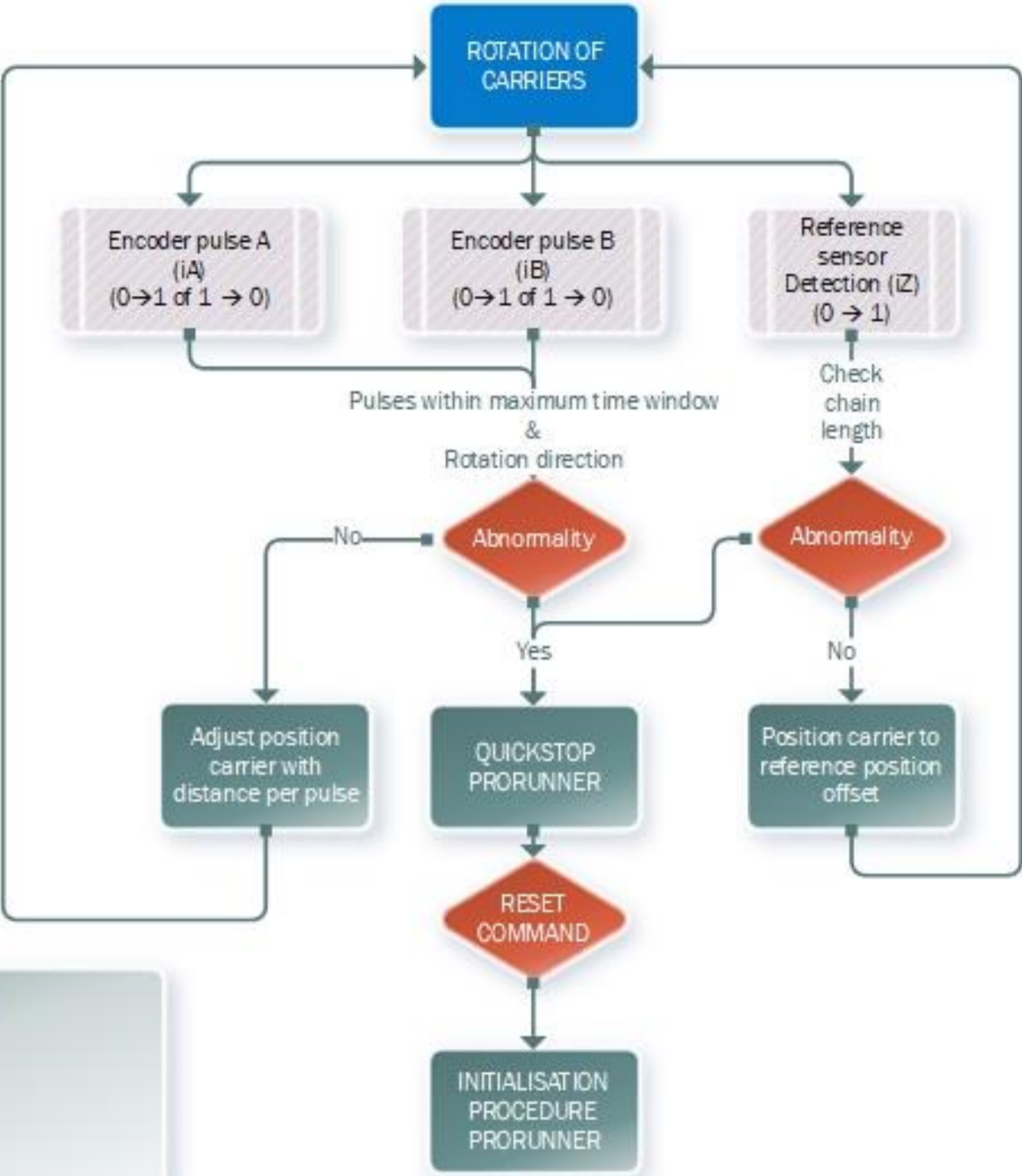


With this software block, the encoder processing is called up. This block must be specially called, because of the high/higher frequency of the encoder. This must be approximately 2ms.

7.1 Function block specifications

Block name:	FB_Prорunner_mk5_Encoder
Block number:	FB1051
Version:	V1.0
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_Prорunner_mk5_Carrier UDT_Prорunner_mk5_Customer_Information UDT_Prорunner_mk5_Encoder_Handshake_IN UDT_Prорunner_mk5_Encoder_Handshake_OUT
Function block call	Time interrupt 2ms (OB1050 – Time interrupt)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

7.2 Flowchart diagram encoder



Prorunner mk5 Encoder
V1.0 – BJA – 18/9/2017
V1.1 – BJA – 23/3/2018

7.3 Inputs encoder

Inputs are required for the system to function properly. The required sensors and the communication interface with the Prorunner are declared here.

Some inputs must be called via peripheral addressing. Explanation can be found here: → [Siemens - Where and when do you need peripheral addressing?](#)

Tabel 10: Inputs encoder Prorunner mk5

Naming	Data type	Unit	Description
iE2-A	Bool	0/1	Pulse A encoder - for this, always use the input from the input periphery. Example: 'i=MM-E2-A':P This queries the actual status of the input instead of the saved status of the input in the normal cycle of the PLC. Adding the ':P' takes care of this.
iE2-B	Bool	0/1	Pulse B encoder - for this, always use the input from the input periphery. Example: 'i=MM-E2-B':P This queries the actual status of the input instead of the saved status of the input in the normal cycle of the PLC. Adding the ':P' takes care of this.
iB1	Bool	0/1	Carrier reference position - for this, always use the input from the input periphery. Example: 'i=MM-B1':P This queries the actual status of the input instead of the saved status of the input in the normal cycle of the PLC. Adding the ':P' takes care of this.
iHS_Prорunner	UDT		UDT_Prорunner_mk5_Encoder_Handshake_OUT This is the communication interface (HandShakes) of PRORUNNER → Encoder. This is a direct reference to the DB of the main module.

7.4 Interfaces encoder

With the interface, information is exchanged between encoder and main module.

These interfaces are used for:

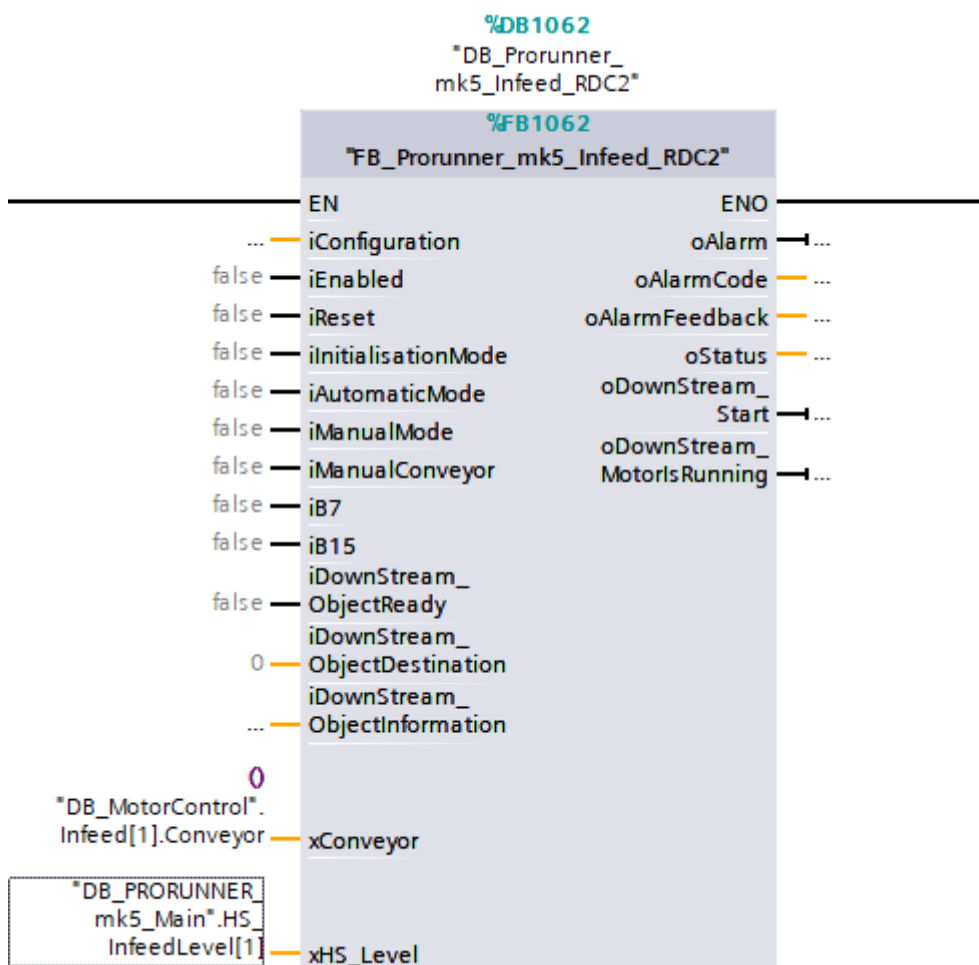
- Communication encoder \leftrightarrow Prorunner
- Updating carriers PRORUNNER

Tabel 11: Interface encoder Prorunner mk5

Naming	Data type	Unit	Description
xHS_Encoder	UDT		UDT_Prorunner_mk5_Encoder_Handshake_IN This is the communication interface (HandShakes) of Encoder \rightarrow PRORUNNER. This is a direct reference to the DB of the main module.
xCarrier			UDT_Prorunner_mk5_Carrier This is a direct reference to the DB of the main module. This allows the encoder to directly update the position of the carriers.

8 Function block infeed module RDC2

Figuur 09: Function block infeed module RDC2



With this software block, the infeed module RDC2 is called.

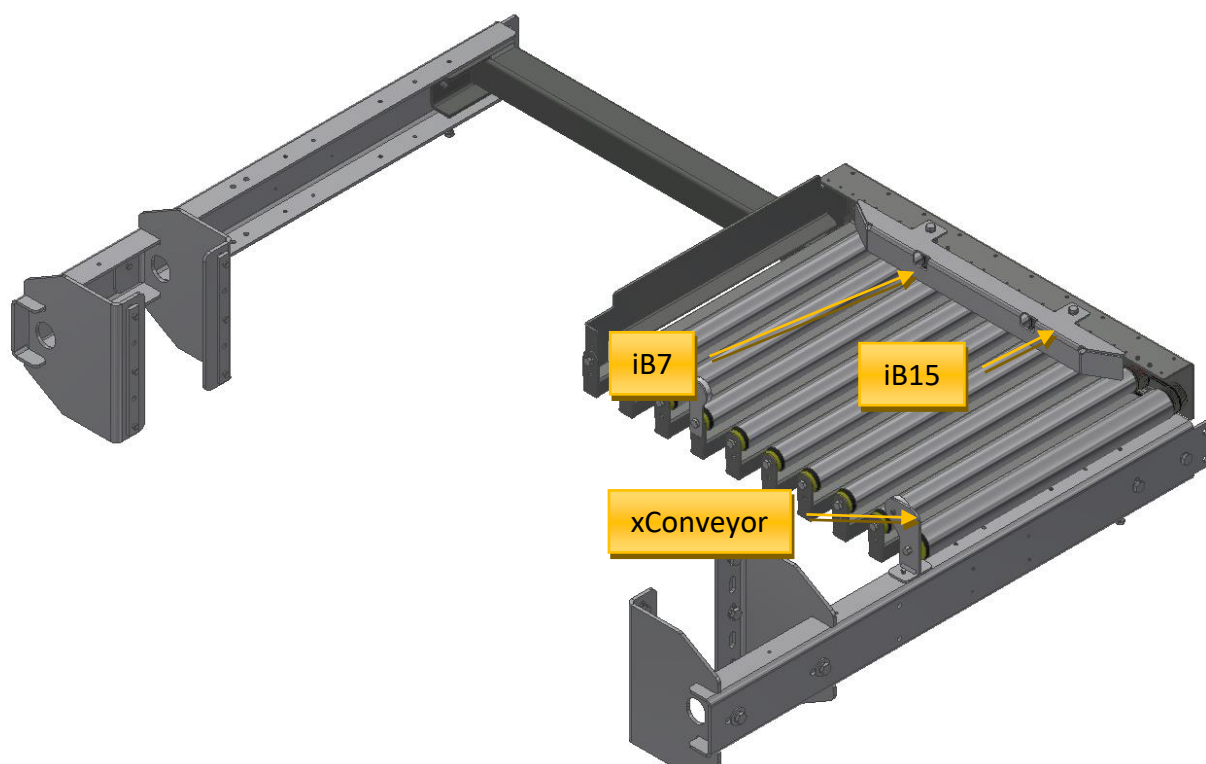
'RDC2' stands for: Roller Drive Conveyor type 2 → Fixed position

8.1 Function block specifications infeed module RDC2

Block name:	FB_Prорunner_mk5_Infeed_RDC2
Block number:	FB1062
Version:	V1.2
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_MotorControl UDT_Prорunner_mk5_Carrier UDT_Prорunner_mk5_Customer_Information UDT_Prорunner_mk5_Infeed_Configuration UDT_Prорunner_mk5_Infeed_Handshake
Function block call	Cyclical (OB1)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

8.2 Components infeed module RDC2

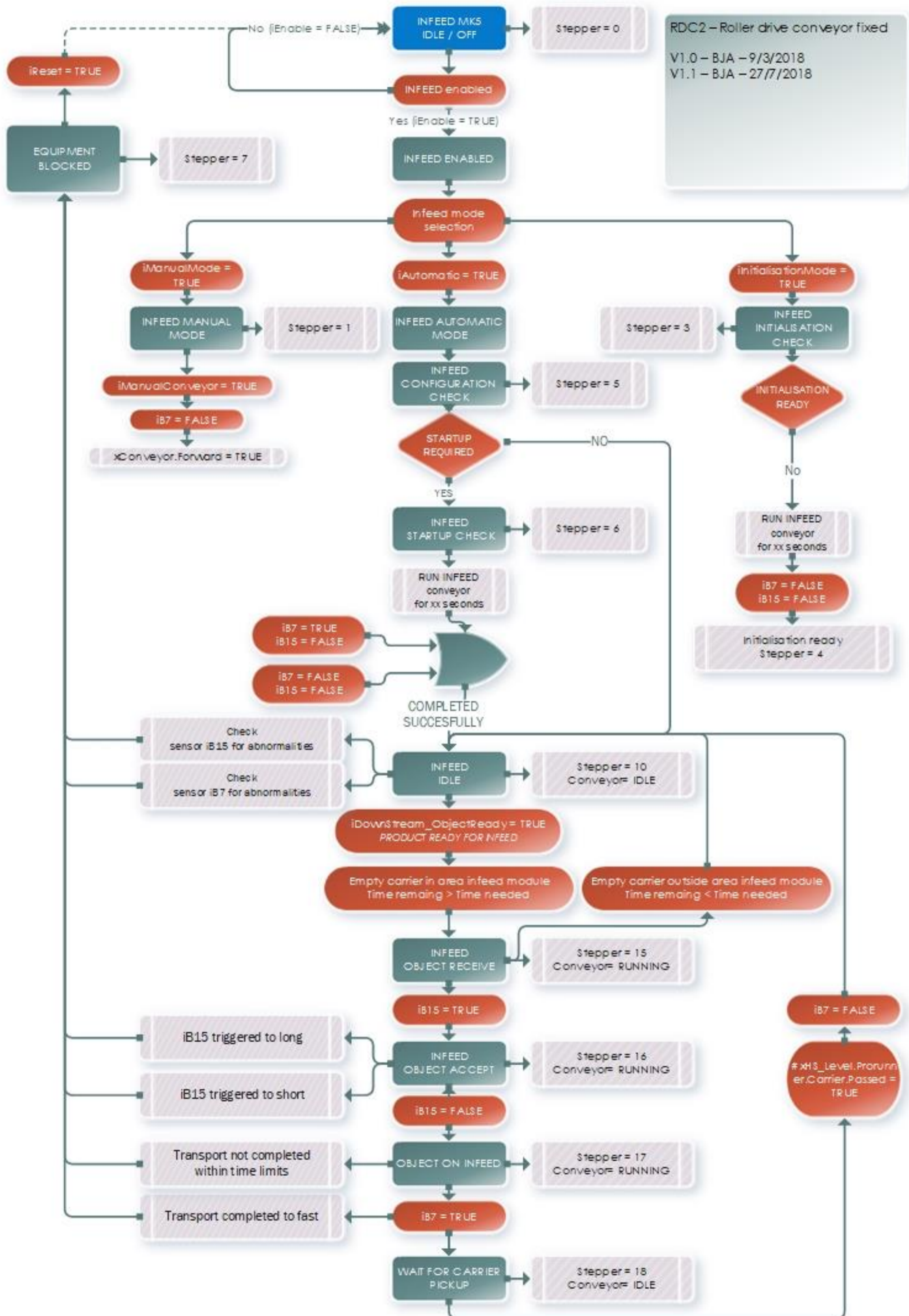
Figuur 10: Components infeed module RDC2



Tabel 12: Components infeed module RDC2

Code		Type
iB15	Sensor/ photocell: Product position monitoring. This photocell monitors whether the product has completely entered the infeed roller conveyor.	GL6-P4211 C42-2
iB15	Sensor/ photocell: Product entered. This photocell detects whether the product has entered the infeed roller conveyor	GL6-P4211 C42-2
xConveyor	Actuation roller conveyor	Drivecontrol 20/54 Rollerdrive EC310

8.3 Flow chart functionality infeed module RDC2



8.4 Configuration infeed module RDC2

Configuration according to UDT_Prorunner_mk5_Infeed_Configuration

Tabel 13: Configuration infeed module RDC2

Naming	Data type	Unit	Description
Module.Level	Int	1..20	Level number.
Module.Height	Real	mm.	Exact height of the conveyor, measured from the ground floor
Options.LowspeedPickup	Bool	0/1	Picking up by using lower speed (1=active)
Conveyor.Speed	Real	m/s.	Speed of conveyor track at rated actuation (metre per second)
Conveyor.Length	Real	mm.	Exact length of infeed conveyor
Transport.CycleTime	Real	s.	Time necessary to perform infeed transport.
Transport.Speed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Transport.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Transport.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill

8.5 Inputs infeed module RDC2

Inputs must be declared by the customer. Inputs provide information such as configuration, status of installation, status sensors, manual pushbutton and communication signals with downstream transport

Tabel 14: Inputs infeed module RDC2

Naming	Data type	Unit	Description
iConfiguration	UDT		UDT as discussed in section above
iEnabled	Bool	0/1	System is switched on. Safety functions are operational and power supplies to the controllers are switched on.
iReset	Bool	0/1	Reset command to reset faults.
iInitialisationMode	Bool	0/1	Start of initialisation procedure.
iAutomaticMode	Bool	0/1	Installation is in automatic mode. Operation is influenced by the sensors and communication with the infeed/outfeed conveyor.
iManualMode	Bool	0/1	Installation is in manual mode. The operation is influenced by the manual commands.
iManualConveyor	Bool	0/1	Command for manual actuation of roller conveyor
iB7	Bool	0/1	Sensor for position of entered product
iB15	Bool	0/1	Sensor for product position control
iDownStream_ObjectReady	Bool	0/1	Communication signal of downstream transport: Object ready for infeeding.
iDownStream_ObjectDestination	Int	1..20	Communication signal of downstream transport: Object destination level.
iDownStream_ObjectInformation	UDT		Communication signal of downstream transport: 'UDT_Prорunner_mk5_Customer_Information' Any information that can be sent along with each object.

8.6 Outputs infeed module RDC2

Outputs must be declared by the customer. Outputs provide feedback such as faults, status and communication signals with downstream transport.

Tabel 15: Outputs infeed module RDC2

Naming	Data type	Unit	Description
oAlarm	Bool	0/1	Fault active
oAlarmCode	Int	0	Active fault code (See chapter 'Faults' for the correct overview)
oAlarmFeedback	Word	0	Active fault feedback for HMI. One bit is reserved for each fault. Fault 1 = bit 0, fault 2 = bit 1, etc.
oStatus	Int	0	See chapter 'Status Prорunner mk5' for further information.
oDownStream_Start	Bool	0/1	Communication signal to downstream transport: Infeed module ready/busy for transport to the end of the roller conveyor
oDownStream_MotorIsRunning	Bool	0/1	Communication signal to downstream transport: Infeed module roller conveyor running. If this signal stops, the roller conveyor downstream must also stop.

8.7 Motor interface infeed module RDC2

One motor is used for the infeed module. All commands are prepared via this motor interface to handle different forms of motor actuation.

For more information, please refer to chapter '12 - Motor interface'

Tabel 16: Motor interface infeed module RDC2

Naming	Data type	Unit	Description
xConveyor	UDT		UDT_MotorControl Motor interface infeed module RDC2

8.8 Status infeed module RDC2

The status of the infeed module is output, giving the user a clear picture of the status/movements of the infeed module RDC2

Tabel 17: Status infeed module RDC2

Code	Description / Cause
0	NO ACTION: Infeed module switched off / no mode active
1	MANUAL OPERATION: Infeed module in manual mode
3	INITIALISATION PROCEDURE ACTIVE: Infeed module busy with initialisation procedure.
4	INITIALISATION PROCEDURE READY: Infeed module finished with initialisation procedure.
5	CONFIGURATION CHECK: Infeed module finished with initialisation procedure.
6	RESTART / START-UP PROCEDURE: Procedure after fault; this involves checking whether an object is present and whether it is correctly positioned.
7	FAULT: Infeed module blocked by fault
10	WAITING: Infeed module ready for use, object infeeding can be started
15	OBJECT INFEEEDING BUSY: Infeed module busy with object infeeding.
16	OBJECT ACCEPTED ON INFEEED CONVEYOR: Infeed module has received object on belt (sensor iB15)
17	OBJECT PRESENT ON INFEEED CONVEYOR: Infeed module in the process of moving object to the end of the roller conveyor (sensor iB7 covered and iB15 uncovered).
18	WAIT UNTIL OBJECT HAS BEEN PICKED UP: Infeed module waits until Prorunner mk5 has picked up object. Information is also made available in this status/step.

8.9 Faults infeed module RDC2

With the outfeed signal 'oAlarm' it is possible to indicate a fault by, for example, switching on a red lamp. With the fault code 'oAlarmCode' you can indicate this better.

With the fault word 'oAlarmFeedback' you can generate a clearly defined fault message by means of an HMI. The faults are divided over 16 bits. Code 1 = bit 0 ... Code 16 = bit 15.

Below is the list of faults that may be active with explanation/result and solution.

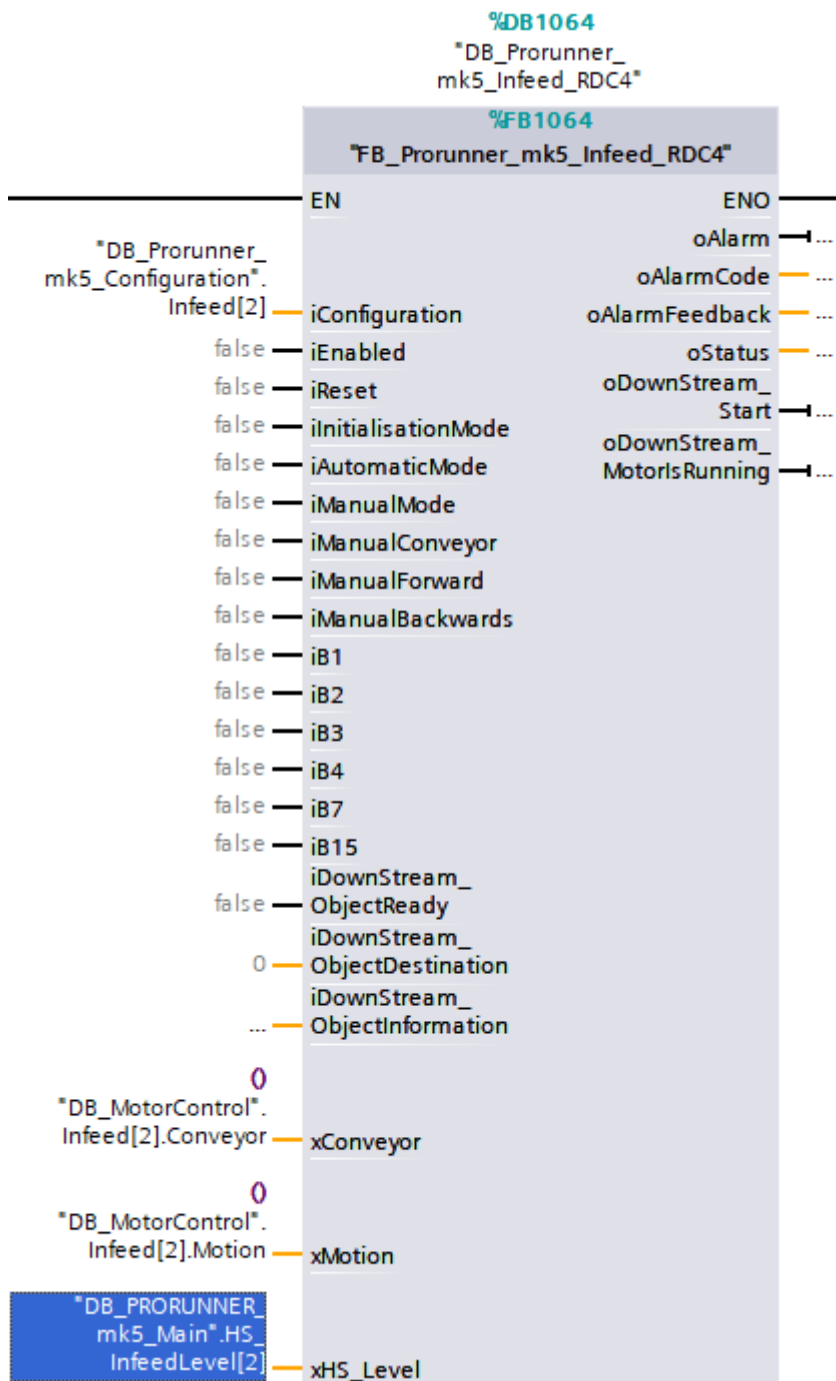
Table 18: Faults list infeed module RDC2

Code	Description / Cause	Effect	Solution
1	Feeding object in is taking too long. Sensor object detection does not see the object within time limits	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
2	Object fed in too fast. It is possible that the product is too long and/or that an unknown object has been seen.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check for unknown objects. Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
3	Unknown object detected. Sensors triggered while the infeed module is not busy with this step.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check for unknown objects. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
4	Object too small. iB15 triggered for too short a time. Time is calculated with the shortest product length	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
5	Object too big. iB15 triggered for too short a time. Time is calculated with the smallest product length	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
6	Unknown destination of object on infeed conveyor. This must higher than '0' and less than or equal to the number of levels present.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Remove object from infeed conveyor and check communication with downstream transport. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
8	Configuration is not entered correctly	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Check configuration settings. Make sure that everything is entered correctly and not below the minimum value.

Code	Description / Cause	Effect	Solution
9	Start-up procedure time-out. Sensor iB15 does not become untriggered.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault.
13	Fault controller/drive conveyor track.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. (Depending on position)	Check operation/fault controller/drive. Make sure that it is ready for operation again. Give the reset command to reset the fault.
15	Initialisation procedure: object still detected on conveyor track.	Initialisation has not been completed.	Empty the conveyor track. Restart procedure by giving initialisation command again.

9 Function block infeed module RDC4

Figuur 11: Function block infeed module RDC4



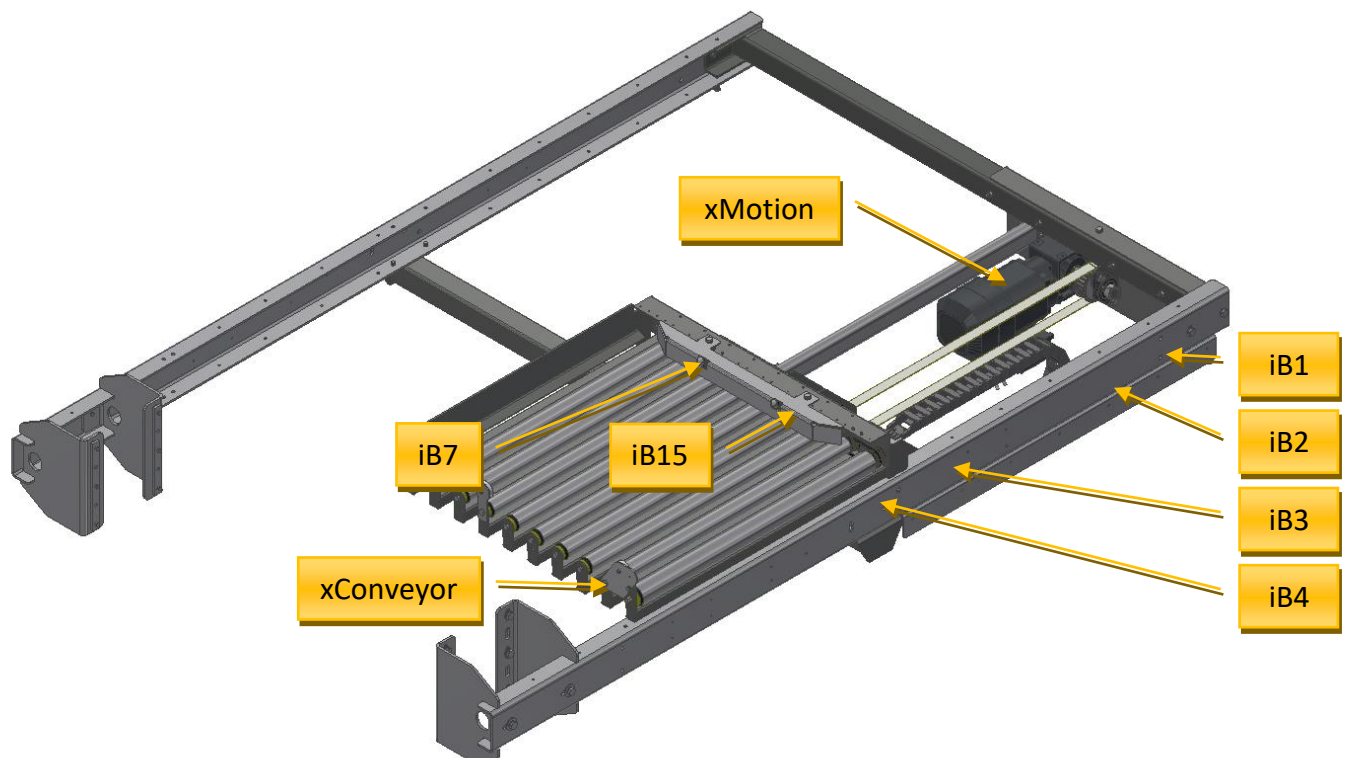
With this software block, the infeed module RDC4 is called.
RDC4 stands for: Roller Drive Conveyor type 4 → Moveable

9.1 Function block specifications infeed module RDC4

Block name:	FB_Prorunner_mk5_Infeed_RDC4
Block number:	FB1064
Version:	V1.2
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_MotorControl UDT_Prorunner_mk5_Carrier UDT_Prorunner_mk5_Customer_Information UDT_Prorunner_mk5_Infeed_Configuration UDT_Prorunner_mk5_Infeed_Handshake
Function block call	Cyclical (OB1)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

9.2 Components infeed module RDC4

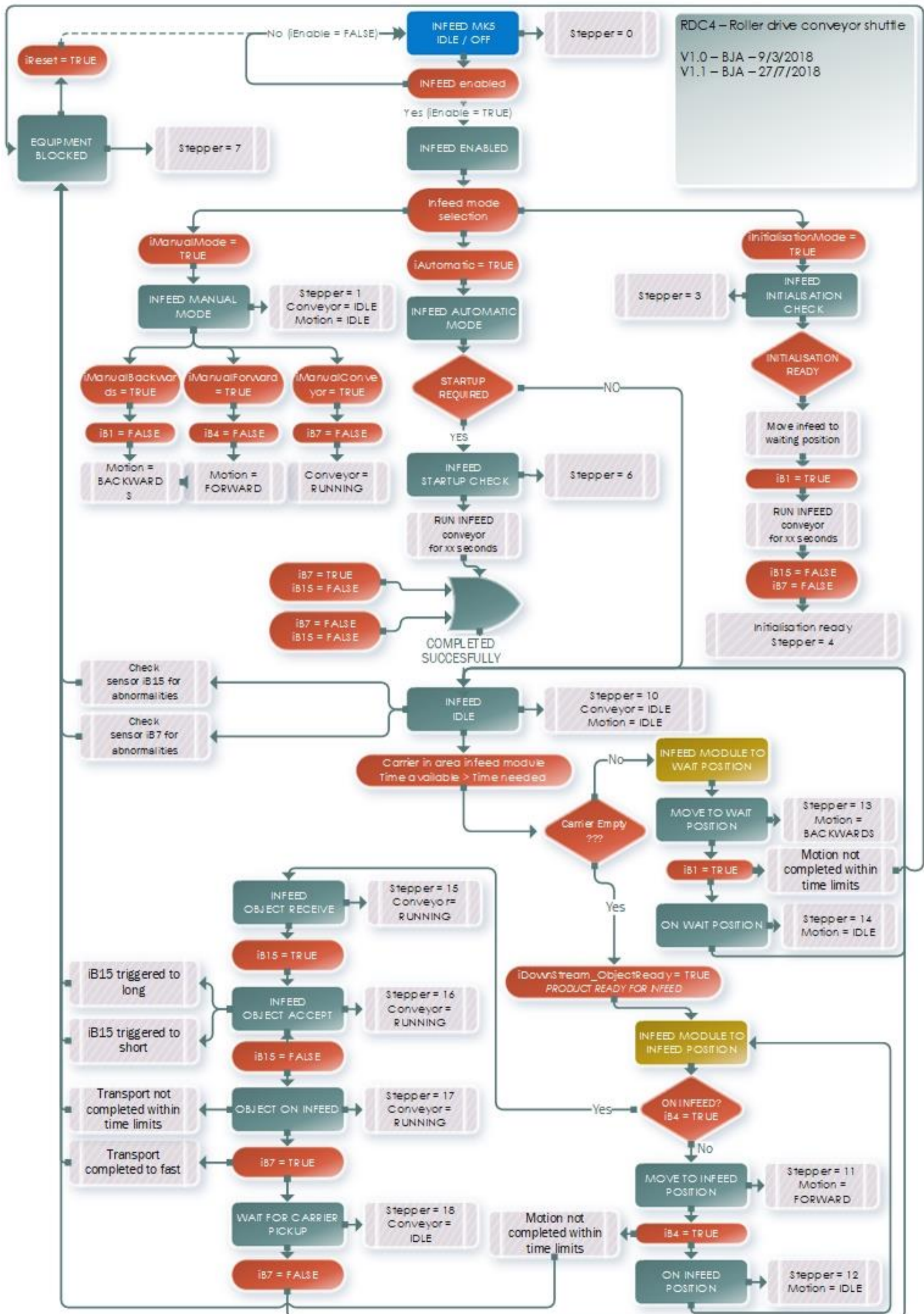
Figuur 12: Components infeed module RDC4



Tabel 19: Components infeed module RDC4

Code		Type
iB15	Sensor/ photocell: Product position monitoring. This photocell monitors whether the product has completely entered the infeed roller conveyor.	GL6-P4211 C42-2
iB7	Sensor/ photocell: Product entered. This photocell detects whether the product has entered the infeed roller conveyor	GL6-P4211 C42-2
iB12	Inductive approach: Wait for position (from Prorunner) Sensor detection that horizontal movement is in waiting position. This is the position in which no object can be entered, and the conveyor is in a safe position for the Prorunner carriers.	IME18-08BPSZC0K
iB13	Inductive approach: Low speed waiting position Sensor detection that horizontal movement should switch to waiting position to low speed.	IME18-08BPSZC0K
iB14	Inductive approach: Infeed position (In Prorunner) Sensor detection that horizontal movement is at Infeed position. This is the position in which an object can be fed in.	IME18-08BPSZC0K
iB15	Inductive approach: Low-speed infeed position Sensor detection that horizontal movement should switch to waiting position to low speed.	IME18-08BPSZC0K
xConveyor	Actuation roller conveyor	Drivecontrol 20/54 Rollerdrive EC310
xMotion	Actuation horizontal movement	SEW drive See Qimarox info

RDC4 – Roller drive conveyor shuttle
V1.0 – BJA – 9/3/2018
V1.1 – BJA – 27/7/2018



9.5 Configuration infeed module RDC4

Configuration according to UDT_Prorunner_mk5_Infeed_Configuration

Tabel 20: Configuration infeed module RDC4

Naming	Data type	Unit	Description
Module.Level	Int	1..20	Level number.
Module.Height	Real	mm.	Exact height of the conveyor measured from the floor
Options.LowspeedPickup	Bool	0/1	Picking up by using lower speed (1=active)
Conveyor.Speed	Real	m/s.	Speed of conveyor track at rated actuation (metre per second)
Conveyor.Length	Real	mm.	Exact length of infeed conveyor
Transport.CycleTime	Real	s.	Time necessary to perform infeed transport.
Transport.Speed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Transport.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Transport.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill
Movement.CycleTime	Real	s.	Time needed to move conveyor track between infeed position and waiting position.
Movement.InfeedPosition	Real	mm.	Position: infeed position
Movement.BasicPosition	Real	mm.	Position: waiting position
Movement.Speed	Real	m/s.	Speed of movement
Movement.HighSpeed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Movement.LowSpeed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Movement.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Movement.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill

9.6 Inputs infeed module RDC4

Inputs must be declared by the customer. Inputs provide information such as configuration, status of installation, status sensors, manual pushbutton and communication signals with downstream transport

Tabel 21: Inputs infeed module RDC4

Naming	Data type	Unit	Description
iConfiguration	UDT		UDT as discussed in section above
iEnabled	Bool	0/1	System is switched on. Safety functions are operational and power supplies to the controllers are switched on.
iReset	Bool	0/1	Reset command to reset faults.
iInitialisationMode	Bool	0/1	Start of initialisation procedure.
iAutomaticMode	Bool	0/1	Installation is in automatic mode. Operation is influenced by the sensors and communication with the infeed/outfeed conveyor.
iManualMode	Bool	0/1	Installation is in manual mode. The operation is influenced by the manual commands.
iManualConveyor	Bool	0/1	Command for manual actuation of roller conveyor
iManualForwards	Bool	0/1	Command manual actuation of horizontal movement to the infeed position
iManualBackwards	Bool	0/1	Command manual actuation of horizontal movement to the waiting position
iB1	Bool	0/1	Sensor for detection of waiting position
iB2	Bool	0/1	Sensor for detection of low speed waiting position
iB3	Bool	0/1	Sensor for detection of low-speed infeed position
iB4	Bool	0/1	Sensor for detection of infeed position
iB7	Bool	0/1	Sensor for position of entered product
iB15	Bool	0/1	Sensor for product position control
iDownStream_ObjectReady	Bool	0/1	Communication signal of downstream transport: Object ready for infeeding.
iDownStream_ObjectDestination	Int	1..20	Communication signal of downstream transport: Object destination level.
iDownStream_ObjectInformation	UDT		Communication signal of downstream transport: 'UDT_Prорunner_mk5_Customer_Information' Any information that can be sent along with each object. This UDT can be configured by yourself.

9.7 Outputs infeed module RDC4

Exits must be declared by the customer. Outputs provide feedback such as faults, status and communication signals with downstream transport.

Tabel 22: Outputs infeed module RDC4

Naming	Data type	Unit	Description
oAlarm	Bool	0/1	Fault active
oAlarmCode	Int	0	Active fault code (See chapter 'Faults' for the correct overview)
oAlarmFeedback	Word	0	Active fault feedback for HMI. One bit is reserved for each fault. Fault 1 = bit 0, fault 2 = bit 1, etc.
oStatus	Int	0	See chapter 'Status Prorunner mk5' for further information.
oDownStream_Start	Bool	0/1	Communication signal to downstream transport: Infeed module ready/busy for transport to the end of the roller conveyor
oDownStream_MotorIsRunning	Bool	0/1	Communication signal to downstream transport: Infeed module roller conveyor running. If this signal stops, the roller conveyor downstream must also stop.

9.8 Motor interface infeed module RDC4

Two motors are used for the infeed module. All commands are prepared via this motor interface to handle different forms of motor actuation.

For more information, please refer to chapter '12 - Motor interface'

Tabel 23: Motor interface infeed module RDC4

Naming	Data type	Unit	Description
xConveyor	UDT		UDT_MotorControl Motor interface roller conveyor infeed module RDC4
xMotion	UDT		UDT_MotorControl Motor interface horizontal movement infeed module RDC4

9.9 Status infeed module RDC4

The status of the infeed module is output, giving the user a clear picture of the status/movements of the infeed module RDC4

Tabel 24: Status infeed module RDC4

Code	Description / Cause
0	NO ACTION: Infeed module switched off / no mode active
1	MANUAL OPERATION: Infeed module in manual mode
3	INITIALISATION PROCEDURE ACTIVE: Infeed module busy with initialisation procedure.
4	INITIALISATION PROCEDURE READY: Infeed module finished with initialisation procedure.
5	CONFIGURATION CHECK: Infeed module finished with initialisation procedure.
6	RESTART / START-UP PROCEDURE: Procedure after fault; this involves checking whether an object is present and whether it is correctly positioned.
7	FAULT: Infeed module blocked by fault
10	WAITING: Infeed module ready for use, object infeeding can be started
11	MOVEMENT TO INFEED POSITION: Infeed module busy with horizontal movement to infeed position
12	AT INFEED POSITION: Infeed module ready for transport of object
13	MOVEMENT TO WAITING POSITION: Infeed module busy with horizontal movement to waiting position
14	AT WAIT POSITION: Infeed module in waiting position.
15	OBJECT INFEEEDING BUSY: Infeed module busy with object infeeding.
16	OBJECT ACCEPTED ON INFEED CONVEYOR: Infeed module has received object on belt (sensor iB15)
17	OBJECT PRESENT ON INFEED CONVEYOR: Infeed module in the process of moving object to the end of the roller conveyor (sensor iB7 covered and iB15 uncovered).
18	WAIT UNTIL OBJECT HAS BEEN PICKED UP: Infeed module waits until Prorunner mk5 has picked up object. Information is also made available in this status/step.

9.10 Faults infeed module RDC4

With the outfeed signal 'oAlarm' it is possible to indicate a fault by, for example, switching on a red lamp. With the fault code 'oAlarmCode' you can indicate this better.

With the fault word 'oAlarmFeedback' you can generate a clearly defined fault message by means of an HMI. The faults are divided over 16 bits. Code 1 = bit 0 ... Code 16 = bit 15.

Below is the list of faults that may be active with explanation/effect and solution.

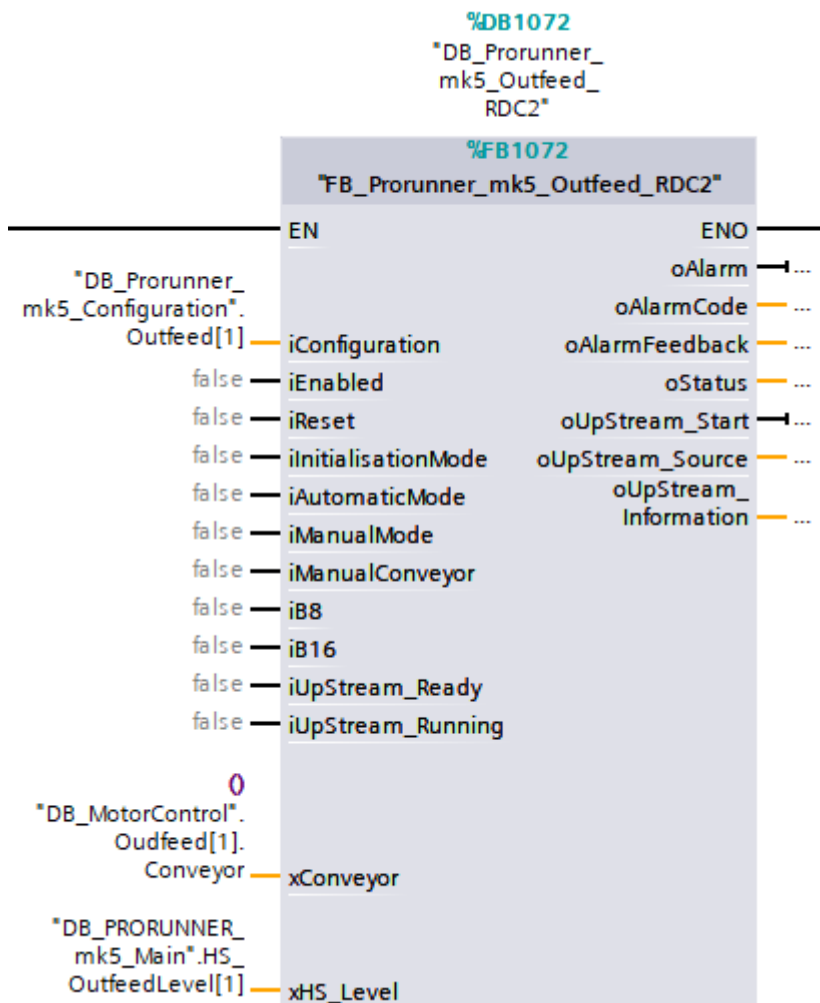
Table 25: Faults list infeed module RDC4

Code	Description / Cause	Effect	Solution
1	Feeding object in is taking too long. Sensor object detection does not see the object within time limits	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
2	Object fed in too fast. It is possible that the product is too long and/or that an unknown object has been seen.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check for unknown objects. Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
3	Unknown object detected. Sensors triggered while the infeed module is not busy with this step.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check for unknown objects. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
4	Object too small. iB15 triggered for too short a time. Time is calculated with the shortest product length	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
5	Object too big. iB15 triggered for too short a time. Time is calculated with the smallest product length	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
6	Unknown destination of object on infeed conveyor. This must higher than '0' and less than or equal to the number of levels present.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. Start-up procedure of infeed conveyor must be carried out.	Remove object from infeed conveyor and check communication with downstream transport. Give the reset command to reset the fault. Resetting is not possible if sensor iB15 is covered.
8	Configuration is not entered correctly	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Check configuration settings. Make sure that everything is entered correctly and not below the minimum value.

Code	Description / Cause	Effect	Solution
9	Start-up procedure time-out. Sensor iB15 does not become untriggered.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault.
10	Movement to waiting position is taking too long	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Check why this is taking too long. Check adjustment/operation of the iB1 sensor Give the reset command to reset the fault.
11	Movement to waiting position is taking too long	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Check why this is taking too long. Check adjustment/operation of the iB4 sensor Give the reset command to reset the fault.
12	Position horizontal movement no longer correct.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked.	Check adjustment/operation of the iB1 & iB4 sensor. Give the reset command to reset the fault.
13	Fault controller/drive conveyor track.	Automatic operation is stopped. Command to PRORUNNER that infeed module is blocked. (Depending on position)	Check operation/fault controller/drive. Make sure that it is ready for operation again. Give the reset command to reset the fault.
14	Fault controller/drive horizontal movement.	See fault 13	See fault 13
15	Initialisation procedure: object still detected on conveyor track.	Initialisation has not been completed.	Empty the conveyor track. Restart procedure by giving initialisation command again.
16	Initialisation procedure: Safe position not reached	Initialisation has not been completed.	Check the movement of the infeed module. Check operation and adjustment of position indicator sensors. Restart procedure by giving initialisation command again.

10 Function block outfeed module RDC2

Figuur 13: Function block outfeed module RDC2



With this software block, the outfeed module RDC2 is called.

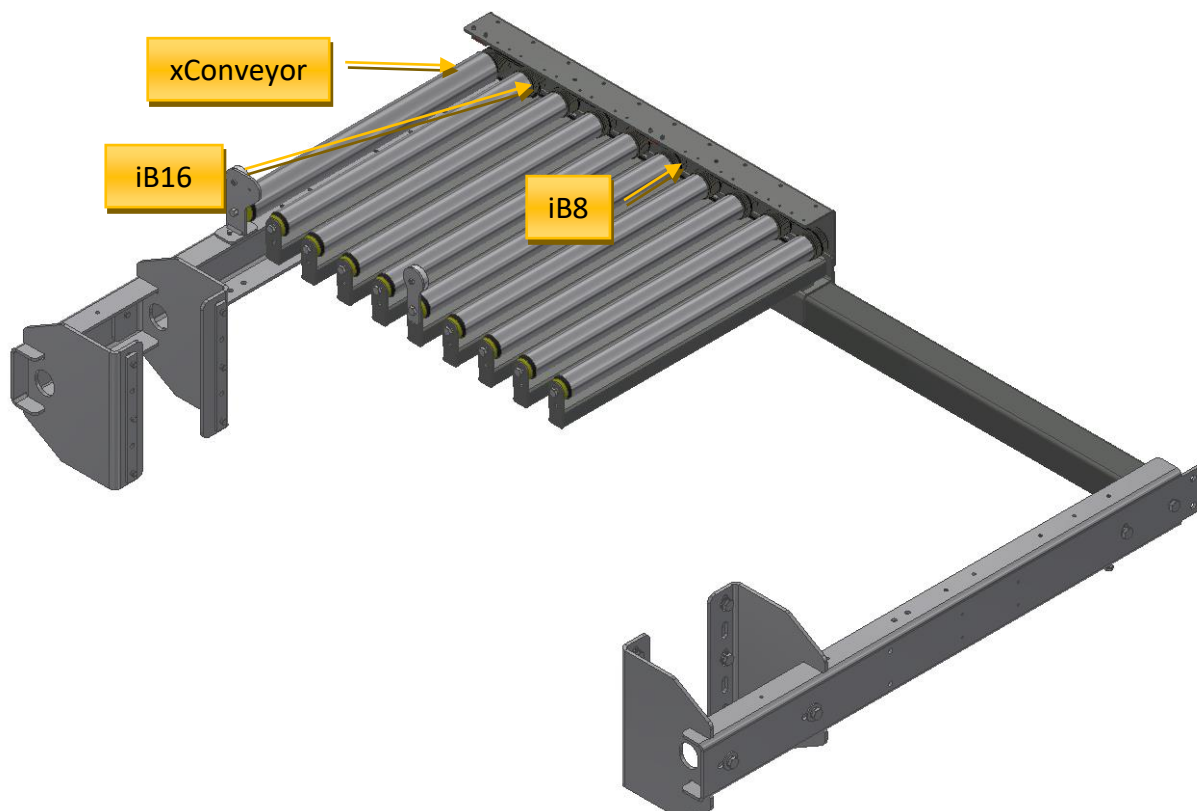
'RDC2' stands for: Roller Drive Conveyor type 2 → Fixed position

10.1 Function block specifications outfeed module RDC2

Block name:	FB_Prorunner_mk5_Outfeed_RDC2
Block number:	FB1072
Version:	V1.2
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_MotorControl UDT_Prorunner_mk5_Carrier UDT_Prorunner_mk5_Customer_Information UDT_Prorunner_mk5_Outfeed_Configuration UDT_Prorunner_mk5_Outfeed_Handshake
Function block call	Cyclical (OB1)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

10.2 Components outfeed module RDC2

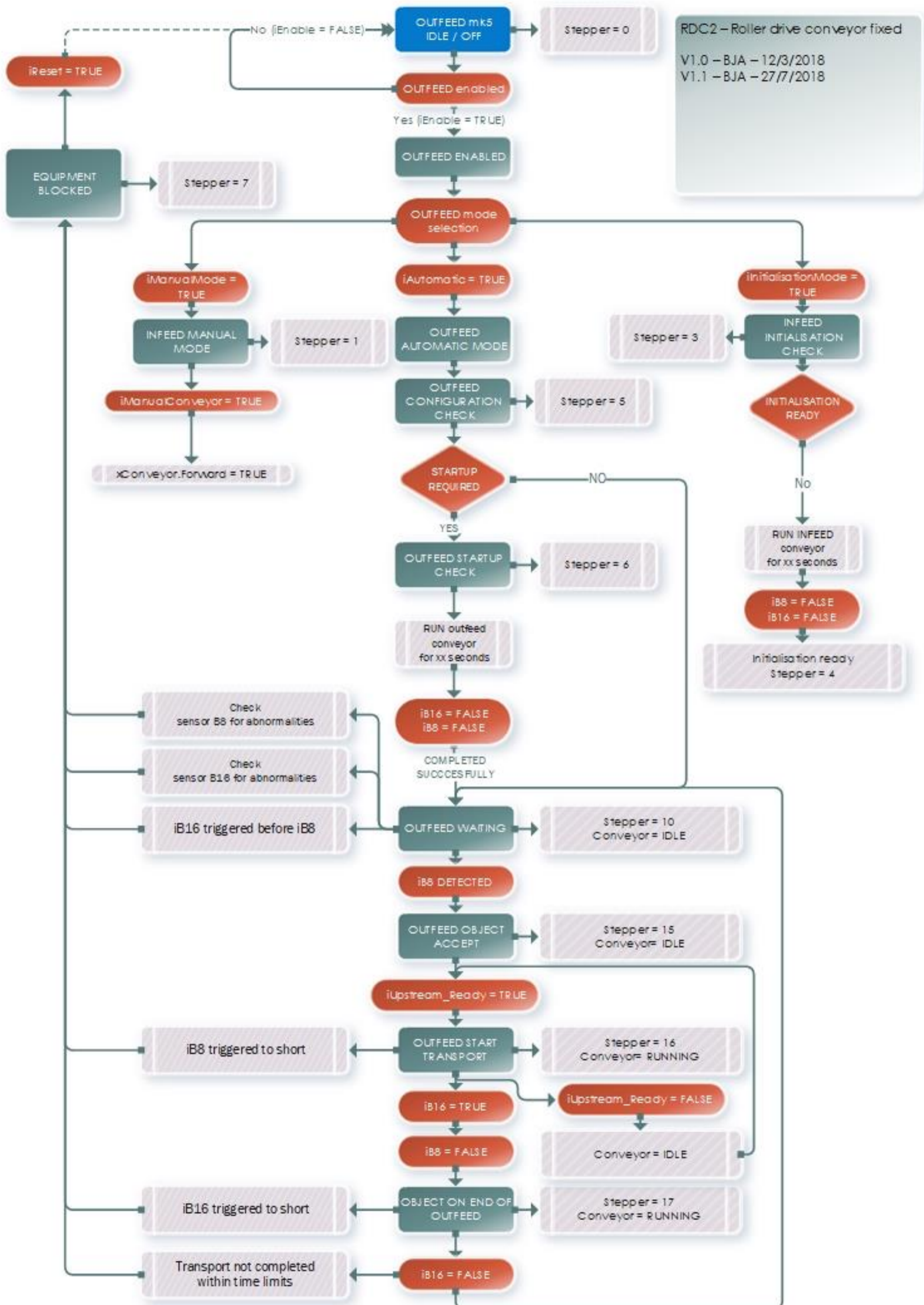
Figuur 14: Components outfeed module RDC2



Tabel 26: Components outfeed module RDC2

Code		Type
iB8	Sensor/Photocell: Product present on module This photocell monitors whether a product is present on the outfeed conveyor.	GL6-P4211 C42-2
iB16	Sensor/photocell; Product has left module This photocell detects whether the product has run out on the outfeed conveyor	GL6-P4211 C42-2
xConveyor	Actuation roller conveyor	Drivecontrol 20/54 Rollerdrive EC310

10.3 Flow chart functionality outfeed module RDC2



10.4 Configuration outfeed module RDC2

Configuration according to UDT_Prorunner_mk5_Outfeed_Configuration

Tabel 27: Configuration outfeed module RDC2

Naming	Data type	Unit	Description
Module.Level	Int	1..20	Level number.
Module.Height	Real	mm.	Exact height of the belt, measured from the floor
Options.LowspeedDropoff	Bool	0/1	Drop off by means of use of lower speed (1=active)
Conveyor.Speed	Real	m/s.	Speed of conveyor track at rated actuation (metre per second)
Conveyor.Length	Real	mm.	Exact length of outfeed conveyor
Transport.CycleTime	Real	s.	Time needed to perform outfeed transport.
Transport.Speed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Transport.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Transport.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill

10.5 Inputs outfeed module RDC2

Inputs must be declared by the customer.

Inputs provide information such as configuration, status of installation, status sensors, manual pushbutton and communication signals with upstream transport

Tabel 28: Inputs outfeed module RDC2

Naming	Data type	Unit	Description
iConfiguration	UDT		UDT as discussed in section above
iEnabled	Bool	0/1	System is switched on. Safety functions are operational and power supplies to the controllers are switched on.
iReset	Bool	0/1	Reset command to reset faults.
iInitialisationMode	Bool	0/1	Start of initialisation procedure.
iAutomaticMode	Bool	0/1	Installation is in automatic mode. Operation is influenced by the sensors and communication with the infeed/outfeed conveyor.
iManualMode	Bool	0/1	Installation is in manual mode. The operation is influenced by the manual commands.
iManualConveyor	Bool	0/1	Command for manual actuation of roller conveyor
iB8	Bool	0/1	Sensor for registering product presence
iB16	Bool	0/1	Sensor for detection of product running out onto outfeed conveyor
iUpStream_Ready	Bool	0/1	Communication signal of upstream transport: Transport system ready to receive object
iUpStream_Running	Bool	0/1	Communication signal of upstream transport: Transport system systems ready & transport running.

10.6 Outputs outfeed module RDC2

Exits must be declared by the customer.

Outputs provide feedback such as faults, status and communication signals with upstream transport.

Tabel 29: Outputs outfeed module RDC2

Naming	Data type	Unit	Description
oAlarm	Bool	0/1	Fault active
oAlarmCode	Int	0	Active fault code (See chapter 'Faults' for the correct overview)
oAlarmFeedback	Word	0	Active fault feedback for HMI. One bit is reserved for each fault. Fault 1 = bit 0, fault 2 = bit 1, etc.
oStatus	Int	0	See chapter 'Status Prorunner mk5' for further information.
oUpStream_ObjectReady	Bool	0/1	Communication signal to the upstream transport: Object ready for outfeed.
oUpStream_ObjectSource	Int	1..20	Communication signal to the upstream transport: Object source level.
oUpStream_ObjectInformation	UDT		Communication signal to the upstream transport: 'UDT_Prорunner_mk5_Customer_Information' Additional information that was sent with the object.

10.7 Motor interface outfeed module RDC2

One motor is used for the outfeed module. All commands are prepared via this motor interface to handle different forms of motor actuation.

For more information, please refer to chapter '12 - Motor interface'

Tabel 30: Motor interface outfeed module RDC2

Naming	Data type	Unit	Description
xConveyor	UDT		UDT_MotorControl Motor interface outfeed module RDC2

10.8 Status outfeed module RDC2

The status of the outfeed module is output, giving the user a clear picture of the status/movements of the outfeed module RDC2.

Tabel 31: Status outfeed module RDC2

Code	Description / Cause
0	NO ACTION: Outfeed module switched off / no mode active
1	MANUAL OPERATION: Outfeed module in manual mode
3	INITIALISATION PROCEDURE ACTIVE: Outfeed module busy with initialisation procedure.
4	INITIALISATION PROCEDURE READY: Outfeed module finished with initialisation procedure.
5	CONFIGURATION CHECK: Outfeed module finished with initialisation procedure.
6	RESTART / START-UP PROCEDURE: Procedure after fault; this involves checking whether an object is present and whether it is correctly positioned.
7	FAULT: Outfeed module blocked by fault
10	WAITING: Outfeed module ready for use; depositing objects allowed
15	OBJECT ACCEPTED ON OUTFEED TRACK: Outfeed module has received object on track (sensor iB8)
16	OBJECT TRANSPORT BUSY Outfeed module in the process of moving object to the end of the conveyor track (sensor iB16 covered)
17	OBJECT AT END OF TRACK: Object at end of the conveyor track, waiting for iB16 to be uncovered.

10.9 Faults outfeed module RDC2

With the outfeed signal 'oAlarm' it is possible to indicate a fault by, for example, switching on a red lamp. With the fault code 'oAlarmCode' you can indicate this better.

With the fault word 'oAlarmFeedback' you can generate a clearly defined fault message by means of an HMI. The faults are divided over 16 bits. Code 1 = bit 0 ... Code 16 = bit 15.

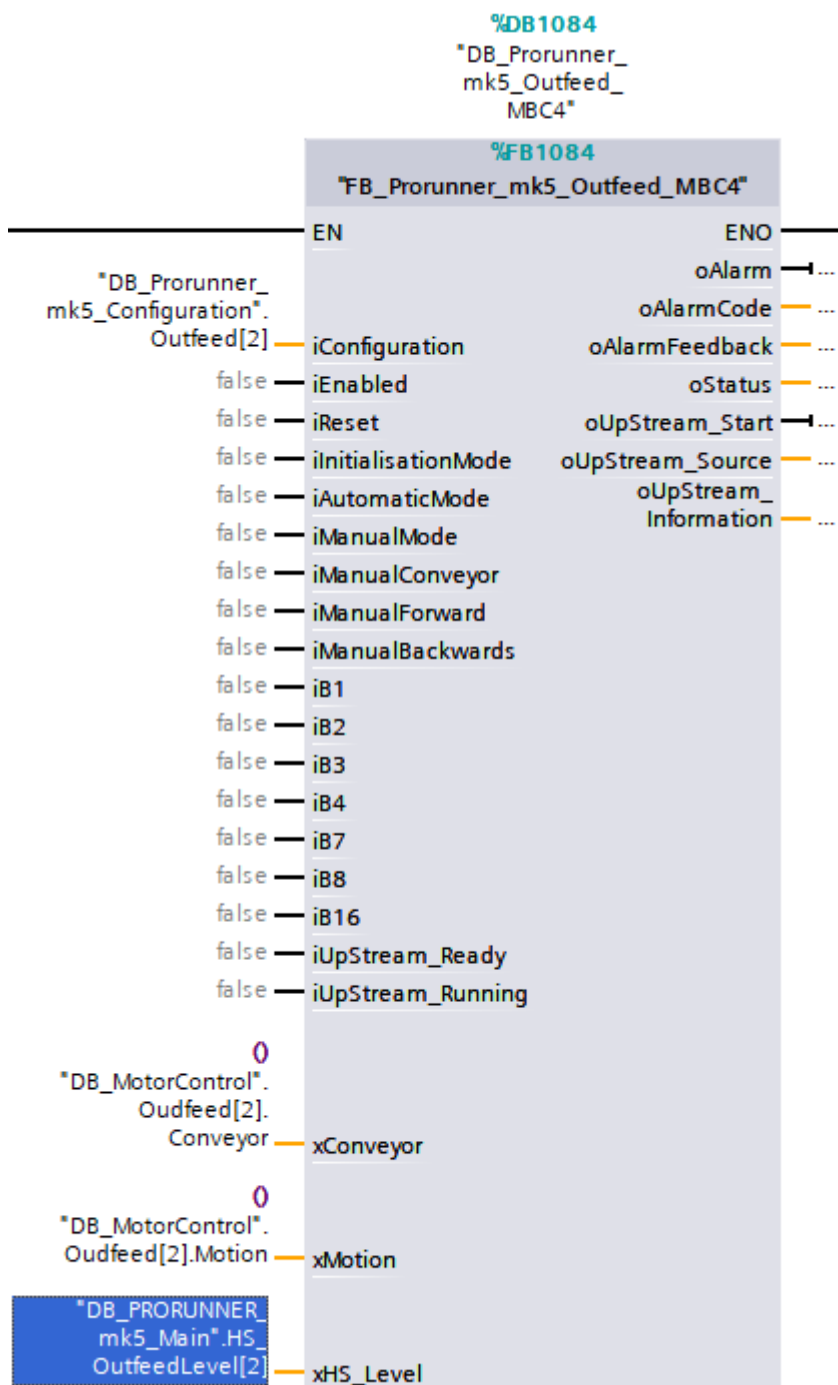
Below is the list of faults that may be active with explanation/result and solution.

Tabel 32: Faults list outfeed module RDC2

Code	Description / Cause	Effect	Solution
1	Object outfeeding is taking too long.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Remove object manually from the conveyor track Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
3	Unknown object detected. Sensors triggered while the outfeed module is not busy with the corresponding step.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check for unknown objects. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
4	Object too small. iB16 triggered for too short a time. Time is calculated with the minimum product length	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
5	Object too big. iB16 triggered for too short a time. Time is calculated with the maximum product length	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
8	Configuration is not entered correctly	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Check configuration settings. Make sure that everything is entered correctly and not below the minimum value.
9	Start-up procedure time-out. Sensor iB8 & iB16 does not become untriggered.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Remove object manually; iB8 uncovered & iB16 uncovered. Give the reset command to reset the fault.
13	Fault controller/drive conveyor track.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. (Depending on position)	Check operation/fault controller/drive. Make sure that it is ready for operation again. Give the reset command to reset the fault.
15	Initialisation procedure: object still detected on conveyor track.	Initialisation has not been completed.	Empty the conveyor track. Restart procedure by giving initialisation command again.

11 Function block outfeed module MBC4

Figuur 15: Function block outfeed module MBC4



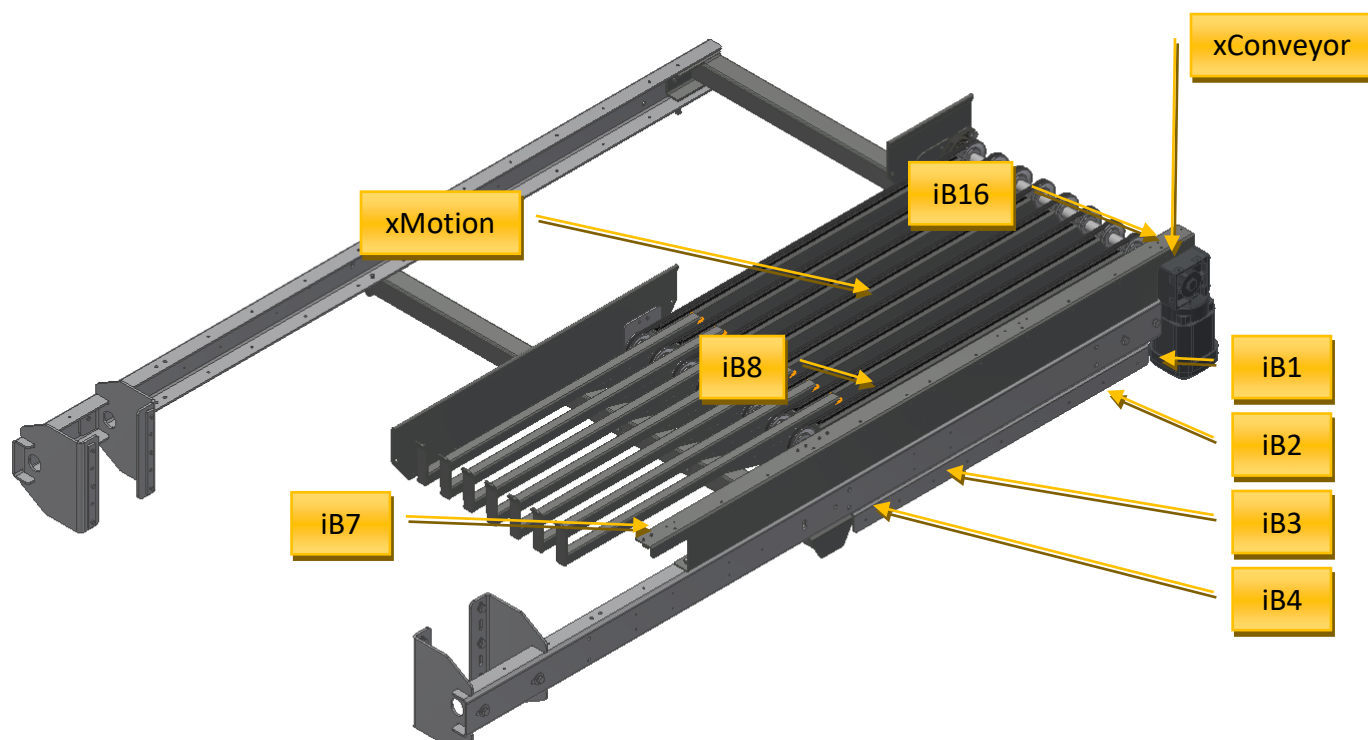
With this software block, the outfeed module MBC4 is called.
MBC4 stands for: Multi Belt Drive type 4 → Moveable

11.1 Function block specifications outfeed module MBC4

Block name:	FB_Prorunner_mk5_Outfeed_MBC4
Block number:	FB1084
Version:	V1.2
Required PLC blocks:	TON_TIMER
Required PLC data types:	UDT_MotorControl UDT_Prorunner_mk5_Carrier UDT_Prorunner_mk5_Customer_Information UDT_Prorunner_mk5_Infeed_Configuration UDT_Prorunner_mk5_Infeed_Handshake
Function block call	Cyclical (OB1)
Optimised block access	Yes/ No
Programming language	LAD, SCL
STEP7 version	TIA Portal V14 + SP1

11.2 Components outfeed module MBC4

Figuur 16: Components outfeed module MBC4



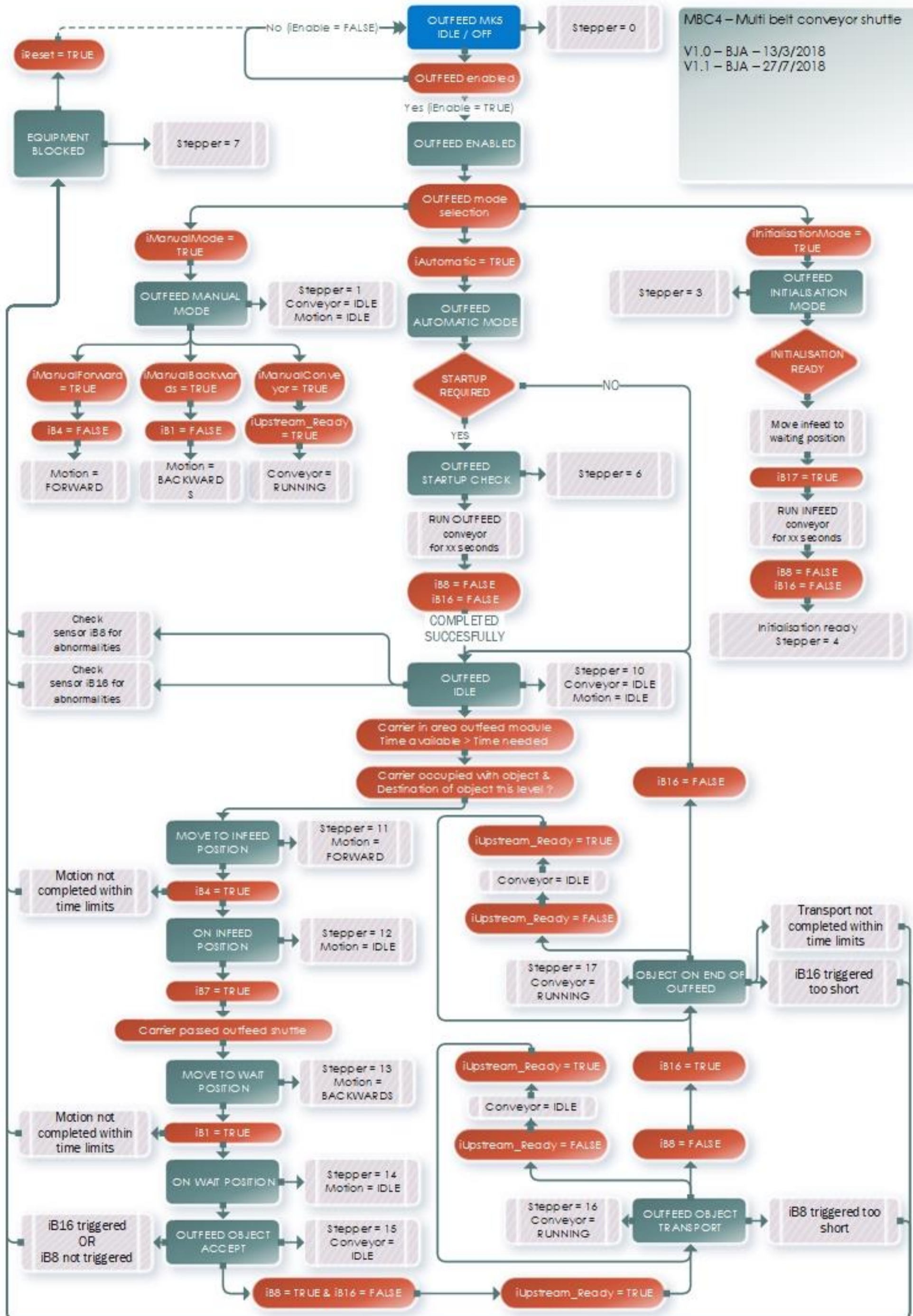
Tabel 33: Components outfeed module MBC4

Code		Type
iB8	Sensor/photocell: Product present on track	GL6-P4211 C42-2
iB16	Sensor/photocell: Product run out onto conveyor	GL6-P4211 C42-2
iB17	Inductive approach: Wait for position (from Prorunner) Sensor detection that horizontal movement is in waiting position. This is the position in which no object can be entered, and the conveyor is in a safe position for the Prorunner carriers.	IME18-08BPSZC0K
iB18	Inductive approach: Low speed waiting position Sensor detection that horizontal movement should switch to waiting position to low speed.	IME18-08BPSZC0K
iB19	Inductive approach: Outfeed position (In Prorunner) Sensor detection that horizontal movement is at Infeed position. This is the position in which an object can be fed in.	IME18-08BPSZC0K
iB20	Inductive approach: Low-speed outfeed position Sensor detection that horizontal movement should switch to waiting position to low speed.	IME18-08BPSZC0K
iB21	Sensor/probe: Product deposited on rake	GTB6-P4211
xConveyor	Actuation roller conveyor	SEW drive See Qimarox info
xMotion	Actuation horizontal movement	SEW drive See Qimarox info

MBC4 – Multi belt conveyor shuttle

V1.0 – BJA – 13/3/2018

V1.1 – BJA – 27/7/2018



11.5 Configuration outfeed module MBC4

Configuration according to UDT_Prorunner_mk5_Outfeed_Configuration

Tabel 34: Configuration outfeed module MBC4

Naming	Data type	Unit	Description
Module.Level	Int	1..20	Level number.
Module.Height	Real	mm.	Exact height of the conveyor measured from the floor
Options.LowspeedDropoff	Bool	0/1	Drop off by means of use of lower speed (1=active)
Conveyor.Speed	Real	m/s.	Speed of conveyor track at rated actuation (metre per second)
Conveyor.Length	Real	mm.	Exact length of outfeed conveyor
Transport.CycleTime	Real	s.	Time needed to perform outfeed transport.
Transport.Speed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Transport.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Transport.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill
Movement.CycleTime	Real	s.	Time needed to move conveyor track between outfeed position and waiting position.
Movement.InfeedPosition	Real	mm.	Position: outfeed position
Movement.BasicPosition	Real	mm.	Position: waiting position
Movement.Speed	Real	m/s.	Speed of movement
Movement.HighSpeed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Movement.LowSpeed	Int	Hz	Number of Hertz (Nominal = 50 Hz)
Movement.Acceleration	Int	ms.	Number of milliseconds to accelerate from standstill to desired speed
Movement.Deceleration	Int	ms.	Number of milliseconds to decelerate from current speed to standstill

11.6 Inputs outfeed module MBC4

Inputs must be declared by the customer.

Inputs provide information such as configuration, status of installation, status sensors, manual pushbutton and communication signals with upstream transport

Tabel 35: Inputs outfeed module MBC4

Naming	Data type	Unit	Description
iConfiguration	UDT		UDT as discussed in section above
iEnabled	Bool	0/1	System is switched on. Safety functions are operational and power supplies to the controllers are switched on.
iReset	Bool	0/1	Reset command to reset faults.
iInitialisationMode	Bool	0/1	Start of initialisation procedure.
iAutomaticMode	Bool	0/1	Installation is in automatic mode. Operation is influenced by the sensors and communication with the infeed/outfeed conveyor.
iManualMode	Bool	0/1	Installation is in manual mode. The operation is influenced by the manual commands.
iManualConveyor	Bool	0/1	Command for manual actuation of roller conveyor
iManualForwards	Bool	0/1	Command manually actuation of horizontal movement to outfeed position
iManualBackwards	Bool	0/1	Command manual actuation of horizontal movement to the waiting position
iB1	Bool	0/1	Sensor for detection of waiting position
iB2	Bool	0/1	Sensor for detection of low speed waiting position
iB3	Bool	0/1	Sensor for detection of low-speed prorunner position
iB4	Bool	0/1	Sensor for detection of prorunner position
iB7	Bool	0/1	Sensor for detection object dropped off
iB8	Bool	0/1	Sensor for detection object present outfeed conveyor
iB16	Bool	0/1	Sensor for detection object left outfeed conveyor
iUpStream_Ready	Bool	0/1	Communication signal of upstream transport: Transport system ready to receive object
iUpStream_Running	Bool	0/1	Communication signal of upstream transport: Transport system systems ready & transport running.

11.7 Outputs outfeed module MBC4

Exits must be declared by the customer.

Outputs provide feedback such as faults, status and communication signals with upstream transport.

Tabel 36: Outputs outfeed module MBC4

Naming	Data type	Unit	Description
oAlarm	Bool	0/1	Fault active
oAlarmCode	Int	0	Active fault code (See chapter 'Faults' for the correct overview)
oAlarmFeedback	Word	0	Active fault feedback for HMI. One bit is reserved for each fault. Fault 1 = bit 0, fault 2 = bit 1, etc.
oStatus	Int	0	See chapter 'Status Prorunner mk5' for further information.
oUpStream_ObjectReady	Bool	0/1	Communication signal to the upstream transport: Object ready for outfeed.
oUpStream_ObjectSource	Int	1..20	Communication signal to the upstream transport: Object source level.
oUpStream_ObjectInformation	UDT		Communication signal to the upstream transport: 'UDT_Prорunner_mk5_Customer_Information' Additional information that was sent with the object.

11.8 Motor interface outfeed module MBC4

Two motors are used for the outfeed module. All commands are prepared via this motor interface to handle different forms of motor actuation.

For more information, please refer to chapter '12 - Motor interface'

Tabel 37: Motor interface outfeed module MBC4

Naming	Data type	Unit	Description
xConveyor	UDT		UDT_MotorControl Motor interface roller conveyor outfeed module RDC4
xMotion	UDT		UDT_MotorControl Motor interface horizontal movement outfeed module RDC4

11.9 Status outfeed module MBC4

The status of the outfeed module is output, giving the user a clear picture of the status/movements of the outfeed module MBC4

Tabel 38: Status outfeed module MBC4

Code	Description / Cause
0	NO ACTION: Outfeed module switched off / no mode active
1	MANUAL OPERATION: Outfeed module in manual mode
3	INITIALISATION PROCEDURE ACTIVE: Outfeed module busy with initialisation procedure.
4	INITIALISATION PROCEDURE READY: Outfeed module finished with initialisation procedure.
5	CONFIGURATION CHECK: Outfeed module finished with initialisation procedure.
6	RESTART / START-UP PROCEDURE: Procedure after fault; this involves checking whether an object is present and whether it is correctly positioned.
7	FAULT: Outfeed module blocked by fault
10	WAITING: Outfeed module ready for use; feeding object in can be started
11	MOVEMENT TO INFEEED POSITION: Outfeed module busy with horizontal movement to infeed position
12	AT INFEEED POSITION: Outfeed module ready for transport of object
13	MOVEMENT TO WAITING POSITION: Outfeed module busy with horizontal movement to waiting position
14	AT WAIT POSITION: Outfeed module waiting at waiting position.
15	OBJECT ACCEPTED ON OUTFEED TRACK: Outfeed module has received object on track (sensor iB8)
16	OBJECT TRANSPORT BUSY Outfeed module in the process of moving object to the end of the conveyor track (sensor iB16 covered)
17	OBJECT AT END OF TRACK: Object at end of the conveyor track, waiting for iB16 to be uncovered.

11.10 Faults outfeed module MBC4

With the outfeed signal 'oAlarm' it is possible to indicate a fault by, for example, switching on a red lamp. With the fault code 'oAlarmCode' you can indicate this better.

With the fault word 'oAlarmFeedback' you can generate a clearly defined fault message by means of an HMI. The faults are divided over 16 bits. Code 1 = bit 0 ... Code 16 = bit 15.

Below is the list of faults that may be active with explanation/result and solution.

Tabel 39: Faults list outfeed module MBC4

Code	Description / Cause	Effect	Solution
1	Object outfeeding is taking too long.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Remove object manually from the conveyor track Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
3	Unknown object detected. Sensors triggered while the outfeed module is not busy with the corresponding step.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check for unknown objects. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
4	Object too small. iB16 triggered for too short a time. Time is calculated with the minimum product length	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
5	Object too big. iB16 triggered for too short a time. Time is calculated with the maximum product length	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. Start-up procedure of outfeed conveyor must be carried out.	Check object length. Give the reset command to reset the fault. Resetting is not possible if sensor iB8 or iB16 is covered.
8	Configuration is not entered correctly	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Check configuration settings. Make sure that everything is entered correctly and not below the minimum value.
9	Start-up procedure time-out. Sensor iB15 does not become untriggered.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Put the object down manually in the sensors; iB15 uncovered, iB7 covered. Give the reset command to reset the fault.
10	Movement to waiting position is taking too long	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Check why this is taking too long. Check adjustment/operation of the iB1 sensor Give the reset command to reset the fault.
11	Movement to waiting position is taking too long	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Check why this is taking too long. Check adjustment/operation of the iB4 sensor Give the reset command to reset the fault.

Code	Description / Cause	Effect	Solution
12	Position horizontal movement no longer correct.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked.	Check adjustment/operation of the iB1 & iB4 sensor. Give the reset command to reset the fault.
13	Fault controller/drive conveyor track.	Automatic operation is stopped. Command to PRORUNNER that outfeed module is blocked. (Depending on position)	Check operation/fault controller/drive. Make sure that it is ready for operation again. Give the reset command to reset the fault.
14	Fault controller/drive horizontal movement.	See fault 13	See fault 13
15	Initialisation procedure: object still detected on conveyor track.	Initialisation has not been completed.	Empty the conveyor track. Restart procedure by giving initialisation command again.
16	Initialisation procedure: Safe position not reached	Initialisation has not been completed.	Check the movement of the outfeed module. Check operation and adjustment of position indicator sensors. Restart procedure by giving initialisation command again.

12 Motor interface

All commands are prepared through the motor interface to handle different motor controls. The motor interface handles the communication with the motor used to control the respective function.

The motor is controlled by means of commands and settings. The motor provides necessary feedback to maintain interaction through status information.

Tabel 40: Motor control interface incoming commands → **UDT_Motorcontrol_Status**

Naming	Data type	Unit	Description
Status.AtTargetSpeed	Bool	0/1	Motor is running at desired speed
Status.MotorIsRunning	Bool	0/1	Motor is running. In the case of a controller, this is used to check whether the output stage is actually running.
Status.Blocked	Bool	0/1	Motor/Drive blocked
Status.Pulse	Bool	0/1	Motor/Drive has moved itself 1cm (10mm)
Status.ActualSpeed	Real	m/s.	Speed in meter per second on which the Prorunner is rotating
Status.ActualFrequency	Real	Hz	Number of Hertz (frequency) what the motor is currently running at.

Tabel 41: Motor control interface outgoing commands → **UDT_Motorcontrol_Command**

Naming	Data type	Unit	Description
Command.Enable	Bool	0/1	Switch on motor control
Command.Reset	Bool	0/1	Reset motor control faults
Command.QuickStop	Bool	0/1	Stop motor quickly due to faults. (0.1 sec deceleration)
Command.Energize	Bool	0/1	Pre-magnetise motor
Command.Forward	Bool	0/1	Run motor forward, or send up
Command.Reverse	Bool	0/1	Run motor backwards, or send down
Command.HighSpeed	Bool	0/1	Use high speed (if this is fixed in the motor itself)
Command.Settings.Speed	Int	Hz	Number of Hertz of rated speed (Rated = 50 Hz)
Command.Settings.Acceleration	Int	ms	Number of milliseconds to accelerate from standstill to desired speed
Command.Settings.Deceleration	Int	ms	Number of milliseconds to decelerate from current speed to standstill

We recommend to carrying out all asynchronous motors with a frequency controller because there are different requirements for control:

- Mechanism is less loaded if engine starts slower by means of acceleration
- Quickstop must be possible

Quickstop = quickest possible stop with pre-defined deceleration in controller

This should be set to less than or equal to 0.1 second.