


# FUNCTIONAL DESCRIPTION

## GroupStart

Prep.	RTG/Bo Bengtsson	2002-04-22	Function Description			No. of p.
Appr.	PA/R/ Bengt Persson	Approved	Groupstart			32
Resp. dept.						
	ABB AB		Doc. no.	Lang.	Rev. ind.	Page
			3AST 001 595D001	en	I	1

**Contents**

<b>1</b>	<b>General .....</b>	<b>4</b>
<b>2</b>	<b>Configuration .....</b>	<b>4</b>
<b>3</b>	<b>Function Block GroupStart1 &amp; GroupStart2 .....</b>	<b>5</b>
<b>4</b>	<b>GROUPSTART1 Datatypes .....</b>	<b>9</b>
4.1	GROUPSTART1_InPar .....	9
4.2	GROUPSTART1_OutPar .....	9
4.3	GROUPSTART1_Opr.....	10
<b>5</b>	<b>Function.....</b>	<b>11</b>
5.1	Basic Properties .....	11
5.2	Control Modes .....	11
5.2.1	LOCAL.....	11
5.2.2	MANUAL.....	12
5.2.3	EXTERNAL1.....	12
5.2.4	DISABLED.....	12
5.3	Ready for Start .....	13
5.4	Start-up .....	13
5.5	Interlocks.....	13
5.6	Start and Stop .....	13
5.6.1	Start and Stop Order at Different Control Modes.....	14
5.6.2	Start Order Selection .....	14
5.7	Objects.....	15
5.8	Application Example .....	15
5.9	Event handling.....	15
5.10	Interaction Window.....	16
5.10.1	Groupstart1 Interaction Window.....	16
5.10.2	General Parameters .....	16
5.10.3	Order Block.....	17
5.10.4	Alarm configuration & Event Block.....	17
5.10.5	Interlock Settings .....	18
5.10.6	Data and Timer settings.....	18
5.10.7	Texts.....	19
5.10.8	Time Delays.....	19
5.10.9	Groupstart2 Interaction Window.....	21
<b>6</b>	<b>Operator Functions.....</b>	<b>23</b>
6.1	Presentation .....	23
6.1.1	Display Elements .....	23
6.1.2	Time-logged Properties.....	24
6.2	Faceplate(Dialog) .....	25
6.3	Alarm and Event Handling .....	27
6.3.1	General.....	27
6.3.2	Alarm and Event Message .....	28
6.4	Faceplate tabs.....	29
6.4.1	Alarm and Event Blocking.....	29
6.4.2	Status .....	29
6.4.3	Timers .....	30
6.4.4	Info .....	30
<b>7</b>	<b>Step Name Texts .....</b>	<b>31</b>



## 1 General

The GroupStart is a functional unit to start several objects in a predefined sequential order and to stop them in the reversed order. It is possible to select individual start and stop delay for each object. If an object fails to start the sequence will stop at that point with an alarm announcement. Objects can be blocked and are in this case ignored in the start/stop sequence.

## 2 Configuration

The GroupStart comprises two function block types, GroupStart1 and GroupStart2 for logic functions in Control<sup>IT</sup>, a faceplate and an object display in Operate<sup>IT</sup> for operator functions. GroupStart1 is the main function block type with common control logic and connection to operator interface. It is designed for ten objects. Several function blocks of type GroupStart2 for five objects each, can be used to expand the number of objects in the group to a maximum of 125 objects. They are linked together with parameters ExtParOut and ExtParIn. The order they are linked decides the start order.

GroupStart2 cannot be used separately.

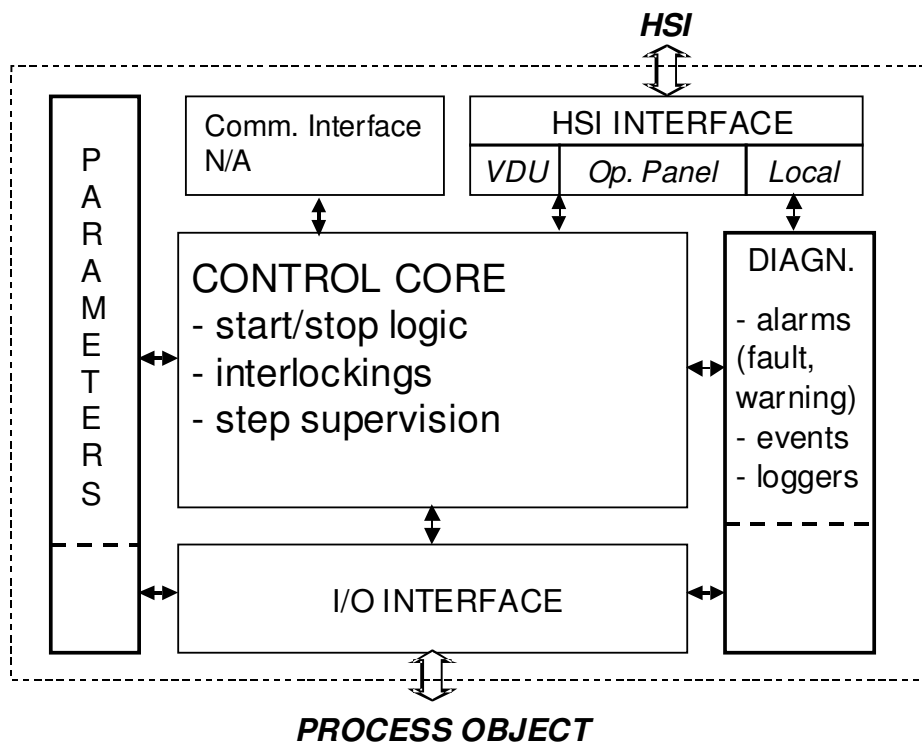


Figure 2-1. The Structure of the Functional Unit

Prep.	RTG/Bo Bengtsson	2002-04-22	Function Description		No. of p.
Appr.	PA/R/ Bengt Persson	Approved	Groupstart		32
Resp. dept.					
<b>ABB</b>		ABB AB	Doc. no.	3AST 001 595D001	Lang. en, Rev. ind. I, Page 4

### 3 Function Block GroupStart1 & GroupStart2

FUNCTION OF INPUT TERMINALS	GroupStart1		FUNCTION OF OUTPUT TERMINALS
Object name	Name	ExtParOut	Link to next group
Object description	Description	NoInt	No Interlocks
Enable object	Enable	Blk	Standby
Link from last extension group	ExtParIn	RFS	Ready for start
Process Interlock 1	IB1	Run	Running
Process Interlock 2	IB2	LocalInd	Local mode
Process Interlock 3	IB3	ManInd	Man mode
Process Interlock 4	IB4	E1Ind	E1 mode
Sequence Interlock 1	IA1	HoldInd	Hold mode
Sequence Interlock 2	IA2	Pos	Active step number
Start order in E1 mode	E1Start	Start_1	Start object 1
Stop order in external mode	Stop	Stop_1	Stop object 1
Reset	Reset	Start_2	Start object 2
Order mode to Local	LEnbl	Stop_2	Stop object 2
Start order in Local mode	L1	Start_3	Start object 3
Hold order in Local mode	LHold	Stop_3	Stop object 3
Local Stop order	LStop	Start_4	Start object 4
Order mode to Man	SeqMan	Stop_4	Stop object 4
Order mode to E1	SeqE1	Start_5	Start object 5
Block Object 1	Blk_1	Stop_5	Stop object 5
Object 1 ready for start	RFS_1	Start_6	Start object 6
Object 1 running	Run_1	Stop_6	Stop object 6
Block Object 2	Blk_2	Start_7	Start object 7
Object 2 ready for start	RFS_2	Stop_7	Stop object 7
Object 2 running	Run_2	Start_8	Start object 8
Block Object 3	Blk_3	Stop_8	Stop object 8
Object 3 ready for start	RFS_3	Start_9	Start object 9
Object 3 running	Run_3	Stop_9	Stop object 9
Block Object 4	Blk_4	Start_10	Start object 10
Object 4 ready for start	RFS_4	Stop_10	Stop object 10
Object 4 running	Run_4	OutPar	Out Parameter
Block Object 5	Blk_5	Opr	Operator order
Object 5 ready for start	RFS_5		
Object 5 running	Run_5		
Block Object 6	Blk_6		
Object 6 ready for start	RFS_6		
Object 6 running	Run_6		
Block Object 7	Blk_7		
Object 7 ready for start	RFS_7		
Object 7 running	Run_7		
Block Object 8	Blk_8		
Object 8 ready for start	RFS_8		
Object 8 running	Run_8		
Block Object 9	Blk_9		
Object 9 ready for start	RFS_9		
Object 9 running	Run_9		
Block Object 10	Blk_10		
Object 10 ready for start	RFS_10		
Object 10 running	Run_10		
Block alarm	AlcBlk		
Acknowledge alarm	AlarmAck		
In Parameter	InPar		
Event name	EventName		

Figure 3-1. GroupStart1 Function Block, Complete symbol

Table 3-1 below illustrates the default properties of each terminal of the GroupStart1 function block.

Name	Data Type	Attributes	Direction	FD Port	Initial value	Description
Name	string	coldretain	in	yes	'GroupStart1'	Object name
Description	string	coldretain	in	yes	'Descr'	Object description
Enable	bool	coldretain	in	yes	true	Enable object
ExtParIn	dint	retain	in	yes		Link from last extension group
IB1	bool	retain	in	yes	true	Process Interlock 1
IB2	bool	retain	in	yes	true	Process Interlock 2
IB3	bool	retain	in	yes	true	Process Interlock 3
IB4	bool	retain	in	yes	true	Process Interlock 4
IA1	bool	retain	in	yes	true	Sequence Interlock 1
IA2	bool	retain	in	yes	true	Sequence Interlock 2
E1Start	bool	retain	in	yes		Start order in E1 mode
Stop	bool	retain	in	yes		Stop order in external mode
Reset	bool	retain	in	yes		Reset
LEnbl	bool	retain	in	yes		Order mode to Local
L1	bool	retain	in	yes		Start order in Local mode
LHold	bool	retain	in	yes		Hold order in Local mode
LStop	bool	retain	in	yes		Local Stop order
SeqMan	bool	retain	in	yes		Order mode to Man
SeqE1	bool	retain	in	yes		Order mode to E1
Blk_1	bool	retain	in	yes	true	Block Object 1
RFS_1	bool	retain	in	yes		Object 1 ready for start
Run_1	bool	retain	in	yes		Object 1 running
Blk_2	bool	retain	in	yes	true	Block Object 2
RFS_2	bool	retain	in	yes		Object 2 ready for start
Run_2	bool	retain	in	yes		Object 2 running
Blk_3	bool	retain	in	yes	true	Block Object 3
RFS_3	bool	retain	in	yes		Object 3 ready for start
Run_3	bool	retain	in	yes		Object 3 running
Blk_4	bool	retain	in	yes	true	Block Object 4
RFS_4	bool	retain	in	yes		Object 4 ready for start
Run_4	bool	retain	in	yes		Object 4 running
Blk_5	bool	retain	in	yes	true	Block Object 5
RFS_5	bool	retain	in	yes		Object 5 ready for start
Run_5	bool	retain	in	yes		Object 5 running
Blk_6	bool	retain	in	yes	true	Block Object 6
RFS_6	bool	retain	in	yes		Object 6 ready for start
Run_6	bool	retain	in	yes		Object 6 running
Blk_7	bool	retain	in	yes	true	Block Object 7
RFS_7	bool	retain	in	yes		Object 7 ready for start
Run_7	bool	retain	in	yes		Object 7 running
Blk_8	bool	retain	in	yes	true	Block Object 8
RFS_8	bool	retain	in	yes		Object 8 ready for start
Run_8	bool	retain	in	yes		Object 8 running
Blk_9	bool	retain	in	yes	true	Block Object 9
RFS_9	bool	retain	in	yes		Object 9 ready for start
Run_9	bool	retain	in	yes		Object 9 running
Blk_10	bool	retain	in	yes	true	Block Object 10
RFS_10	bool	retain	in	yes		Object 10 ready for start
Run_10	bool	retain	in	yes		Object 10 running
AlcBlk	bool	retain	in	yes		Block alarm
AlarmAck	bool	retain	in	yes		Acknowledge alarm
InPar	GroupStart1_InPar	by_ref	in	yes		In Parameter
EventName	string	coldretain	in	yes	'  GS1_'	Event name
ExtParOut	dint	retain	out	yes		Link to next group
NoInt	bool	retain	out	yes		No Interlocks
Blk	bool	retain	out	yes		Standby
RFS	bool	retain	out	yes		Ready for start
Run	bool	retain	out	yes		Running

Name	Data Type	Attributes	Direction	FD Port	Initial value	Description
LocalInd	bool	retain	out	yes		Local mode
ManInd	bool	retain	out	yes		Man mode
E1Ind	bool	retain	out	yes		E1 mode
HoldInd	bool	retain	out	yes		Hold mode
Pos	dint	retain	out	yes		Active step number
Start_1	bool	retain	out	yes		Start object 1
Stop_1	bool	retain	out	yes		Stop object 1
Start_2	bool	retain	out	yes		Start object 2
Stop_2	bool	retain	out	yes		Stop object 2
Start_3	bool	retain	out	yes		Start object 3
Stop_3	bool	retain	out	yes		Stop object 3
Start_4	bool	retain	out	yes		Start object 4
Stop_4	bool	retain	out	yes		Stop object 4
Start_5	bool	retain	out	yes		Start object 5
Stop_5	bool	retain	out	yes		Stop object 5
Start_6	bool	retain	out	yes		Start object 6
Stop_6	bool	retain	out	yes		Stop object 6
Start_7	bool	retain	out	yes		Start object 7
Stop_7	bool	retain	out	yes		Stop object 7
Start_8	bool	retain	out	yes		Start object 8
Stop_8	bool	retain	out	yes		Stop object 8
Start_9	bool	retain	out	yes		Start object 9
Stop_9	bool	retain	out	yes		Stop object 9
Start_10	bool	retain	out	yes		Start object 10
Stop_10	bool	retain	out	yes		Stop object 10
OutPar	GroupStart1 _OutPar	by_ref	out	yes		Out Parameter
Opr	GroupStart1 _Opr	by_ref	out	yes		Operator order

FUNCTION OF INPUT TERMINALS	GroupStart2		FUNCTION OF OUTPUT TERMINALS
Object Name	Name	ExtParOut	Extension Parameter Output
Object Description	Description	Start_1	Start pulse for object 1
Extension Parameter Input	ExtParIn	Stop_1	Stop pulse for object 1
Blocked input for object 1	Blk_1	Start_2	Start pulse for object 2
Ready for start object 1	RFS_1	Stop_2	Stop pulse for object 2
Run input for object 1	Run_1	Start_3	Start pulse for object 3
Blocked input for object 2	Blk_2	Stop_3	Stop pulse for object 3
Ready for start object 2	RFS_2	Start_4	Start pulse for object 4
Run input for object 2	Run_2	Stop_4	Stop pulse for object 4
Blocked input for object 3	Blk_3	Start_5	Start pulse for object 5
Ready for start object 3	RFS_3	Stop_5	Stop pulse for object 5
Run input for object 3	Run_3		
Blocked input for object 4	Blk_4		
Ready for start object 4	RFS_4		
Run input for object 4	Run_4		
Blocked input for object 5	Blk_5		
Ready for start object 5	RFS_5		
Run input for object 5	Run_5		

Figure 3-2. GroupStart2 Function Block, Complete symbol

Table 3-2 below illustrates the default properties of each terminal of the GroupStart2 function block.

Name	Data Type	Attributes	Direction	FD Port	Initial value	Description
Name	string	coldretain	in	yes	'GroupStart2'	Object name
Description	string	coldretain	in	yes	'Descr'	Object description
ExtParIn	dint	retain	in	yes		Link from previous group
ExtParOut	dint	retain	out	yes		Link to next group
Blk_1	bool	retain	in	yes	true	Block Object 1
RFS_1	bool	retain	in	yes		Object 1 ready for start
Run_1	bool	retain	in	yes		Object 1 running
Start_1	bool	retain	out	yes		Start object 1
Stop_1	bool	retain	out	yes		Stop object 1
Blk_2	bool	retain	in	yes	true	Block Object 2
RFS_2	bool	retain	in	yes		Object 2 ready for start
Run_2	bool	retain	in	yes		Object 2 running
Start_2	bool	retain	out	yes		Start object 2
Stop_2	bool	retain	out	yes		Stop object 2
Blk_3	bool	retain	in	yes	true	Block Object 3
RFS_3	bool	retain	in	yes		Object 3 ready for start
Run_3	bool	retain	in	yes		Object 3 running
Start_3	bool	retain	out	yes		Start object 3
Stop_3	bool	retain	out	yes		Stop object 3
Blk_4	bool	retain	in	yes	true	Block Object 4
RFS_4	bool	retain	in	yes		Object 4 ready for start
Run_4	bool	retain	in	yes		Object 4 running
Start_4	bool	retain	out	yes		Start object 4
Stop_4	bool	retain	out	yes		Stop object 4
Blk_5	bool	retain	in	yes	true	Block Object 5
RFS_5	bool	retain	in	yes		Object 5 ready for start
Run_5	bool	retain	in	yes		Object 5 running
Start_5	bool	retain	out	yes		Start object 5
Stop_5	bool	retain	out	yes		Stop object 5

## 4 GROUPSTART1 Datatypes

### 4.1 GROUPSTART1\_InPar

Name	Data Type	Attributes	Initial value	ISP value	Description
Class	dint	coldretain	500		AE class
Severity	dint	coldretain	1000		AE severity
InitMode	dint	coldretain	5		Init mode (5 = Man ; 7 = E1)
ManBlk	bool	coldretain	false		Block operator order Man mode
E1Blk	bool	coldretain	false		Block operator order E1 mode
StartBlk	bool	coldretain	false		Block operator order Start command
StopBlk	bool	coldretain	false		Block operator order Stop command
ResetBlk	bool	coldretain	false		Block operator order Reset command
HoldBlk	bool	coldretain	false		Block operator order Hold command
EStopBlk	bool	coldretain	false		Block operator order EStop command
AlcBlkEvBlk	bool	coldretain	true		Block event for AlcBlk
LEnblEvBlk	bool	coldretain	true		Block event for LEnbl
SeqManEvBlk	bool	coldretain	true		Block event for SeqMan
SeqE1EvBlk	bool	coldretain	true		Block event for SeqE1
ResetEvBlk	bool	coldretain	true		Block event for Reset
ExtStartEvBlk	bool	coldretain	true		Block event for ExtStart
ExtStopEvBlk	bool	coldretain	true		Block event for ExtStop
LHoldEvBlk	bool	coldretain	true		Block event for LHold
IABlockNotExtMode	bool	coldretain	true		IA blocked when not in E1 or E2 mode
IA1	IAInParType1	coldretain			Configuration for IA1
IA2	IAInParType1	coldretain			Configuration for IA2
IB1	IBInParType3	coldretain			Configuration for IB1
IB2	IBInParType3	coldretain			Configuration for IB2
IB3	IBInParType3	coldretain			Configuration for IB3
IB4	IBInParType3	coldretain			Configuration for IB4
AEConfigSeq	dint	coldretain	1		AE configuration for sequence time out alarm
AEConfigStep	dint	coldretain	1		AE configuration for step time out alarm
MaxSeqTime	time	coldretain	0d0h10m 0s0ms		Maximum sequence time
EStopTime	time	coldretain	0d0h0m5 s0ms		EStop delay time
Step1	GroupStepTime	coldretain			Step 1 delay time
Step2	GroupStepTime	coldretain			Step 2 delay time
Step3	GroupStepTime	coldretain			Step 3 delay time
Step4	GroupStepTime	coldretain			Step 4 delay time
Step5	GroupStepTime	coldretain			Step 5 delay time
Step6	GroupStepTime	coldretain			Step 6 delay time
Step7	GroupStepTime	coldretain			Step 7 delay time
Step8	GroupStepTime	coldretain			Step 8 delay time
Step9	GroupStepTime	coldretain			Step 9 delay time
Step10	GroupStepTime	coldretain			Step 10 delay time

### 4.2 GROUPSTART1\_OutPar

Name	Data Type	Attributes	Initial value	ISP value	Description
AlarmBlk	bool	retain			Alarm blocked
IntlkBlk	bool	retain			Interlock blocked
IntlkBlkActive	bool	retain			Interlock blocked active
EnOverrideAll	bool	retain			Override All button enabled
Mode	dint	retain			Active mode
NormalMode	bool	retain			Normal mode (Active mode = Init mode)
IA1Ind	bool	retain			IA1 interlocked
IA2Ind	bool	retain			IA2 interlocked

Name	Data Type	Attributes	Initial value	ISP value	Description
IB1Ind	bool	retain			IB1 interlocked
IB2Ind	bool	retain			IB2 interlocked
IB3Ind	bool	retain			IB3 interlocked
IB4Ind	bool	retain			IB4 interlocked
Seq	AlarmInd	retain			Alarm indication for sequence time out
Step	AlarmInd	retain			Alarm indication for step time out
NoOfSteps	dint	retain			Number of steps
SeqStart	bool	retain			Group start in starting sequence
SeqStop	bool	retain			Group start in stopping sequence
MaxStepTime	time	retain			Maximum step time
StepTimeElapsed	time	retain			Step time elapsed
SeqTimeElapsed	time	retain			Sequence time elapsed
TransitionTime	time	retain			Transition time
TransitionTimeElapsed	time	retain			Transition time elapsed

## 4.3 GROUPSTART1\_Opr

Name	Data Type	Attributes	Initial value	ISP value	Description
BlockAlarm	bool	retain			Operator block alarms
Man	bool	retain			Operator order Manual mode
E1	bool	retain			Operator order E1 mode
Start	bool	retain			Operator order Start command
Stop	bool	retain			Operator order Stop command
Reset	bool	retain			Operator order Reset command
Hold	bool	retain			Operator order Hold command
EStop	bool	retain			Operator order EStop command
OverrideAll	bool	retain			Operator override all interlocks
IB1Override	bool	retain			Operator override IB1 interlock
IB2Override	bool	retain			Operator override IB2 interlock
IB3Override	bool	retain			Operator override IB3 interlock
IB4Override	bool	retain			Operator override IB4 interlock
IA1Override	bool	retain			Operator override IA1 interlock
IA2Override	bool	retain			Operator override IA2 interlock

## 5 Function

### 5.1 Basic Properties

The GroupStart functional unit is designed for starting and/or stopping of a group of process objects.

GroupStart unit consists of the following basic functions.

- Evaluation of interlocks
- Control of start/stop
- Supervision of sequence and step time
- Control from the local operator's panel
- Manual/External running
- Supervision of operations via Operate<sup>IT</sup> Operator Station

### 5.2 Control Modes

GroupStart is intended for control from Operate<sup>IT</sup> Operator Station, i.e. from a central control room. However, it is also possible to select other modes of control and thus control GroupStart from other locations.

All the control modes can be selected from the central operator station. The LOCAL control mode may also be selected from the local control station.

The different modes of control are as follows:

- Local
- Manual
- External 1

By selecting control mode, the operator decides from which location the groupstart is to be controlled. All control modes may be selected through the dialog in the Operate<sup>IT</sup> Operator Station. The LOCAL control mode may also be selected from the local control station through the Function Block input :LEnbl, provided that the current control mode is not DISABLE.

For the different control modes, GroupStart is controlled as follows. For a summary of the control modes, see Table 4-1.

#### 5.2.1 LOCAL

The groupstart is controlled from a local desk or panel. Control from other locations is blocked.

LOCAL control mode, like other points of control, may be selected from Operate<sup>IT</sup> Operator Station. In addition, LOCAL control mode may be selected from the local control panel through the input :LEnbl as follows:

:LEnbl = 1: The control mode is LOCAL, as long as the Operate<sup>IT</sup> Operator Station operator does not request STAND BY. All other control modes are blocked and :LEnbl must be released before a mode change can occur.

LEnbl -> 0: GroupStart resumes the previous or the latest control mode called for from Operate<sup>IT</sup> Operator Station.

The operator controls the groupstart through the inputs :L1, :LHold and :LStop.

- :L1 Local start order
- :LHold Local hold order
- :LStop Local stop order (active low)

### 5.2.2 MANUAL

The groupstart is controlled from Operate<sup>IT</sup> Operator Station. Control from other locations is blocked. This is the default control mode. To start, the operator has to press the key START. To stop or hold the groupstart, the key STOP or HOLD has to be pressed.

### 5.2.3 EXTERNAL1

The groupstart is controlled from external signals, e.g. from a process signal. Control from other locations is blocked.

The function is enabled by input :SeqE1.

The process controls the groupstart through the inputs :E1Start and :Stop. The operator is able to stop the groupstart by issuing stop order from Operate<sup>IT</sup> Operator Station. The control mode then changes to Manual, to prevent the groupstart from being restarted.

- :Stop** Stop order (active low)
- :E1Start** Start order

### 5.2.4 DISABLED

The groupstart is stopped and all orders to it are blocked. Used to take the groupstart Out of Service.

DISABLED mode can be selected at the operator station, if a start or stop sequence is not in progress. In this mode the GroupStart functions are disabled.

Mode changes to other modes than Standby can be made when start or stop sequence is running.

The following table shows command signals in priority order and control modes selected (shadowed) under different conditions.

Commands		Mode selected			
Signal	Source	Standby	Manual	E1	Local
Standby 1)	Operator				
Lenbl	Program				
Man	Operator				
SeqMan	Program				
E1	Operator				
SeqE1	Program				
Local	Operator				

Table 4-1. Control modes selection and priority.

- 1) Standby cannot be selected if input parameter LEnbl is True.

### 5.3 Ready for Start

Ready for start means that all the interlocks are satisfied, that there are no alarms and that the control is not DISABLED.

Ready for start is indicated by the GroupStart displays in the Operate<sup>IT</sup> Operator Station and by the output terminal :RFS (Ready For Start).

### 5.4 Start-up

An initialization phase begins at start of the system. The result of this becomes:

The outputs are reset (to zero) except for the ManInd terminal which is set (to 1).

Manual is the default control mode at system initialization. With an additional circuit it is possible to force the control mode to other modes at system initialization.

### 5.5 Interlocks

The groupstart control can be interlocked by signals from the process and also by signals from the control logic. The interlocks are divided into four groups with the following designations and functions.

**Process Interlocks.** There are 2 normal process interlocks, :IB1 to :IB2. Both interlocks can be configured to accept blocking by the operator. Both interlocks can also be configured as start interlocks. A start interlock will prevent the groupstart from starting, but does not stop a running groupstart.

**External (Sequence) Interlocks.** Interlocks on terminals :IA1 and :IA2 are used if the object is to be interlocked against other objects such as a pump motor or a group start or other conditions in an automatic sequence. The interlocks cannot be blocked and the interlocks is not included in the conditions for indication of Ready for start.

An interlocking is active when the input is 0 (FALSE).

### 5.6 Start and Stop

Start and stop commands for GroupStart may originate from dialog with Operate<sup>IT</sup> Operator Station or from Function Block inputs, depending on the control mode selected (See Section 5.2, and Table 4.1).

When all objects are ready for start, the GroupStart can be started. Blocked objects are ignored. The first object is started immediately and after start acknowledge the GroupStart will wait a predefined delay time until next object will start. When the last object has started the output parameter Run is set, and the GroupStart goes to its rest position. In the case a blocked object has later become ready for start, it is possible to make a new startorder of the GroupStart, to start even these objects. Objects, already started will be bypassed without any delay.

If an object doesn't start, an alarm is given and the GroupStart will stop at this step in a Hold mode. Hold can also be ordered by the operator or locally via an input parameter. Hold mode is left with a new start or stop order, which will start the sequence from the beginning.

Order Instant Stop interrupts start and stop sequences and sets all Stop\_x output parameters for four seconds.

### 5.6.1 Start and Stop Order at Different Control Modes

The table below describes the commands which can be given at the different control modes of GroupStart.

Order	Local	Manual	Ext1
Input L1	Start	-	-
Input LHold	Hold	-	-
Input LStop	Stop	-	-
Operator Start	-	Start	-
Operator Hold	-	Hold	-
Operator Stop	-	Stop	Stop
Input E1Start	-	-	Start
Stop	-	-	Stop

Table 4-2. Relation between commands and control modes

### 5.6.2 Start Order Selection

The Function Block GroupStart has ten start and stop order outputs:

:Start\_1 ...:Start\_10 Start order 1 to 10

:Stop\_1 ...:Stop\_10 Stop order 1 to 10

The activation of the output is determined as follows from the different points of control.

#### 5.6.2.1 Control mode LOCAL

FB input :L1, Start

FB input :LHold, Hold

FB input :LStop, Stop

#### 5.6.2.2 Control mode MAN

The operator issues a Start order from Operate<sup>IT</sup> Operator Station.

#### 5.6.2.3 Control mode Ext1

A superior control program sets the input :E1Start to issue a new start order.

See the table below.

:E1Start	Stop	Descr
0	0	No change.
1	0	No change.
1	1	Start order

#### 5.6.2.4 Control mode DISABLE

The groupstart cannot be started.

### 5.7 Objects

Each object is connected to three input and two output parameters of the GroupStart.

Input :Blk\_x is used if an object is blocked or a step of the GroupStart is not used. If the input is True, this step is ignored during start and stop and will be bypassed without any step delay.

Input :RFS\_x is used to indicate that the object is ready for start. All objects, that are not blocked, must be ready for start. This is indicated by the output parameter RFS.

Input :Run\_x is used to indicate that the object is running. When it is set to True, the GroupStart waits selected delay time, and then steps to the next object. If the object is already running, next step will be entered without a step delay.

Outputs :Start\_x and :Stop\_x give order pulses to the objects. Outputs will stay True until the object has started or stopped or until the step alarm delay has elapsed.

A feature is that a GroupStart can be an object to another GroupStart, by connecting the object parameters to the parameters Blk, RFS, Run, E1Start and Stop of the subordinate GroupStart1.

### 5.8 Application Example

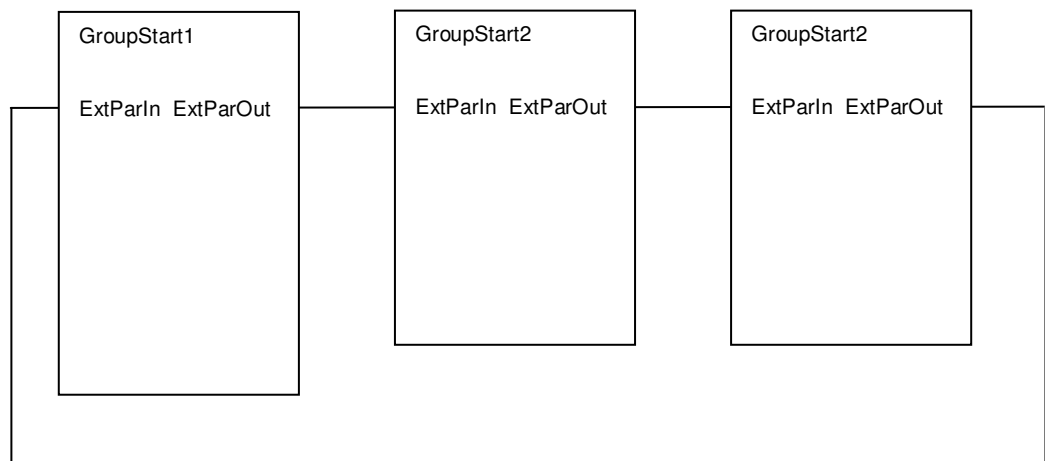


Figure 4-1. Example of linking GroupStart1 and two GroupStart2 function blocks via ExtParOut and ExtParIn.

### 5.9 Event handling

Event are generated for status change on the signals defined in interaction window in chapter 4.11.4.

The layout of the event is described in chapter 5.3.2

All Operator Events are reported by Audit Trail Functionality and not included in the FunctionBlock.

The individual text string for each event is stored in the Alarm and Event Translator aspect. This text can be NLS handled.

## 5.10 Interaction Window

The interaction window is available in the Control<sup>IT</sup> Control Builder. The interaction window is an engineering aid used to simplify configuration and blocking of signals not available on the faceplates. Changes to values in the Interaction window are only available in 'Online' mode in Control<sup>IT</sup>.

### 5.10.1 Groupstart1 Interaction Window

Interaction window overview. Name and description are shown. The buttons are links to sub-windows.

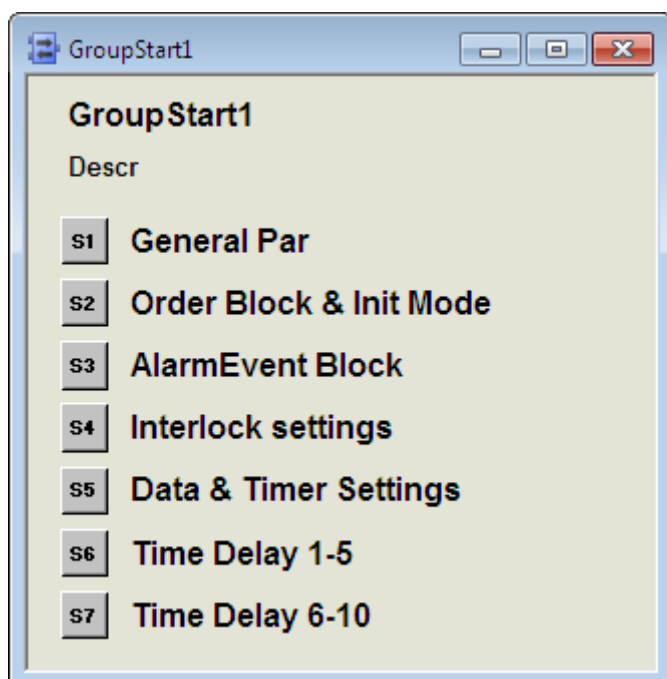


Figure 4-1 Main Interaction Window.

### 5.10.2 General Parameters

“Class” defines the process section or area in which alarms are grouped. By utilizing class the alarms can be filtered. Valid values are user defined. A suggestion would be to use mill area numbers as class values.

“Severity” defines the alarm priority for general alarms. The severity for alarms is entered in window “Alarm & Event”. Valid values are 1 –1000 where 1000 is the highest priority.

The screenshot shows a window titled "GroupStart1" with the following fields:

- Name :
- Description :
- Class :
- Severity :

Figure 4-2 General Parameters.

### 5.10.3 Order Block

Blocking of operator order are entered in this window.

The screenshot shows a window titled "GroupStart1" with two columns of settings:

Order Block	Init Mode
<input type="checkbox"/> Manual Mode	<input checked="" type="radio"/> Manual Mode
<input type="checkbox"/> E1 Mode	<input type="radio"/> E1 Mode
<input type="checkbox"/> Start	
<input type="checkbox"/> Stop	
<input type="checkbox"/> Reset	
<input type="checkbox"/> Fast Stop	
<input type="checkbox"/> Hold	

Figure 4-3 Order Block.

### 5.10.4 Alarm configuration & Event Block

Configuration of Alarm and Event are entered in this window. For Alarm Configuration the following values are valid

- 0 No Alarm or Event are generated
- 1 Alarm and Event are generated
- 2 Event is generated

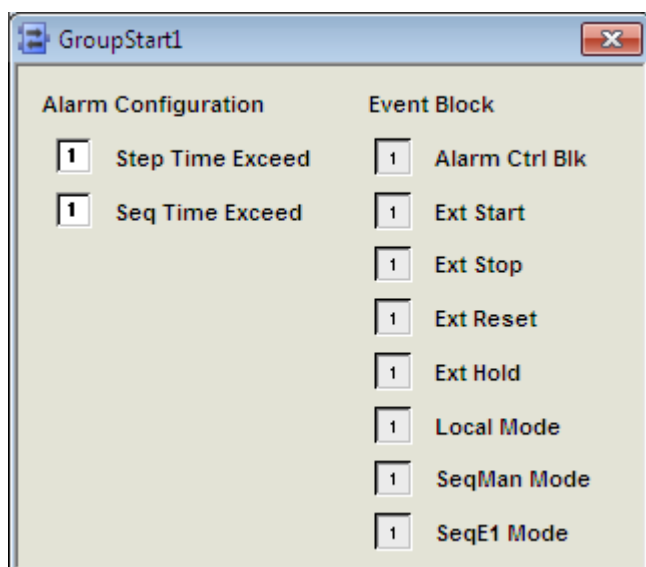


Figure 4-4 Alarm & Event Block.

### 5.10.5 Interlock Settings

The different settings for interlocks are entered in this window.

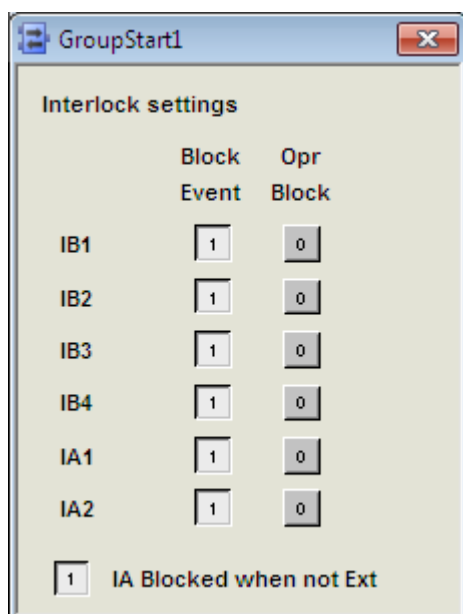


Figure 4-5 Interlock Settings.

### 5.10.6 Data and Timer settings

Setting of maximum sequence time and pulse length for e-stop is entered in this window. The active step number is also indicated.

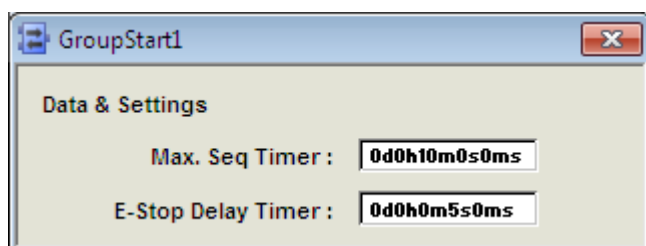
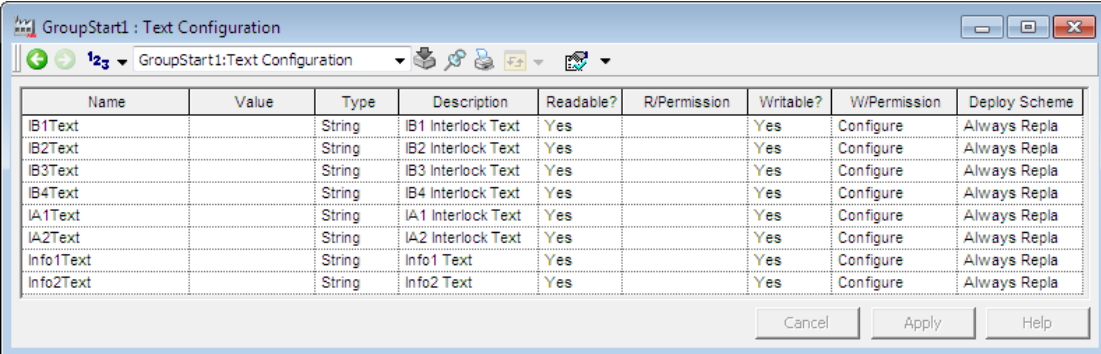


Figure 4-6 Data & Timer settings.

### 5.10.7 Texts

The different interlock and information texts are entered in the aspect Text Properties. The length of the text is limited to about 60 characters, by the size of presentation element in the Interlock Display.



Name	Value	Type	Description	Readable?	R/Permission	Writable?	W/Permission	Deploy Scheme
IB1Text		String	IB1 Interlock Text	Yes		Yes	Configure	Always Repla
IB2Text		String	IB2 Interlock Text	Yes		Yes	Configure	Always Repla
IB3Text		String	IB3 Interlock Text	Yes		Yes	Configure	Always Repla
IB4Text		String	IB4 Interlock Text	Yes		Yes	Configure	Always Repla
IA1Text		String	IA1 Interlock Text	Yes		Yes	Configure	Always Repla
IA2Text		String	IA2 Interlock Text	Yes		Yes	Configure	Always Repla
Info1Text		String	Info1 Text	Yes		Yes	Configure	Always Repla
Info2Text		String	Info2 Text	Yes		Yes	Configure	Always Repla

Figure 4-7 Texts.

### 5.10.8 Time Delays

The different time delays for start, stop and alarm is entered in these windows.

The screenshot shows a software window titled "GroupStart1" with a close button in the top right corner. The window content is titled "Time Delay Setting Step 1 to 5". It contains a list of settings for five steps, each with three delay fields: Start Delay, Stop Delay, and Alarm Delay. The values are as follows:

Step	Start Delay	Stop Delay	Alarm Delay
Step 1	0d0h0m1s0ms	0d0h0m1s0ms	0d0h0m5s0ms
Step 2	0d0h0m1s0ms	0d0h0m1s0ms	0d0h0m5s0ms
Step 3	0d0h0m1s0ms	0d0h0m1s0ms	0d0h0m5s0ms
Step 4	0d0h0m1s0ms	0d0h0m1s0ms	0d0h0m5s0ms
Step 5	0d0h0m1s0ms	0d0h0m1s0ms	0d0h0m5s0ms

GroupStart1

Time Delay Setting Step 6 to 10

Step 6 Start Delay :	0d0h0m1s0ms
Step 6 Stop Delay :	0d0h0m1s0ms
Step 6 Alarm Delay :	0d0h0m5s0ms
Step 7 Start Delay :	0d0h0m1s0ms
Step 7 Stop Delay :	0d0h0m1s0ms
Step 7 Alarm Delay :	0d0h0m5s0ms
Step 8 Start Delay :	0d0h0m1s0ms
Step 8 Stop Delay :	0d0h0m1s0ms
Step 8 Alarm Delay :	0d0h0m5s0ms
Step 9 Start Delay :	0d0h0m1s0ms
Step 9 Stop Delay :	0d0h0m1s0ms
Step 9 Alarm Delay :	0d0h0m5s0ms
Step 10 Start Delay :	0d0h0m1s0ms
Step 10 Stop Delay :	0d0h0m1s0ms
Step 10 Alarm Delay :	0d0h0m5s0ms

Figure 4-8 Time Delays.

#### 5.10.9 Groupstart2 Interaction Window

The different time delays for start, stop and alarm for the extension group is entered in this windows.

GroupStart2

Descr

**s1** Time Delay 1-5

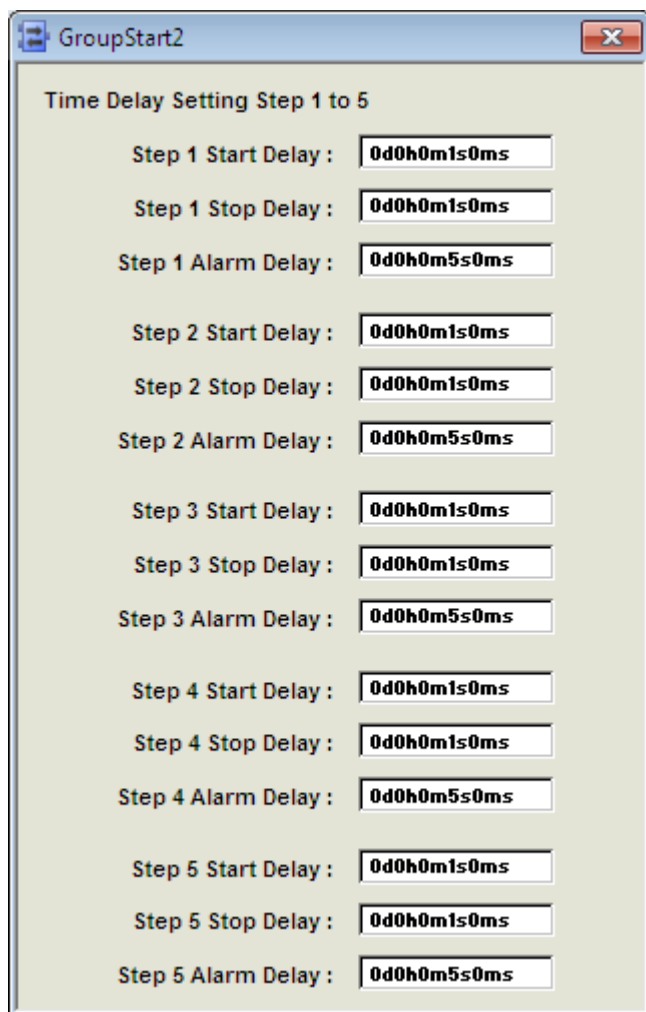


Figure 4-9 Groupstart 2, Time Delays.

## 6 Operator Functions

The Operator functions are divided in principle into 4 parts:

- Presentation (Display elements, Time logged properties)
- Faceplate (Dialog)
- Alarm and Event handling

### 6.1 Presentation

#### 6.1.1 Display Elements

Display elements, which can be used for different display types, are available for use in the functional unit GroupStart.

The display elements show the status and the controls of the process with different degrees of detail and are intended for the following displays:

- Object display
- Process display
- Interlock display

Examples of different display elements which could be used are given in the following sections.

##### 6.1.1.1 Object display



Figure 5-1 Object Display.

6.1.1.2 Process display



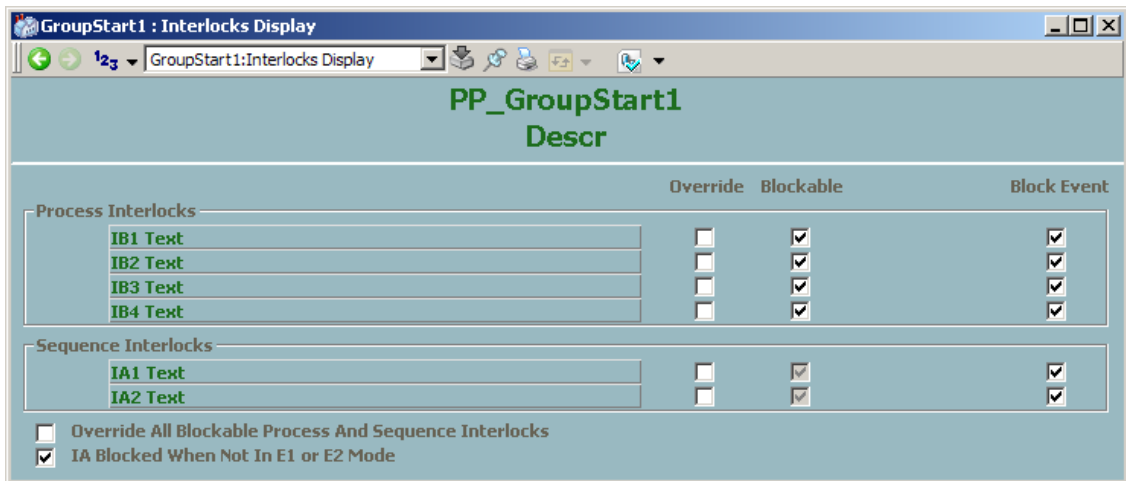
Figure 5-2 Process Display Element.

6.1.1.3 Interlock display

This display shows the actual status of all Interlock. The operator can override individual interlocks or all interlock.

Interlocks that can be override must be set to Blockable. This can be done from this display if the user has permission Configure or from the Interaction Window see chapter 5.10.5.

Start Interlock, Block Event and IA Blocked when no in E1 or E2 mode are parameters that can be set from this display if the user has Permission Configure or from Interaction Window.



6.1.2 Time-logged Properties

Measured values stored can be presented graphically in the form of curves on the display screen. Such a display, a **Trend display**, can consist of 1- 4 curves. All properties for the object GroupStart are available to be logged on the trend curves.

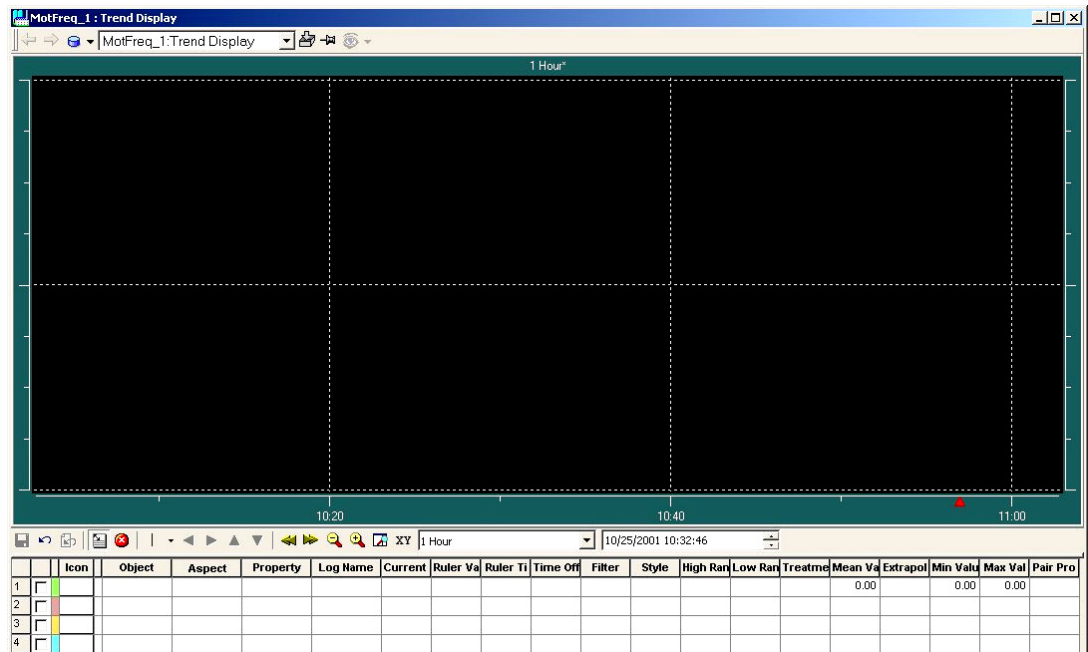


Figure 5-3 Trend Curve

## 6.2 Faceplate(Dialog)

The display screen is supplemented with a mouse and keyboard for operator communication with the functional unit/object.

By using Operate<sup>IT</sup> Operator Station the operator can view and control the process through faceplates. The dialogue consists of buttons, indicators and graphic presentations within a Faceplate. A faceplate has three levels of dialogue, which are presented by the following three runtime views:

- Reduced Faceplate, where the size and contents typically have been optimized to cover most of the normal process operator actions. Minimum dialogue. This is the default view.
- Faceplate, which typically covers all normal process operator actions. This view is disabled as default.
- Extended Faceplate, with functions and information intended for the process engineer or the advanced operator. Maximum dialogue.

The figures 5-4 to 5-11 below and overleaf illustrate the various presentations of the faceplate.



Figure 5-4 Reduced Faceplate

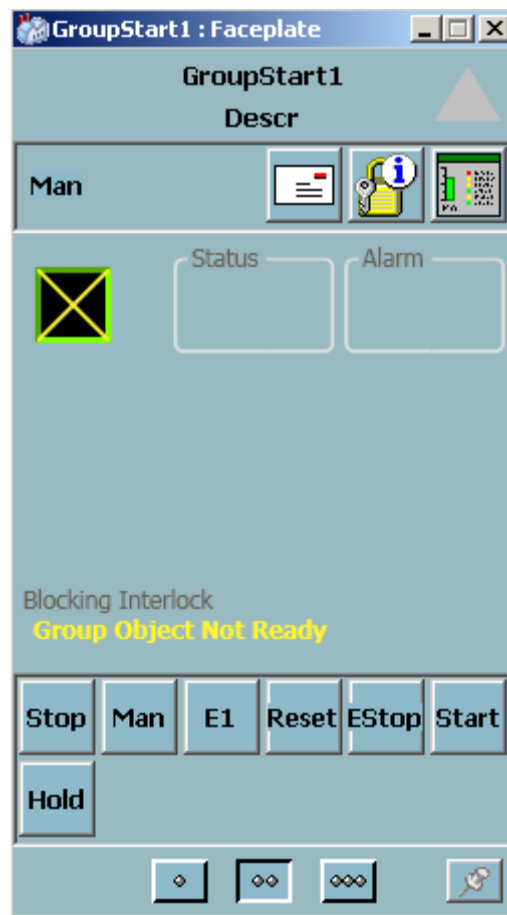


Figure 5-5 Faceplate

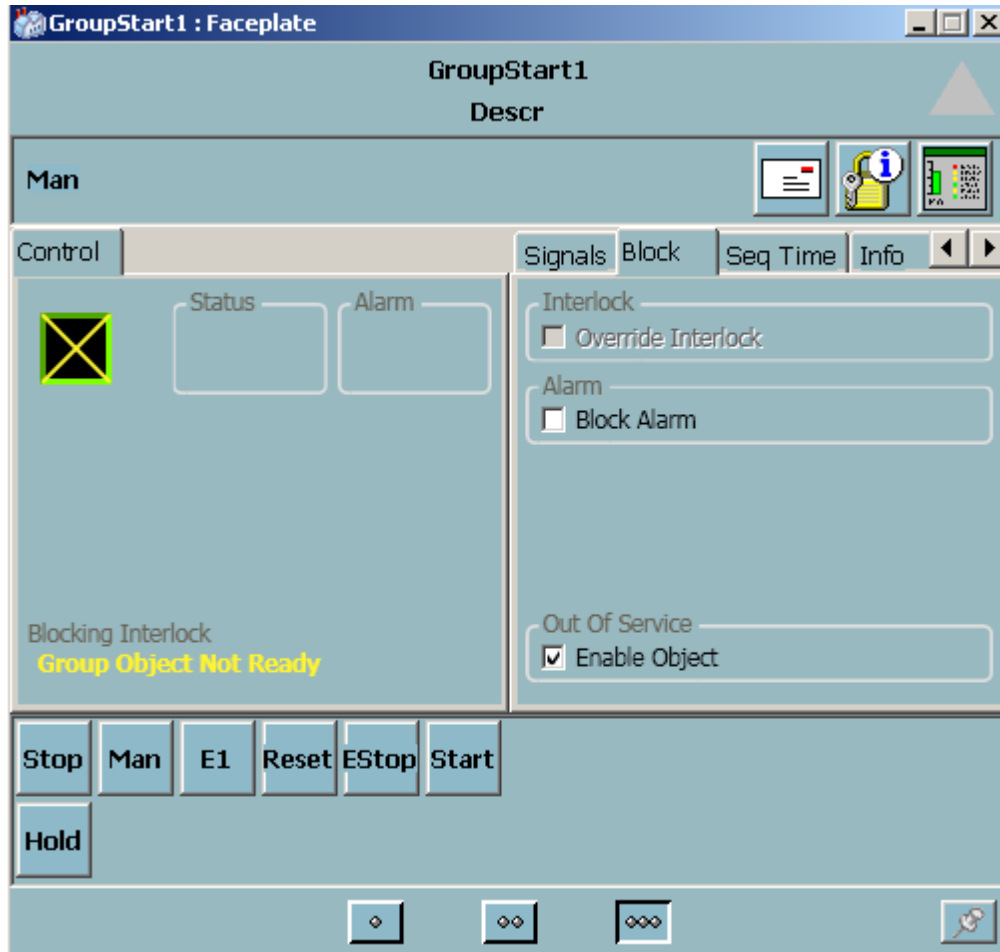


Figure 5-6 Extended Faceplate

## 6.3 Alarm and Event Handling

### 6.3.1 General

This section contains a description of all alarms and events in the functional unit GroupStart.

If an object doesn't start, any step or the sequence itself takes too long time, an alarm and an event is generated and can be viewed on the Operate<sup>IT</sup> Operator Station. The alarms are indicated in the faceplate, object display and in the alarm and event list.

The time stamping of the alarm is done when the function block is executed.

The possibilities of the operator to block alarms are shown under the heading 'Event and alarm blocking' below.

The alarm limits for GroupStart can be controlled individually

Event Time	Object Name	Object Description	Condition	Message Description
02-05-24 03:46:16:763	264M500.RUN	Pulp Mixer Running	Status	Alarm
02-05-24 01:13:04:785	192.168.0.51-0.11.5		HWError	For info see 'Errors and
02-05-24 00:22:18:784	192.168.0.51-0.11.4		HWError	For info see 'Errors and
02-05-23 23:32:31:458	500F1920	Washing Pulp Flow	AI_Err	Alarm
02-05-23 23:32:31:458	500F1920	Washing Pulp Flow	MV_L1	Alarm
02-05-23 23:32:31:458	500F1920	Washing Pulp Flow	MV_L2	Alarm
02-05-23 23:23:01:784	192.168.0.51-0.11.3		HWError	For info see 'Errors and
02-05-23 20:00:07:762	500F1920	Washing Pulp Flow	MV_L2	Alarm
02-05-23 01:38:52:762	500F1920	Washing Pulp Flow	MV_L1	Alarm
02-05-23 00:18:26:995	500F1920	Washing Pulp Flow	AI_Err	Alarm
02-05-22 00:10:00:709	Mot01	Test Mot01	ControlV	Alarm
02-05-20 17:32:45:784	192.168.0.51-0.11.2		HWError	For info see 'Errors and

Figure 5-7 Alarm List

### 6.3.2 Alarm and Event Message

The following alarm texts are generated by the functional unit Groupstart1. The “Message Description” is “hard coded” in the function block and can not be modified.

The “Condition” text are stored in the Alarm and Event Translator aspect and can be NLS handled.

Object Name	Object Description	Condition	Message Description
<Name>	<Description>	SeqAlarm	Alarm
<Name>	<Description>	StepAlarm	Alarm

The following event texts are generated. The “Message Description” text are stored in the Alarm and Event Translator aspect and can be NLS handled.

\* ) Ext Start/Stop/Hold is a combination of all Start/Stop/Hold orders from Application.

SourceName	ObjectDescription	Condition	Message Description
<Name>	<Description>		SeqE1 Mode
<Name>	<Description>		SeqMan Mode
<Name>	<Description>		Local Mode
<Name>	<Description>		Ext Start On *)
<Name>	<Description>		Ext Start Off *)
<Name>	<Description>		Ext Stop On *)
<Name>	<Description>		Ext Stop Off *)
<Name>	<Description>		LHold On *)
<Name>	<Description>		LHold Off *)
<Name>	<Description>		IB1 On
<Name>	<Description>		IB1 Off
<Name>	<Description>		IB2 On

<Name>	<Description>		IB2 Off
<Name>	<Description>		IA1 On
<Name>	<Description>		IA1 Off
<Name>	<Description>		IA2 On
<Name>	<Description>		IA2 Off
<Name>	<Description>		Ext Reset On
<Name>	<Description>		Ext Reset Off
<Name>	<Description>		Alarm Acknowledge
<Name>	<Description>		Alarm Control Block

## 6.4 Faceplate tabs

### 6.4.1 Alarm and Event Blocking

By using the extended faceplate it is possible for the process engineer to block alarms.

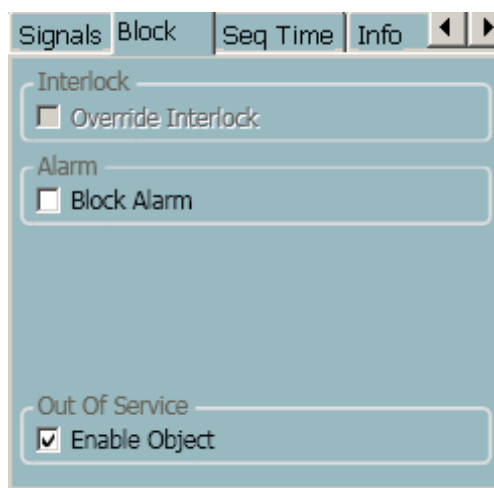


Figure 5-8 Extended Faceplate (Block)



Figure 5-9 Extended Faceplate (Signals)

### 6.4.2 Status

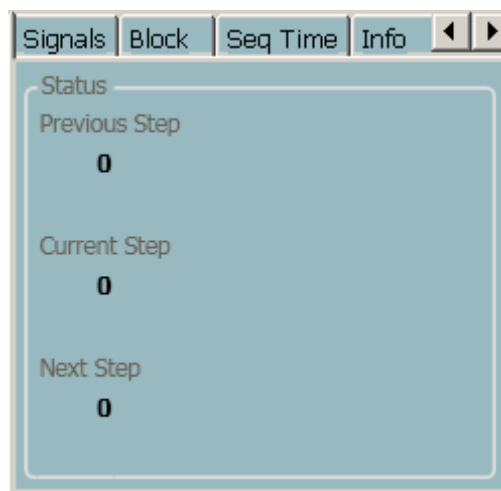


Figure 5-10 Extended Faceplate (Status)

### 6.4.3 Timers

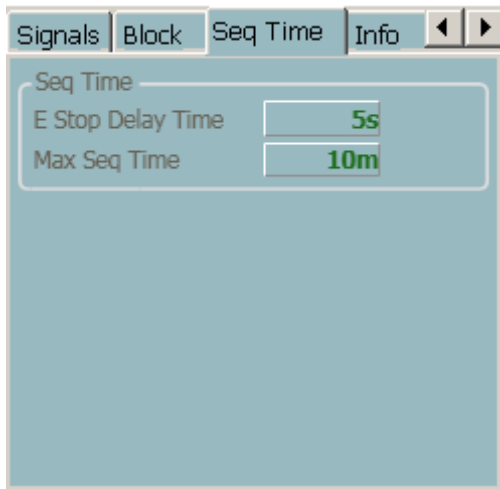


Figure 5-11 Extended Faceplate (Seq. Time)

### 6.4.4 Info



Figure 5-12 Extended Faceplate ( Info)

## 7 Step Name Texts

Each step of the Groupstart can be associated with an Object name or any descriptive text. These texts are, for performance reasons implemented with a “Property Translation” aspect in the Operate<sup>IT</sup> Operator Station. The current, previous and next step text is shown in the Faceplate and on the Object Display.

To enter customized texts for each step, select the “Step Names Property Translations” aspect in the “Control Structure”. Right click and select “Override”. From the Properties window select “CurrStep” and enter your text in the column “Text when true” next to each step number, see example below. Copy the just entered texts to the other conditions, (NextStep and PrevStep).

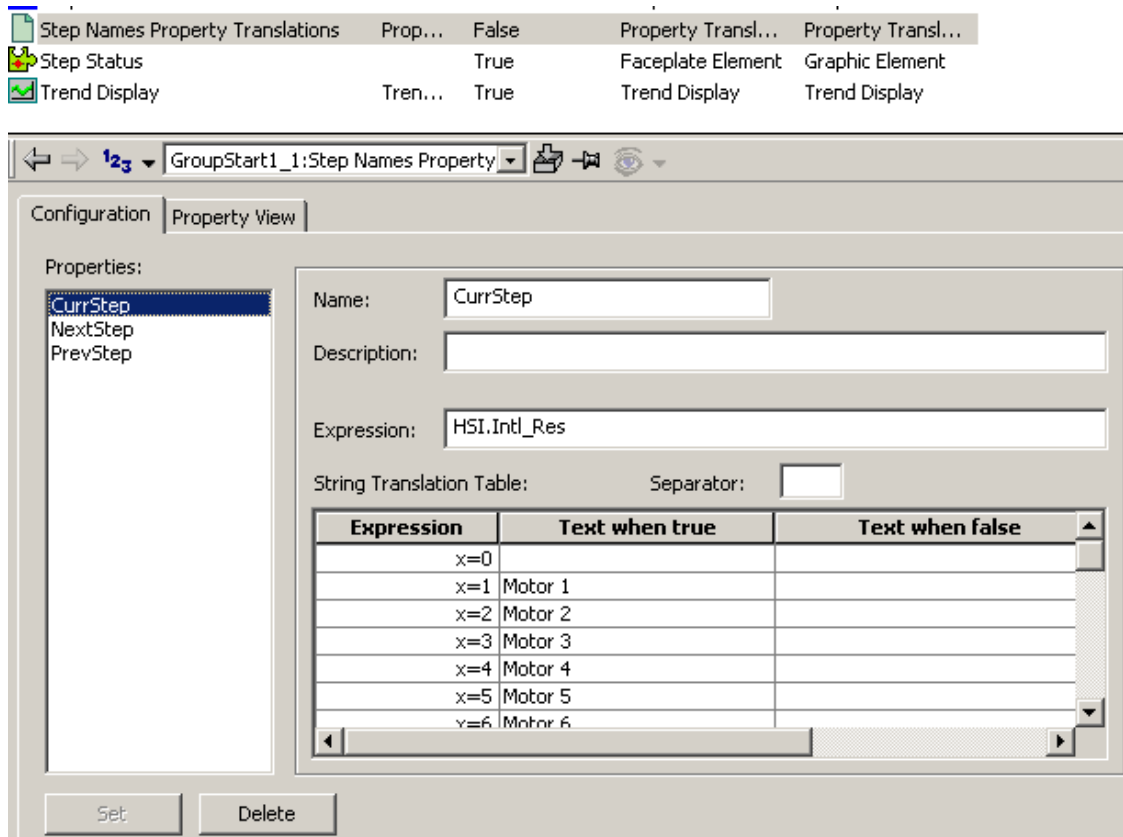


Figure 5-11 Step Names Entry Form.

REVISION

Rev.	Page (P) Chapt. (C)	Description	Date Dept./Init.
A	-	Release 2.0	03-05-23
B		Release 2.0/1, Texthandling modified	03-10-13
C	4, 5, 7	Initialization	04-04-15/FM
D	3	Rev 3.1/2	050319/MP
E	4,5	Event handling is added. Interaction Windows are updated	050401/BP
F	4,5	Faceplate, Rev 4.0/1	050902/MP
G	3	Rev 4.0/5	070510/BP
H		Rev 5.0-1 Interlock functionality is updated	081203/BP
I		Update Rev 5.1-0	101102/BP