Functional Description

MotFreq - Variable Speed Motor

Version 5.2-0

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

TRADEMARKS

Release: February 2012

Document number: 3BTG811792-3029

Table of Contents

Section 1 - Introduction

General

MotFreq is a functional unit for variable speed motor control in Control^{IT}, to be operated from 800xA System. A MotFreq normally performs a complete function independently.

MotFreq has the following functions and properties:

- Different control modes and speed modes, set by operator or by control logic
- Control circuits supervision
- Interlocks function
- Start & stop control
- Speed control
- Motor current supervision
- Alarm & Event detection and handling

Section 2 - Function Block & Data Types

Function Block

Name	Data Type	Attributes	Direction	FD Port	Initial value	Description
Name	string	coldretain	in	yes	'MotFreq'	Object name
Description	string	coldretain	in	yes	'Descr'	Object description
Enable	bool	coldretain	in	yes	true	Enable object
ME	bool	retain	in	yes	true	Enable control circuit alarm
M1	bool	retain	in	yes	true	Control voltage
M2	bool	retain	in	yes	true	Overload
M3	bool	retain	in	yes	true	Motor breaker
M4	bool	retain	in	yes	true	Emergency stop
M5	bool	retain	in	yes	true	Short Circuit
IC1	bool	retain	in	yes	true	Safety Interlock 1
IC2	bool	retain	in	yes	true	Safety Interlock 2
ICs	ICConn	by_ref	in	yes		Safety Interlocks
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
IBs	IBConn	by_ref	in	yes		Process Interlocks
IA1	bool	retain	in	yes	true	Sequence Interlock 1
IA2	bool	retain	in	yes	true	Sequence Interlock 2
IAs	IAConn	by_ref	in	yes		Sequence Interlocks
T1	time	coldretain	in	yes	2s	Start order pulse Time
T2	time	coldretain	in	yes	4s	Supervision Time
T3	time	coldretain	in	yes	5s	Run interlock delay Time
TWarn	time	coldretain	in	yes	0s	Start warning time
E1Start	bool	retain	in	yes		Start order in E1 mode
E2Start	bool	retain	in	yes		Start order in E2 mode
Stop	bool	retain	in	yes	true	Stop order in external mode
Ack1	bool	retain	in	yes		Main contactor acknowledge
MC	real	retain	in	yes		Motor current
JogEnbl	bool	retain	in	yes		Order mode to Jog
JogStart	bool	retain	in	yes		Start order in Jog mode
JogFunc	bool	retain	in	yes		Jog start hold function
LEnbl	bool	retain	in	yes		Order mode to Local
L1	bool	retain	in	yes		Start order in Local mode
LStop	bool	retain	in	yes	true	Local Stop order
SeqMan	bool	retain	in	yes		Order mode to Man
SeqE1	bool	retain	in	yes		Order mode to E1
SeqE2	bool	retain	in	yes		Order mode to E2
RunInt1	bool	retain	in	yes	true	Run Interlock 1
RunInt2	bool	retain	in	yes	true	Run Interlock 2
RunInt2F	bool	retain	in	yes		Function for Run Interlock 2
Track_A	bool	coldretain	in	yes	true	Track A Enable - Bumpless from Ext SP mode to Man SP mode
Track_B	bool	coldretain	in	yes	true	Track B Enable - Bumpless from Local SP mode to Man SP mode

Name	Data Type	Attributes	Direction	FD Port	Initial value	Description
MotTemp	bool	retain	in	yes	true	Motor Temperature
BearTemp	bool	retain	in	yes	true	Bearing Temperature
MV_LEnbl	bool	retain	in	yes		Order SP Mode to Local
SeqManSP	bool	retain	in	yes		Order SP mode to Man
SeqExtSP	bool	retain	in	yes		Order SP mode to External
MV	real	retain	in	yes		Speed Measured Value
Speed1	real	coldretain	in	yes	2	Max setpoint change rate in Man SP mode and Ext SP mode (unit/s)
ExtRef	real	retain	in	yes		External reference setpoint in External SP mode
AlcBlk	bool	retain	in	yes		Block alarm
AlarmAck	bool	retain	in	yes		Acknowledge alarm
MCAlarm	AlarmParCurr	by_ref	in	yes		MC alarm configuration
InPar	MotFreq_InPar	by_ref	in	yes		In Parameter
EventName	string	coldretain	in	yes	'||MotFreq_'	Event name
NoInt	bool	retain	out	yes		No Interlocks
NoICInt	bool	retain	out	yes		No Safety Interlocks (IC)
NoIBInt	bool	retain	out	yes		No Process Interlocks (IB)
NoIAInt	bool	retain	out	yes		No Sequence Interlocks (IA)
Trip	bool	retain	out	yes		Trip
Blk	bool	retain	out	yes		Standby
RFS	bool	retain	out	yes		Ready for start
Ready	bool	retain	out	yes		Motor Ready Indication in E1 or E2
SO1	bool	retain	out	yes		Start order
SAck	bool	retain	out	yes		Start order acknowledgement
StartWarn	bool	retain	out	yes		Start warning
Run	bool	retain	out	yes		Running
Sim	bool	retain	out	yes		Simulation mode
JogInd	bool	retain	out	yes		Jog mode
LocalInd	bool	retain	out	yes		Local mode
ManInd	bool	retain	out	yes		Man mode
E1Ind	bool	retain	out	yes		E1 mode
E2Ind	bool	retain	out	yes		E2 mode
SP_Local	bool	retain	out	yes		Local SP mode
SP_Man	bool	retain	out	yes		Man SP mode
SP_Ext	bool	retain	out	yes		External SP mode
SP_OUT	real	retain	out	yes		Speed setpoint
Curr	real	retain	out	yes		Actual current (%)
NoOfStart	dint	coldretain	out	yes		Number of start
RunTime	real	coldretain	out	yes		Total running time (in minutes)
LastRunTime	real	coldretain	out	yes		Last/current running time (in minutes)
LastReset	date_and_time	coldretain	out	yes		Last reset time
OutPar	MotFreq_OutPar	by_ref	out	yes		Out Parameter
Opr	MotFreq_Opr	by_ref	out	yes		Operator order

Data Types

MotFreq_InPar

MotFreq_InPar data type contains all configuration parameter available in MotFreq. Configuration can be entered by writing value to this structured data type and connecting it to input **InPar** of MotFreq function block.

Name	Data Type	Attributes	Initial value	ISP value	Description
Class	dint	coldretain	500		AE class
Severity	dint	coldretain	1000		AE severity
SpeedRange	RangeReal	coldretain			Speed Range
SpeedSPLimit	RangeLimit	coldretain			Speed SP limit
InitMode	dint	coldretain	5		Init mode (5 = Man ; 7 = E1 ; 8 = E2)
InitSPMode	dint	coldretain	2		Init SP mode (1= Local SP ; 2= Man SP ; 3= ExtSP)
JogBlk	bool	coldretain	false		Block operator order Jog mode
LocalBlk	bool	coldretain	false		Block operator order Local mode
ManBlk	bool	coldretain	false		Block operator order Man mode
E1Blk	bool	coldretain	false		Block operator order E1 mode
E2Blk	bool	coldretain	false		Block operator order E2 mode
LocalSPBlk	bool	coldretain	false		Block operator order Local SP mode
ManSPBlk	bool	coldretain	false		Block operator order Man SP mode
ExtSPBlk	bool	coldretain	false		Block operator order Ext SP mode
StartBlk	bool	coldretain	false		Block operator order Start command
StopBlk	bool	coldretain	false		Block operator order Stop command
AlcBlkEvBlk	bool	coldretain	true		Block event for AlcBlk
LEnblEvBlk	bool	coldretain	true		Block event for LEnbl
JogEnblEvBlk	bool	coldretain	true		Block event for JogEnbl
SeqManEvBlk	bool	coldretain	true		Block event for SeqMan
SeqE1EvBlk	bool	coldretain	true		Block event for SeqE1
SeqE2EvBlk	bool	coldretain	true		Block event for SeqE2
SeqManSPEvBlk	bool	coldretain	true		Block event for SeqManSP
SeqExtSPEvBlk	bool	coldretain	true		Block event for SeqExtSP
MVLEnblEvBlk	bool	coldretain	true		Block event for MV_LEnbl
ExtStartEvBlk	bool	coldretain	true		Block event for ExtStart
ExtStopEvBlk	bool	coldretain	true		Block event for ExtStop
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
IAs1	IAInParType1	coldretain			Configuration for IAs.IA1
IAs2	IAInParType1	coldretain			Configuration for IAs.IA2
IB1	IBInParType1	coldretain			Configuration for IB1
IB2	IBInParType1	coldretain			Configuration for IB2
IB3	IBInParType1	coldretain			Configuration for IB3
IB4	IBInParType1	coldretain			Configuration for IB4
IBs1	IBInParType1	coldretain			Configuration for IBs.IB1
IBs2	IBInParType1	coldretain			Configuration for IBs.IB2
IBs3	IBInParType1	coldretain			Configuration for IBs.IB3
IBs4	IBInParType1	coldretain			Configuration for IBs.IB4
IBs5	IBInParType1	coldretain			Configuration for IBs.IB5
IBs6	IBInParType1	coldretain			Configuration for IBs.IB6
IBs7	IBInParType1	coldretain			Configuration for IBs.IB7
IBs8	IBInParType1	coldretain			Configuration for IBs.IB8
IC1	ICInParType1	coldretain			Configuration for IC1
IC2	ICInParType1	coldretain			Configuration for IC2

Name	Data Type	Attributes	Initial value	ISP value	Description
ICs1	ICInParType1	coldretain			Configuration for ICs.IC1
ICs2	ICInParType1	coldretain			Configuration for ICs.IC2
CurrRange	RangeCurr	coldretain			Current range
ShowCurrent	dint	coldretain	0		Current presentation (0 = no current shown ; 1 = current shown in percent ; 2 = current shown in unit)
ResetSP	bool	coldretain	false		Reset SP when stopping motor in Man Mode
AEConfigBTemp	dint	coldretain	0		AE configuration for Bearing Temperature
AEConfigM1	dint	coldretain	1		AE configuration for M1
AEConfigM2	dint	coldretain	1		AE configuration for M2
AEConfigM3	dint	coldretain	1		AE configuration for M3
AEConfigM4	dint	coldretain	1		AE configuration for M4
AEConfigM5	dint	coldretain	1		AE configuration for M5
AEConfigMCErr	dint	coldretain	1		AE configuration for Main Contactor Error
AEConfigMTemp	dint	coldretain	0		AE configuration for Motor Temperature
AEConfigRInt1	dint	coldretain	0		AE configuration for Run Interlock 1
AEConfigRInt2	dint	coldretain	0		AE configuration for Run Interlock 2
EnNoOfStartCounter	bool	coldretain	false		Enable number of start counter
EnRunTimeCounter	bool	coldretain	false		Enable running time counter

MotFreq_OutPar

MotFreq_OutPar data type contains all information on the object which not available on the output parameter of the function block.

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
SPMode	dint	retain			Active SP mode
NormalMode	bool	retain			Normal mode (Active mode = Init mode)
NormalSPMode	bool	retain			Normal SP mode (Active SP mode = Init SP mode)
IA1Ind	bool	retain			IA1 interlocked
IA2Ind	bool	retain			IA2 interlocked
IB1Ind	bool	retain			IB1 interlocked
IB2Ind	bool	retain			IB2 interlocked
IB3Ind	bool	retain			IB3 interlocked
IB4Ind	bool	retain			IB4 interlocked
IC1Ind	bool	retain			IC1 interlocked
IC2Ind	bool	retain			IC2 interlocked
IAs1Ind	bool	retain			IAs1 interlocked
IAs2Ind	bool	retain			IAs2 interlocked
IBs1Ind	bool	retain			IBs1 interlocked
IBs2Ind	bool	retain			IBs2 interlocked
IBs3Ind	bool	retain			IBs3 interlocked
IBs4Ind	bool	retain			IBs4 interlocked
IBs5Ind	bool	retain			IBs5 interlocked
IBs6Ind	bool	retain			IBs6 interlocked
IBs7Ind	bool	retain			IBs7 interlocked
IBs8Ind	bool	retain			IBs8 interlocked

Name	Data Type	Attributes	Initial value	ISP value	Description
ICs1Ind	bool	retain			ICs1 interlocked
ICs2Ind	bool	retain			ICs2 interlocked
BTemp	AlarmInd	retain			Alarm Indication for BearingTemp
MTemp	AlarmInd	retain			Alarm Indication for MotorTemp
HCurr	AlarmInd	retain			Alarm Indication for HCurr
M1	AlarmInd	retain			Alarm Indication for M1
M2	AlarmInd	retain			Alarm Indication for M2
M3	AlarmInd	retain			Alarm Indication for M3
M4	AlarmInd	retain			Alarm Indication for M4
M5	AlarmInd	retain			Alarm Indication for M5
MCErr	AlarmInd	retain			Alarm Indication for MCErr
RInt1	AlarmInd	retain			Alarm Indication for RInt1
RInt2	AlarmInd	retain			Alarm Indication for RInt2
LastStop	dint	retain			Reason for last stop
StartTime	time	retain			Actual starting time
StopTime	time	retain			Actual stopping time
Ack	bool	retain			Running feedback indication
MV	real	retain			Speed indication
MC	real	retain			Motor current indication

MotFreq_Opr

MotFreq_Opr data type contains all operator order.

Name	Data Type	Attributes	Initial value	ISP value	Description
BlockAlarm	bool	retain			Operator block alarms
Jog	bool	retain			Operator order Jog mode
Local	bool	retain			Operator order Local mode
Man	bool	retain			Operator order Manual mode
E1	bool	retain			Operator order E1 mode
E2	bool	retain			Operator order E2 mode
Start	bool	retain			Operator order Start command
Stop	bool	retain			Operator order Stop command
LocalSP	bool	retain			Operator order Local SP mode
ManSP	bool	retain			Operator order Man SP mode
ExtSP	bool	retain			Operator order Ext SP mode
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
IBs1Override	bool	retain			Operator override IBs.IB1 interlock
IBs2Override	bool	retain			Operator override IBs.IB2 interlock
IBs3Override	bool	retain			Operator override IBs.IB3 interlock
IBs4Override	bool	retain			Operator override IBs.IB4 interlock
IBs5Override	bool	retain			Operator override IBs.IB5 interlock
IBs6Override	bool	retain			Operator override IBs.IB6 interlock
IBs7Override	bool	retain			Operator override IBs.IB7 interlock
IBs8Override	bool	retain			Operator override IBs.IB8 interlock
IA1Override	bool	retain			Operator override IA1 interlock
IA2Override	bool	retain			Operator override IA2 interlock
IAs1Override	bool	retain			Operator override IAs.IA1 interlock
IAs2Override	bool	retain			Operator override IAs.IA2 interlock
SP	real	coldretain	0.0		Operator enter manual SP

Name	Data Type	Attributes	Initial value	ISP value	Description
ResetCounter	bool	retain			Order reset counter
EnableSim	bool	retain			Enable simulated running feedback

Permission

All variables in **MotFreq_InPar** data type are set with *Configure* permission in 800xA system. All variables in **MotFreq_Opr** data type are set with *Operate* permission.

The variable permission can be configured according to the plant requirement. This can be done by adding **Property Attribute Override** aspect for the object in Control Structure or for the object type in Object Type Structure.

Control Modes

There are five different control modes are available in priority order: Jog, Local, Man, E1 and E2. Active mode is indicated on the faceplate and object display. These modes can also be blocked individually from operator access. When motor is in one particular mode, control from other location is blocked.

Jog Mode

Jog mode is suitable for testing motor in the field. The motor is controlled from start / stop station adjacent to the motor. Jog mode can be activated from the **Jog** button on the faceplate or by setting the input **JogEnbl** of the function block.

When motor is in Jog mode, control from other location is blocked. The interlocks supervision is limited. Only the safety interlocks **IC** and faults in **M1** - **M5** will be monitored. When the motor is leaving Jog mode, a stop order is generated automatically. It will then resume the last control mode from operator station.

The motor is started and stopped through input **JogStart** and **JogFunc**. **JogStart** is for receiving start order while **JogFunc** is for receiving the stop or hold order from field. The function of these inputs is set as follow:

JogStart	**JogFunc**	
0	0	Stop order
0	1	No change
1	0	Start and stop order on **JogStart**
1	1	Start order on **JogStart** and stop order on **JogFunc**

Local Mode

The motor is controlled from a local desk or panel. As long as **JogEnbl** is not active, Local mode can be activated from the **Local** button on the faceplate or by setting the input **LEnbl** of the function block.

When motor is in Local mode, control from other location is blocked. When the motor is leaving Local mode, it will resume the last control mode from operator station.

The motor is started with a rising edge pulse to **L1** while **LStop** must be in *True* condition. Motor can be stopped with a falling edge pulse to **LStop**.

Manual Mode

In this mode the operator directly start and stop motor from the faceplate button **Start** and **Stop**. As long as **JogEnbl** and **LEnbl** is not active, Manual mode can be activated from the **Man** button on the faceplate or with a rising edge pulse to the input **SeqMan** of the function block.

E1 and E2 Mode

The motor is controlled from external signals, i.e.: a process signal. It is normally used for automatic start / stop agitators, belt conveyors, etc. As long as **JogEnbl** and **LEnbl** is not active, E1 or E2 mode can be activated from the **E1** or **E2** button on the faceplate or with a rising edge pulse to the input **SeqE1** or **SeqE2** of the function block.

The motor is started with a rising edge pulse to **E1Start** for E1 mode or **E2Start** for E2 mode while **Stop** must be in *True* condition. Motor can be stopped with a falling edge pulse to **Stop** for both modes.

Operator can also stop the motor by pressing the faceplate button **Stop**. Control mode will then change to Manual to prevent motor from being started again automatically.

The table below describes the commands at the different control modes of MotFreq.

Order	Jog	Local	Manual	E1	E2
Input *JogStart*	Start / Stop	-	-	-	-
Input *L1*	-	Start	-	-	-
Input *LStop*	-	Stop	-	-	-
Operator start	-	-	Start	-	-
Operator stop	-	-	Stop	Stop	Stop
Input *E1Start*	-	-	-	Start	-
Input *E2Start*	-	-	-	-	Start
Input *Stop*	-	-	-	Stop	Stop

Setpoint Modes

There are three different setpoint modes are available in priority order: Local, Man and External. By selecting setpoint mode, operator decides from which location the setpoint is controlled. These modes can also be blocked individually from operator access.

There is no correlation between control mode and setpoint mode, i.e.: when motor is in E1 mode, it doesn't necessarily mean setpoint mode is in External mode too. If this functionality is required, then logic is needed to be programmed externally to the function block. All control modes is not affected by changes in setpoint mode and vice versa.

Local Setpoint Mode

Local setpoint mode is suitable for entering the setpoint from the frequency converter. The speed setpoint **SP_OUT** is ignored by the frequency converter as the frequency converter itself is providing the setpoint. The value of speed measurement **MV** is transferred to the output parameter **SP_OUT**.

Local setpoint mode can be activated from the **L SP** button on the faceplate or by setting the input **MV_LEnbl** of the function block. When motor is in Local setpoint mode, setpoint from other location is blocked. It will resume the last setpoint mode from operator station when **MV_LEnbl** is released.

Manual Setpoint Mode

In this mode the operator directly enter the speed setpoint **SP_OUT** from the faceplate's dialog entry window. As long as **MV_LEnbl** is not active, Manual setpoint mode can be activated from the

M SP button on the faceplate or with a rising edge pulse to the input **SeqManSP** of the function block.

External Setpoint Mode

In this mode, setpoint is provided from external signal such as from PID controller output. The value from input parameter **ExtRef** will then be available on the output parameter **SP_OUT**. As long as **MV_LEnbl** is not active, External setpoint mode can be activated from the **E SP** button on the faceplate or with a rising edge pulse to the input **SeqExtSP** of the function block.

Setpoint Ramping

The input parameter **Speed1** is used to define the change rate of the setpoint in Manual and External setpoint mode. In Local setpoint mode, the setpoint is indicated directly without ramping.

Speed1 can be obtained by dividing the setpoint range with the time required to reach the maximum speed from zero.

Setpoint Tracking

The setpoint tracking is controlled by input parameters **Track_A** and **Track_B**.

If **Track_A** is *True*, the external setpoint will be retained when changing setpoint mode from External to Manual in order to provide bumpless transfer. If **Track_B** is *True*, the local setpoint will be retained when changing setpoint mode from Local to Manual.

Interlocks

The motor control can be interlocked by signals from the process or control logic. Interlocks or faults are active when the value is *False*. The interlocks are divided into four groups with the following designations and functions. All interlocks are blocked when motor in Jog mode except for Safety Interlocks.

Safety Interlocks

Safety interlocks are used to interlock the motor for safety purposes. There are 4 safety interlocks available: **IC1**, **IC2**, **ICs.IC1** and **ICs.IC2**. These interlocks cannot be blocked.

Process Interlocks

Process interlocks are used for normal process interlocks. There are 12 process interlocks available: **IB1** - **IB4** and **IBs.IB1** - **IBs.IB8**. All can be configured to accept blocking by the operator. They can also be configured as start interlocks. A start interlock will prevent the motor from starting, but does not stop a running motor.

Sequence Interlocks

Sequence interlocks are used if the motor is to be interlocked against other objects such as a pump motor or a group start or other conditions in an automatic sequence. There are 4 sequence interlocks available: **IA1**, **IA2**, **IAs.IA1** and **IAs.IA2**. These interlocks are configurable as blocked or not when motor is not in E1 or E2 mode.

If **IA blocked when not Ext** is equal to 1

Motor interlocked if **IA1** or **IAs.IA1** is 0 and the motor is in E1 mode

Motor interlocked if **IA2** or **IAs.IA2** is 0 and the motor is in E2 mode

If **IA blocked when not Ext** is equal to 0

Motor interlocked if **IA1**, **IA2**, **IAs.IA1** or **IAs.IA2** is 0 and the motor is in Local, Man, E1 or E2 mode.

Run Interlocks

Run interlocks are used to obtain alarm or fault while motor is running. There are 2 run interlocks available: **RunInt1** and **RunInt2**. Run interlocks are blocked during the time *T3* after starting. Run interlocks have Alarm & Event configuration to define the alarm treatment.

RunInt1 has a motor tripping function whereas the **RunInt2** function can be configured with input **RunInt2F**. When input **RunInt2F** is 1, the motor is tripped and when **RunInt2F** is 0, only an alarm is given.

Ready for Start

Ready for Start means all condition is cleared for motor to start. There is no Safety and Process interlocks, Sequence interlocks are not included. No faults / unacknowledged faults alarms exists. Motor is not in Jog mode.

Ready for Start is indicated by the output parameter **RFS** in the function block.

Ready

The output parameter **Ready** has same functionality for as the RFS. **Ready** will be set to *True* when the motor is Ready for Start and in E1 or E2 mode. This parameter is normally used to check the motor's status when motor is part of a sequence or group start.

Start and Stop

Start and stop commands may come from faceplate button or input parameter, depending on the selected control mode.

When a start command is issued, it will check if start warning should be given. Start warning **StartWarn** is activated if start warning time **TWarn** is greater than zero. After **TWarn** time elapses, the start order **SO1** is sent to the motor.

The motor's main contactor acknowledges the start order by activating the running feedback **Ack1**. The main contactor must acknowledge within **T2** time.

If main contactor acknowledges the start order **SO1** within the time **T2**, MotFreq will keep **SO1** high. Otherwise, the start attempt is deemed abortive and an alarm about the contactor failure is issued. The contactor failure alarm indication disappears when the operator acknowledges the alarm. A new attempt to start the motor then can be made.

MotFreq also issues an alarm about main contactor failure when the contactor acknowledges start orders falsely, i.e. when it sets the inputs **Ack1** high without any start order being issued or after stop order is given. Alarm will be generated after **T2** time.

Control Circuit Faults

Fault supervision is performed in the priority order **M1**, **M2**, **M5**, **M3** and **M4**. This means that if input **M1** is in fault condition, then inputs **M2** - **M5** are not regarded. When a number of faults occur in series, only the first fault will be reported in the alarm. Value at inputs **M1** - **M5** is 1 when there are no faults. Faults are indicated on the output **Trip**.

M1 - **M5** have Alarm & Event configuration to define the alarm treatment. The input **ME** can be used to block the alarm from being generated. No alarm will be issued if motor is not running.

Motor Alarms

MotFreq has two additional alarm supervision is which can be configured. The input parameter **MotTemp** can be used to indicate that motor has overheated and the input parameter **BearTemp** can be used for indication of overheated motor bearing. Value at these inputs is 1 when there are no alarms. These alarms will not trip the motor when activated.

Motor Current Supervision

The input **MC** is connected to the process signal for motor current measurement. The **MC** is normally measured in unit such as *Amp*.

The output parameter **Curr** presents the measured current in percentage. It is obtained by comparing the **MC** value against **Rated Current**. **Rated Current** is a value that corresponds to 100 % load.

Current will be supervised only after motor running for more than **Start Delay** time. High current is indicated when current is greater than the High Current Limit.

Simulation

The function block support internal simulation. Simulation can be activated by enabling the simulation button in Interaction Window or by setting the **Opr.EnableSim** to *True* using the bulk tool. The simulation will be reset when cold download is performed.

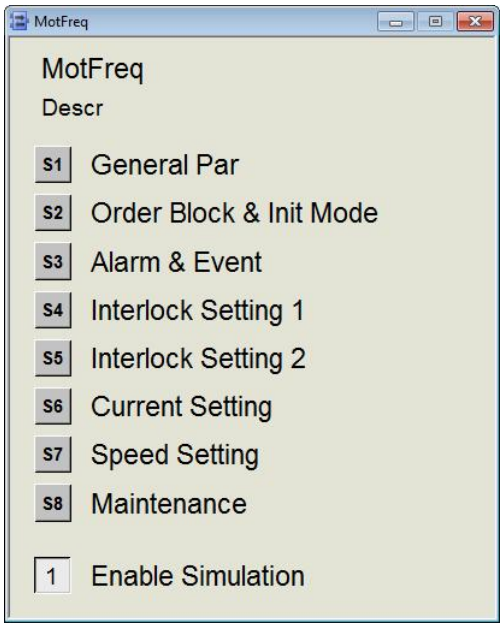
Object in simulation mode will be indicated with a frame on the graphic elements. The default color is flashing yellow as defined by *ppSimulation* in **Pulp & Paper Library Colors** aspect. It also indicates **Sim** next to the mode indication in faceplate.

Running feedback will be activated 2 seconds after start order **SO1** is *True*. The current will be simulated at 50%. Speed feedback will be filtered by 2s from the speed setpoint **SP_OUT**.

Interaction Window

Interaction Window is used to configure the object's parameter in the Control Builder. It is only accessible when Control Builder is in *Online* mode by right clicking the function block.

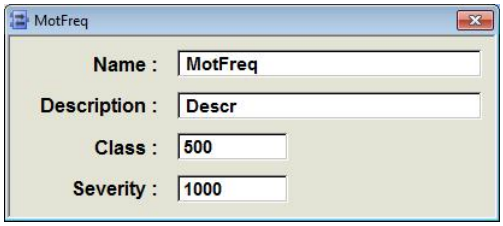
The main Interaction Window shows the name and description of the function block. The buttons are link to open the sub-window.



1. General Par

Name and **Description** can be entered from interaction window or from the function block parameters directly. After performing *Name Uploader*, these values will be used as Object Name and Description and shown in faceplate.

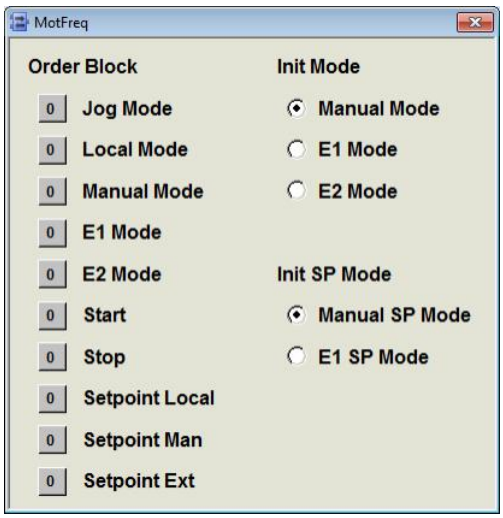
Class defines the process section or area in which alarms are grouped. By utilizing **Class**, the alarms can be filtered. Valid values are 1 - 9999. **Severity** defines the alarm priority for general alarms. Valid values are 1 - 1000 where 1000 is the highest priority.



2. Order Block & Init Mode

Order Block is used to prevent operator from giving an order of certain activity. The faceplate button will be dimmed if order block is activated (set to 1).

Init Mode defines the initial mode of the object when cold download is performed. By default, the init mode for MotFreq is Manual mode and init SP mode is Manual SP mode.



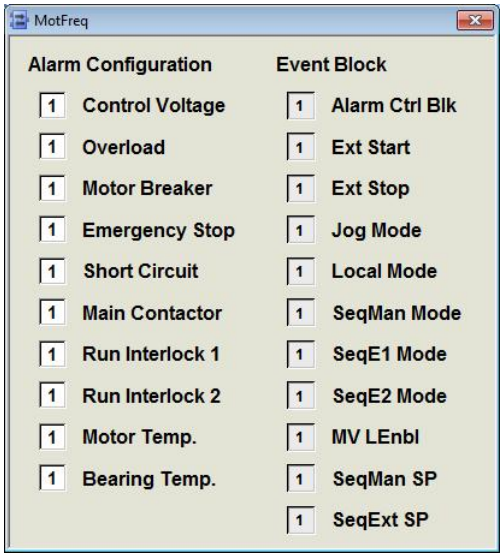
3. Alarm & Event

Alarm Configuration is used to configure how to present status changes of certain conditions. Alarm Configuration can be set in 3 different ways:

- 0 = No alarm or event will be generated
- 1 = Alarm and event will be generated
- 2 = Only event will be generated

Event Block is used to block the event message generated due to changes in value from logic connection. To enable event for selected message, press the button to set the value to 0.

The text message for each alarm / event is stored in the **AE Translator** aspect. List of events for all operator activity is managed in 800xA system with Audit Trail functionality.



4. Interlock Setting 1 & 2

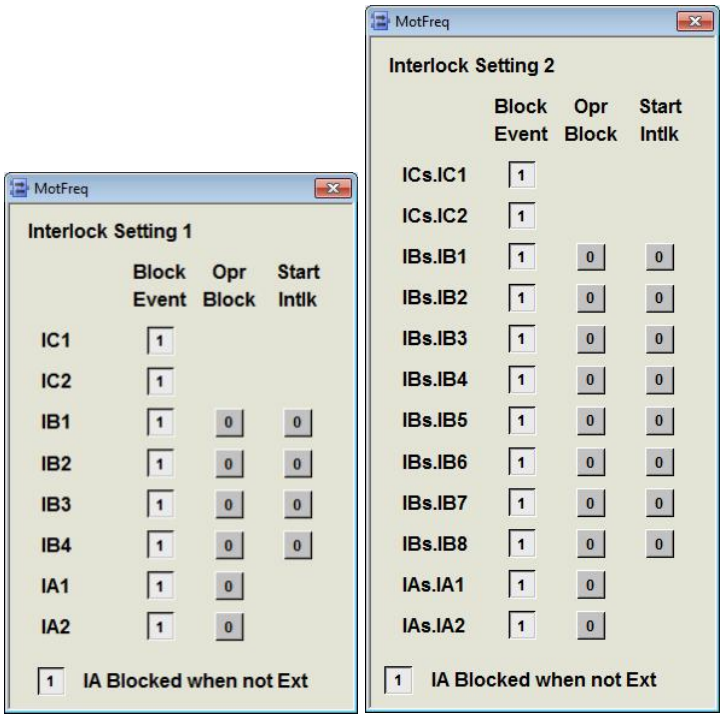
Interlock Setting is split into two. Interlock Setting 1 is for standard interlock **IC1** - **IC2**, **IB1** - **IB4** and **IA1** - **IA2**. Interlock Setting 2 is for extended interlock **ICs.IC1** - **ICs.IC2**, **IBs.IB1** - **IBs.IB8** and **IA2.IA1** - **IAs.IA2**.

When **Block Event** is disabled (set to 0), changes in value of the interlock will generate an event message. The text message for each event is stored in the **AE Translator** aspect.

Enabling the **Opr Block** (set to 1) will give operator the possibility to override the interlock(s) from faceplate.

When an interlock is configured as **Start Interlock**, it will only be monitored during the starting period. Once the motor is started, the interlock will no longer be observed.

By configuring **IA Blocked when not Ext** to 1, IA interlocks will not be monitored when the object is not in E1 or E2 mode. By setting the value to 0, IA interlocks will be observed in all modes.



5. Current Setting

Current Presentation determines how the current is presented in the faceplate. Three options available are no current presentation, current to be presented in percentage (%) and current to be presented in unit as defined in **Current Range**.

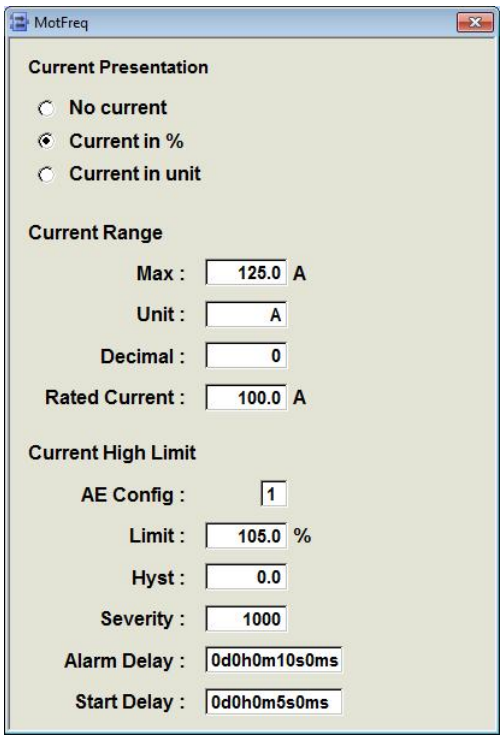
Current Range covers maximum range, unit, decimal and rated current. Max refers to the maximum range of the current I/O signal. **Rated Current** refers to the current value equals to 100%.

Only one current high limit can be configured. AE Config can be set in 3 different ways:

- 0 = No Alarm or Event will be generated
- 1 = Alarm and Event will be generated
- 2 = Only Event will be generated

If AE Config is set to 1, alarm will be generated if the limit stays active longer than alarm delay time. Otherwise, alarm will not be generated. Current will be supervised only after motor running for more than **Start Delay** time.

The alarm limit entered can be considered as in unit or in percentage depending on the current presentation selected. If the current presentation is to show current in %, then the alarm limit is assumed in %. If the current presentation is to show current in unit, then the alarm limit is assumed in unit as well.

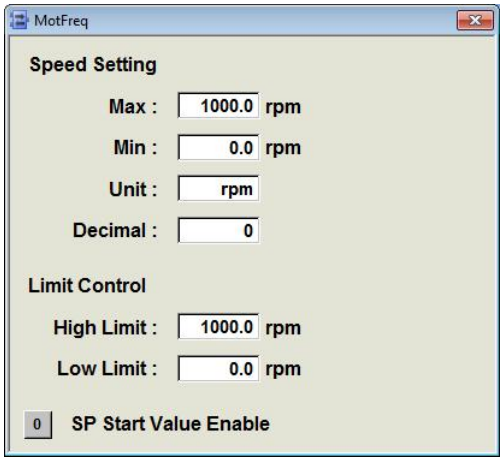


6. Speed Setting

Speed Setting covers minimum and maximum range, unit and decimal number for Speed. Motfreq can be used as two direction variable speeds motor by setting the Min range with the negative value of Max range. When speed setpoint is negative, it means running in reverse direction.

Limit Control defines the High and Low limit of Setpoint operator can enter from faceplate.

Enabling ***SP Start Value Enable*** will reset Man SP to Low Limit when motor is stopped.

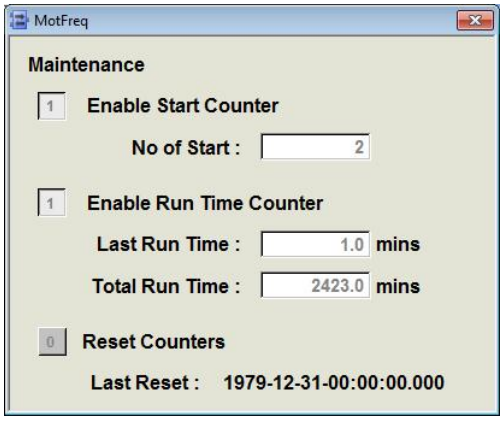


7. Maintenance

Maintenance covers the number of start and running time.

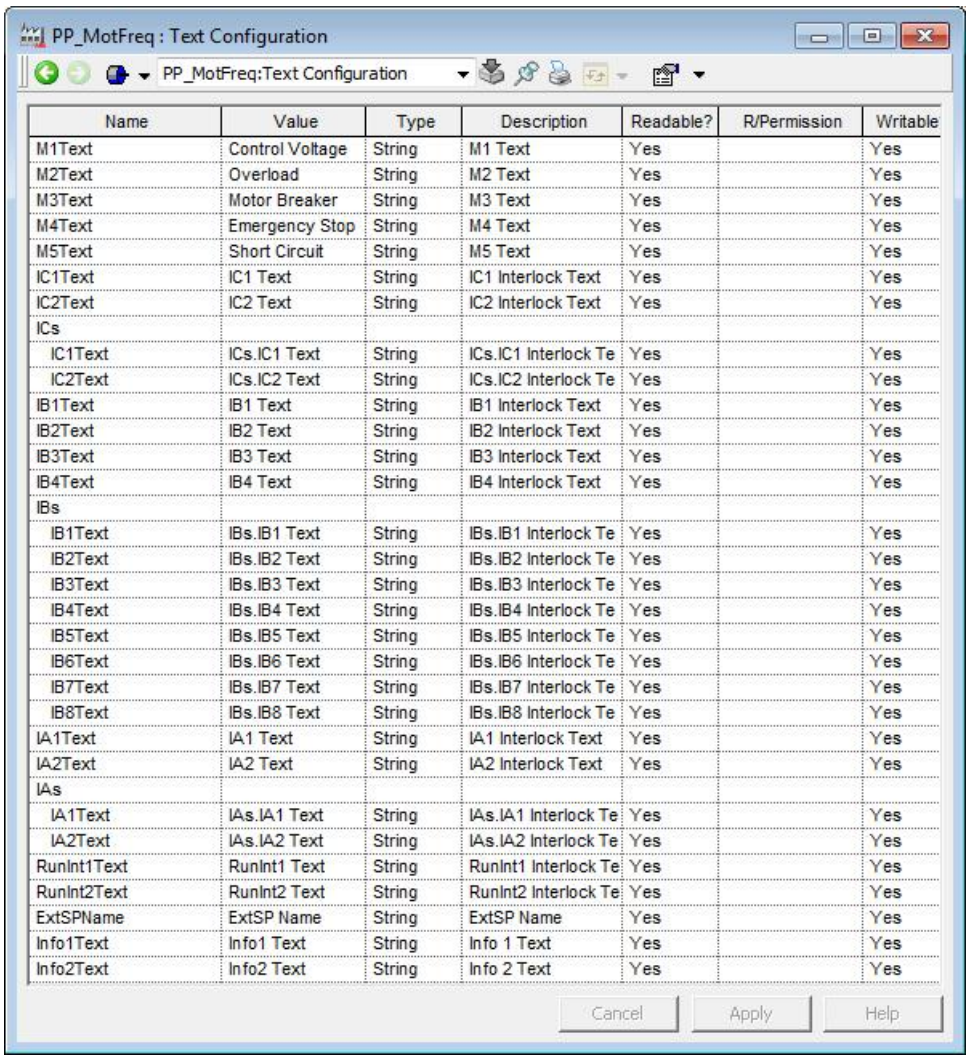
When **Enable Start Counter** is activated, it will count how many times the motor has been successfully started. Two types of run time will be counted when activating **Enable Run Time Counter**. **Last Run Time** indicates the last run time if motor is not running or current running time if motor is still running. **Total Run Time** will count how long the motor has been running since the last reset. Both run time is incremented on per minute period.

These values can be reset to zero by pressing the **Reset Counters** button. Reset can only be done when motor is not running. **Last Reset** indicates the date and time when the last reset is performed. If Reset has not been performed before, the **Last Reset** will show *1979-12-31-00:00:00.000*.



Text Configuration

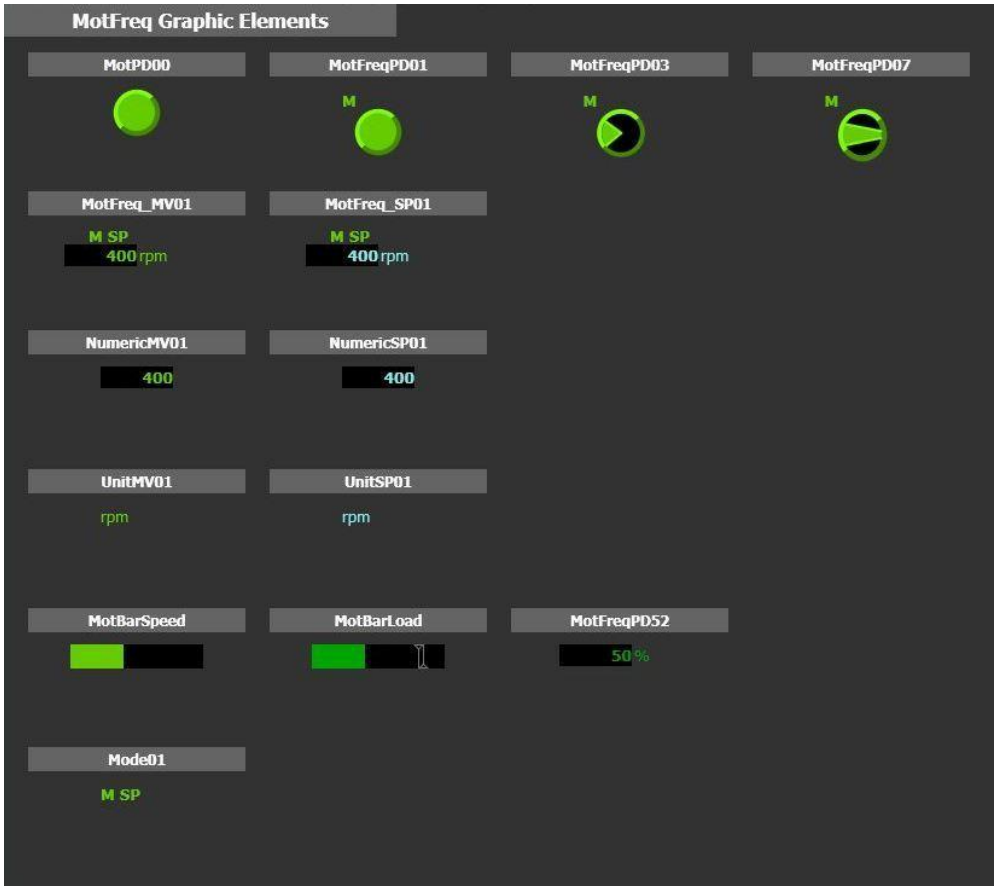
Text Configuration aspect is used to define the text properties of the object, such as ExtSPName, control circuit faults, interlocks and info texts. It is accessible in the Control Structure. The length of the each text should not exceed more than 50 characters, except for ExtSPName which is limited to 10 characters only.



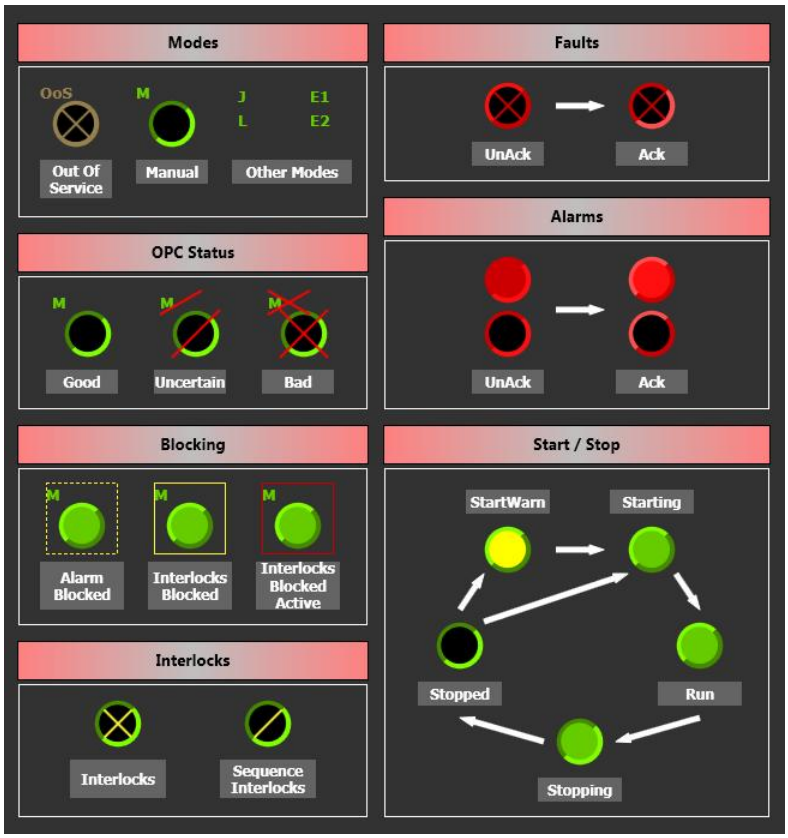
Name	Value	Type	Description	Readable?	R/Permission	Writable
M1Text	Control Voltage	String	M1 Text	Yes		Yes
M2Text	Overload	String	M2 Text	Yes		Yes
M3Text	Motor Breaker	String	M3 Text	Yes		Yes
M4Text	Emergency Stop	String	M4 Text	Yes		Yes
M5Text	Short Circuit	String	M5 Text	Yes		Yes
IC1Text	IC1 Text	String	IC1 Interlock Text	Yes		Yes
IC2Text	IC2 Text	String	IC2 Interlock Text	Yes		Yes
ICs						
IC1Text	ICs.IC1 Text	String	ICs.IC1 Interlock Te	Yes		Yes
IC2Text	ICs.IC2 Text	String	ICs.IC2 Interlock Te	Yes		Yes
IB1Text	IB1 Text	String	IB1 Interlock Text	Yes		Yes
IB2Text	IB2 Text	String	IB2 Interlock Text	Yes		Yes
IB3Text	IB3 Text	String	IB3 Interlock Text	Yes		Yes
IB4Text	IB4 Text	String	IB4 Interlock Text	Yes		Yes
IBs						
IB1Text	IBs.IB1 Text	String	IBs.IB1 Interlock Te	Yes		Yes
IB2Text	IBs.IB2 Text	String	IBs.IB2 Interlock Te	Yes		Yes
IB3Text	IBs.IB3 Text	String	IBs.IB3 Interlock Te	Yes		Yes
IB4Text	IBs.IB4 Text	String	IBs.IB4 Interlock Te	Yes		Yes
IB5Text	IBs.IB5 Text	String	IBs.IB5 Interlock Te	Yes		Yes
IB6Text	IBs.IB6 Text	String	IBs.IB6 Interlock Te	Yes		Yes
IB7Text	IBs.IB7 Text	String	IBs.IB7 Interlock Te	Yes		Yes
IB8Text	IBs.IB8 Text	String	IBs.IB8 Interlock Te	Yes		Yes
IA1Text	IA1 Text	String	IA1 Interlock Text	Yes		Yes
IA2Text	IA2 Text	String	IA2 Interlock Text	Yes		Yes
IAs						
IA1Text	IAs.IA1 Text	String	IAs.IA1 Interlock Te	Yes		Yes
IA2Text	IAs.IA2 Text	String	IAs.IA2 Interlock Te	Yes		Yes
RunInt1Text	RunInt1 Text	String	RunInt1 Interlock Te	Yes		Yes
RunInt2Text	RunInt2 Text	String	RunInt2 Interlock Te	Yes		Yes
ExtSPName	ExtSP Name	String	ExtSP Name	Yes		Yes
Info1Text	Info1 Text	String	Info 1 Text	Yes		Yes
Info2Text	Info2 Text	String	Info 2 Text	Yes		Yes

Process Display

MotFreq has the following graphic elements which can be inserted into graphic displays.



The following display shows the presentation in graphic elements under different states.



Faceplate

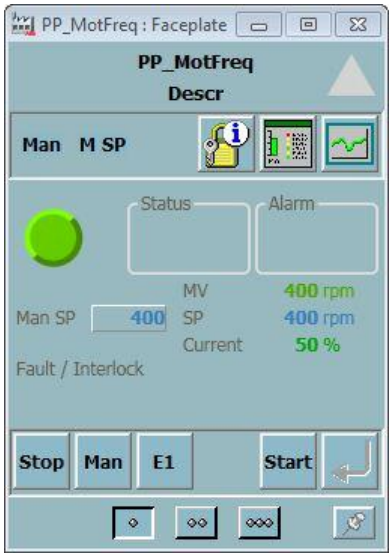
Aspect Link

Icon	Aspect View
	Operator Note
	Interlocks Display
	Object Display
	Object Trend Display

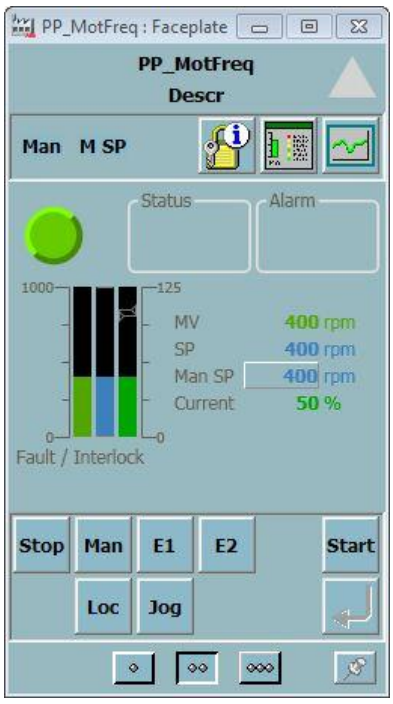
Reduced Faceplate and Faceplate

Reduced Faceplate and **Faceplate** give basic information about the object. Size and contents have been optimized to cover most of the normal process operator action. Process values presented in bargraph are available in **Faceplate**.

Reduce Faceplate is the default view.



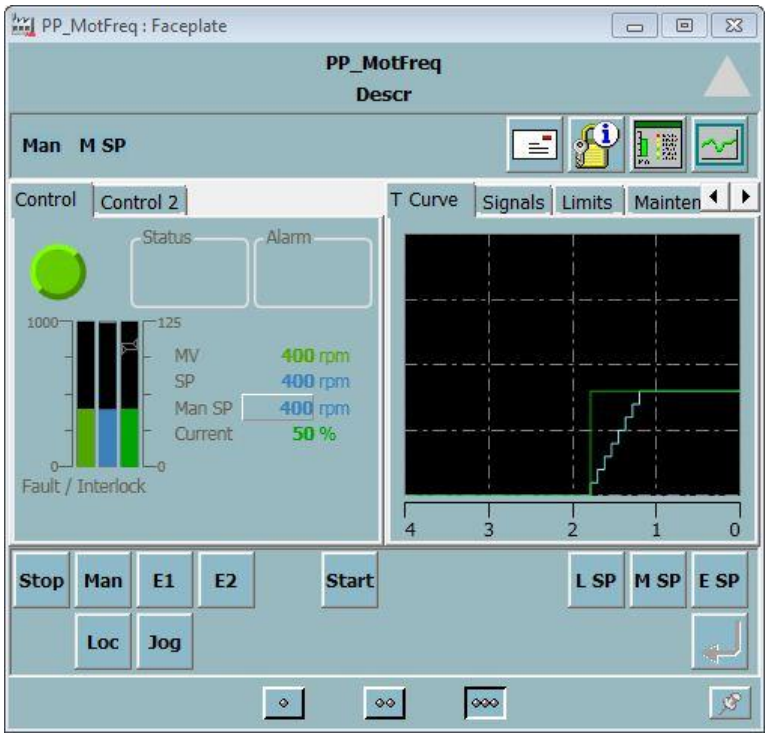
Reduced Faceplate



Faceplate

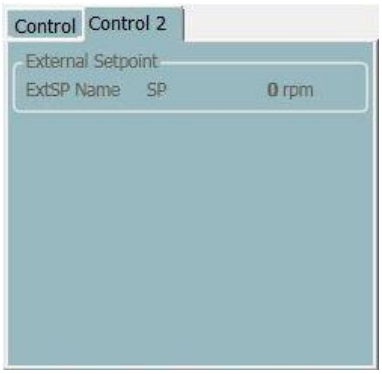
Extended Faceplate

The extended faceplate gives more detailed information about the object. Additional information about the object is available in tabs.



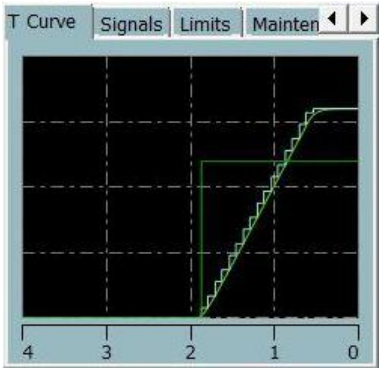
1. Control 2

This tab show the remote speed SP which will be used if the speed control is changed to E SP mode.



2. T Curve

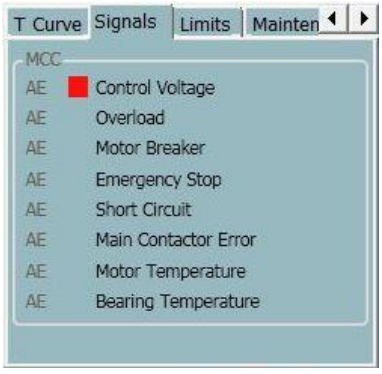
Trim Curve is used to present value in the last 4 minutes period. It contains 3 traces: **MV** (for Speed MV), **SP_OUT** (for Speed SP) and **Curr** (for Current in percentage) or **MC** (for Current in unit) depending on current presentation.



3. Signals

Fault signal can be configured as no alarm & event (no indication), alarm & event (indicated with **AE**) or event only (indicated with **E**). If alarm is blocked, then it will be indicated with **B**.

When a fault is active, a red square indication is shown in front of the fault signal. It will be indicated with the blinking red square indication as long as not acknowledged.



4. Limits

Current alarm limit, hysteresis and alarm delay can be defined in this tab. Alarm limit will be visible if configured as alarm & event (indicated with **AE**) or event only (indicated with **E**). If alarm is blocked, then it will be indicated with **B**.

Current will be supervised only after motor running for more than **Start Delay** time. When alarm limit is active, a red square indication is shown in front of the limit. Alarm will be generated if

the limit stays active for more than the alarm delay time. It will be indicated with the blinking red square indication as long as not acknowledged.

User with *Tune* permission can change the alarm limit, while hysteresis, alarm and start delay can be adjusted by user with *Configure* permission. The alarm limit entered can be considered as in unit or in percentage depending on the current presentation selected. If the current presentation is to show current in %, then the alarm limit is assumed in %. If the current presentation is to show current in unit, then the alarm limit is assumed in unit as well.

Rated Current refers to the current value equal to 100%.

5. Maintenance

Reason for Last Stop shows why the motor is stopped. It can be due to stop command from operator or logic, interlocks or faults.

When activated, counters value will be shown. Otherwise, nothing will be visible. **No Of Start** shows how many times the motor has been successfully started. If motor is running, it shows **Current Run Time**. If motor is stopped, it shows **Last Run Time**. **Total Run Time** shows how long the motor has been running since the last reset.

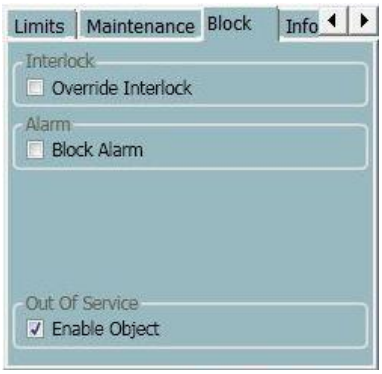
These values can be reset to zero by pressing the **Reset Counter** button. The button is enabled when motor is not running. **Last Reset** indicates the date and time when the last reset is performed. If Reset has not been performed before, the **Last Reset** will not show any value.

6. Block

Checkbox **Override Interlock** will override all blockable interlocks. It is enabled when at least one interlock is configured blockable.

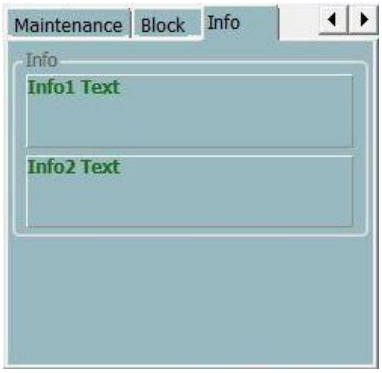
Operator can block the all alarms by ticking on the checkbox **Block Alarm**.

Un-ticking checkbox **Enable Object** will bring the object to **Out Of Service** mode.



7. Info

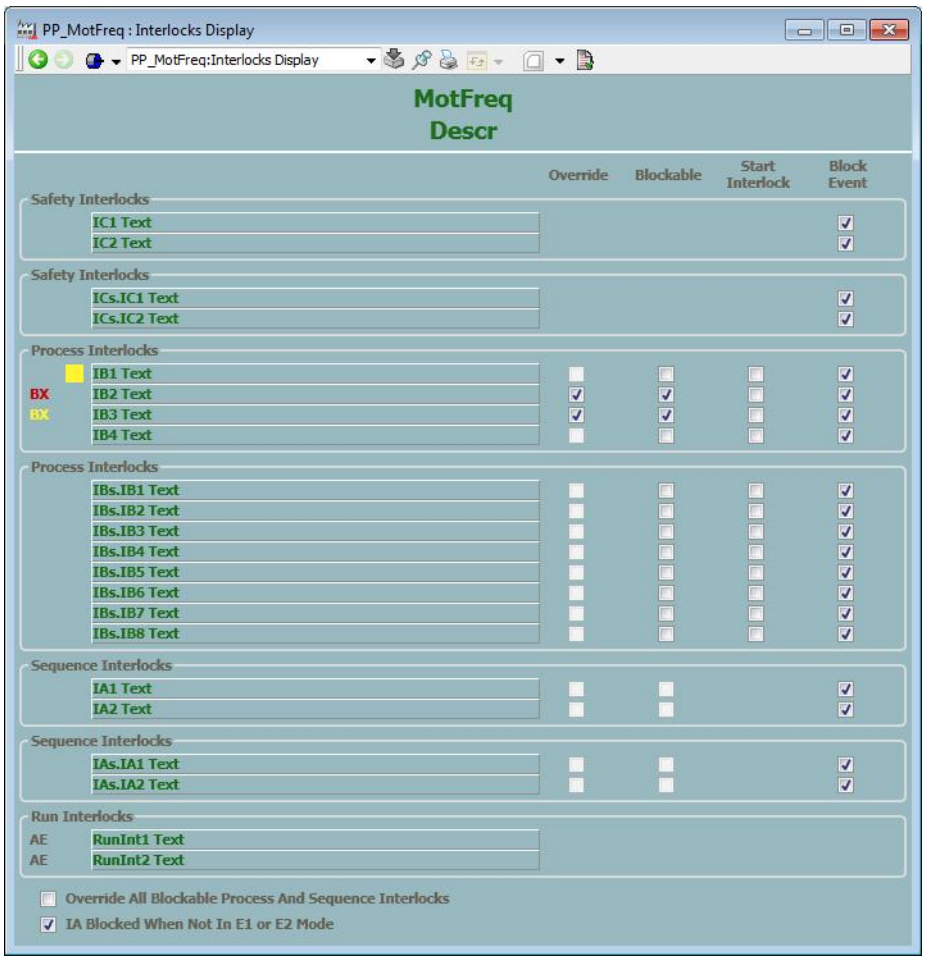
Relevant information about the object can be entered here.



Interlock Display

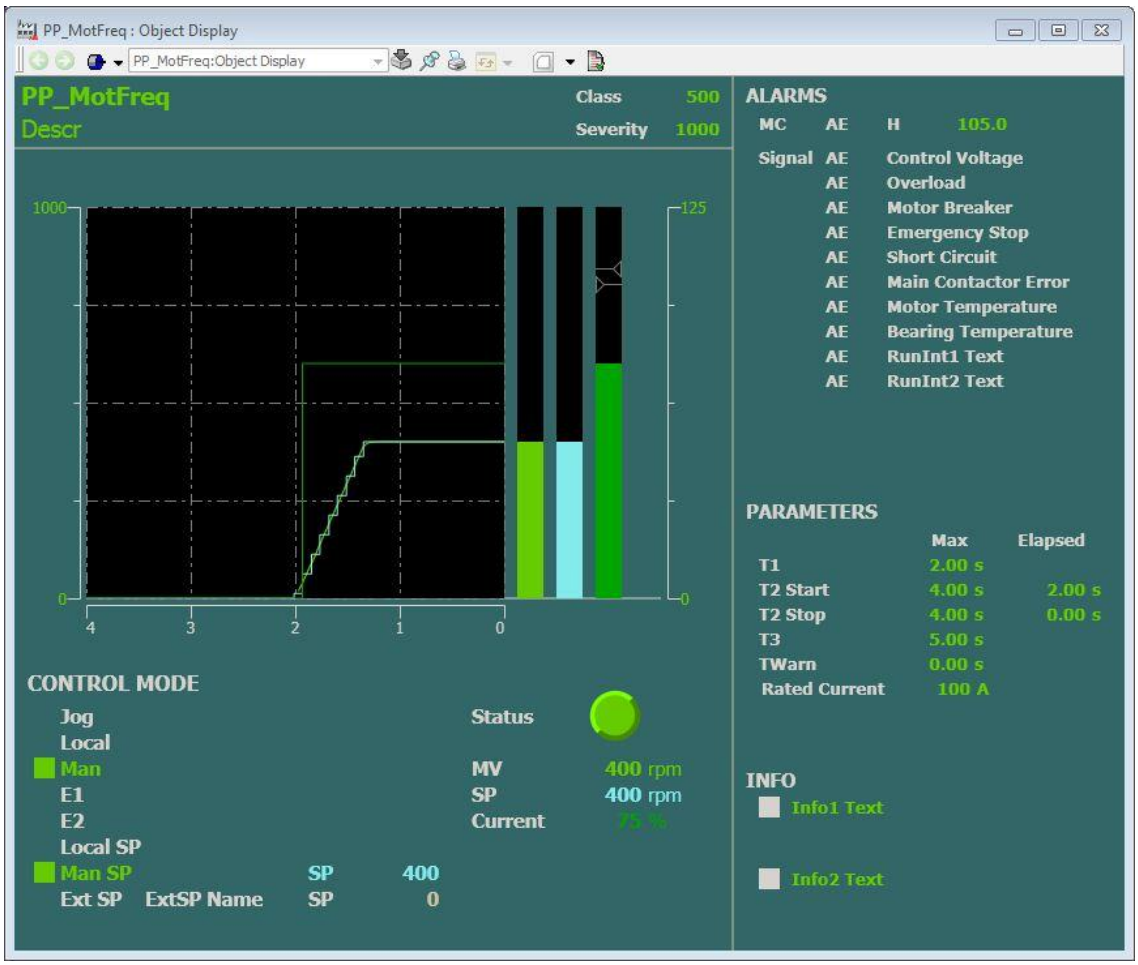
Interlock Display shows the actual status of all interlocks. Operator can override individual interlock or all interlocks which is blockable. **Blockable**, **Start Interlock**, **Block Event** and **Interlock Texts** are set with *Configure* permission.

When interlock is active, a yellow square indication will be shown in front of the interlock. When operator blocks an interlock, a **BX** text indication will be shown. If the interlock is active, it will be in red color. Otherwise it will be in yellow color.



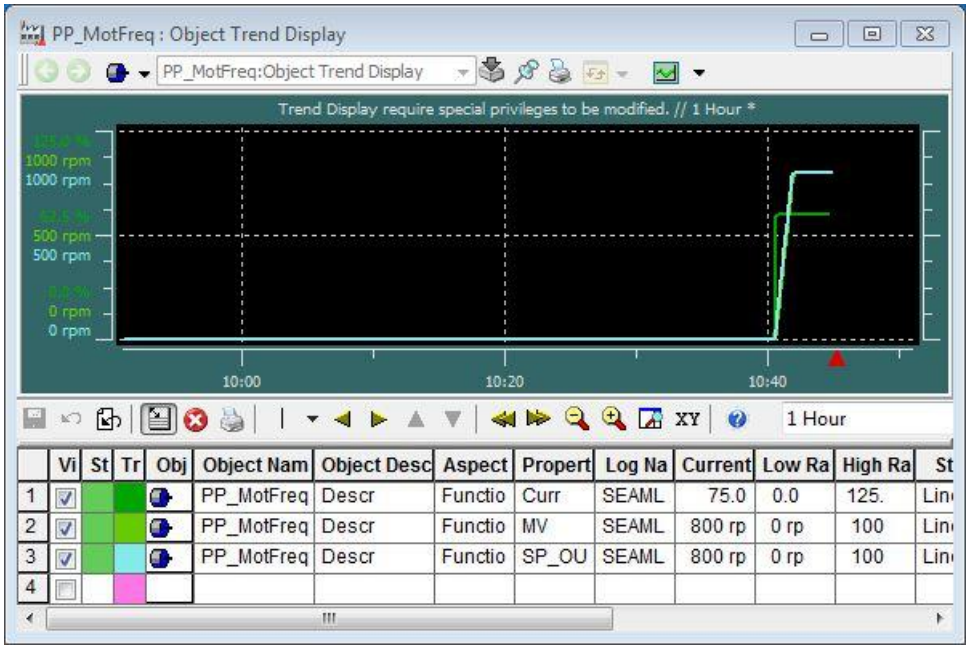
Object Display

Object Display can be used to get the complete overview of the object. It has detailed information of the motor status, speed value and range, current value and range, signal status, parameters and available control modes.



Object Trend Display

Values stored can be presented graphically in the form of curves in **Object Trend Display** aspect. By default, MotFreq presents 3 traces: **Curr**, **MV** and **SP_OUT**. It uses the trend template **PP Library Motor Object Trend** in Library Structure.



Alarm & Event

Alarm & Event List

All alarm & event list aspect is configured to follow the templates defined in **PP Library Alarm & Event List Configurations** in Library Structure.

Alarm Message

The **Message Description** is hardcoded in the function block and cannot be modified. The **Condition** text is stored in the **AE Translator** aspect in Object Type Structure and support NLS functionality.

Resource Id	Condition	Message Description
BT	Bearing Temperature	Alarm
HC	High Current	Alarm
M1	M1, Control Voltage	Fault
M2	M2, Overload	Fault
M3	M3, Motor Breaker	Fault
M4	M4, Emergency Stop	Fault
M5	M5, Short Circuit	Fault
MCE	Main Contactor Error	Fault
MT	Motor Temperature	Alarm
RInt1	Run Interlock 1	Fault
RInt2	Run Interlock 2	Fault

Event Message

The **Message Description** is stored in the **AE Translator** aspect in Object Type Structure and support NLS functionality.

Resource Id	Message Description
AlarmAck	Alarm acknowledged by program
AlcBlk	Alarm blocked by program
ExtStart	Ext Start
ExtStop	Ext Stop
IA1Off	IA1 Off
IA1On	IA1 On
IA2Off	IA2 Off
IA2On	IA2 On
IAs1Off	IAs.IA1 Off
IAs1On	IAs.IA1 On
IAs2Off	IAs.IA2 Off
IAs2On	IAs.IA2 On
IB1Off	IB1 Off
IB1On	IB1 On
IB2Off	IB2 Off
IB2On	IB2 On
IB3Off	IB3 Off
IB3On	IB3 On
IB4Off	IB4 Off
IB4On	IB4 On
IBs1Off	IBs.IB1 Off
IBs1On	IBs.IB1 On
IBs2Off	IBs.IB2 Off

Resource Id	Message Description
IBs2On	IBs.IB2 On
IBs3Off	IBs.IB3 Off
IBs3On	IBs.IB3 On
IBs4Off	IBs.IB4 Off
IBs4On	IBs.IB4 On
IBs5Off	IBs.IB5 Off
IBs5On	IBs.IB5 On
IBs6Off	IBs.IB6 Off
IBs6On	IBs.IB6 On
IBs7Off	IBs.IB7 Off
IBs7On	IBs.IB7 On
IBs8Off	IBs.IB8 Off
IBs8On	IBs.IB8 On
IC1Off	IC1 Off
IC1On	IC1 On
IC2Off	IC2 Off
IC2On	IC2 On
ICs1Off	ICs.IC1 Off
ICs1On	ICs.IC1 On
ICs2Off	ICs.IC2 Off
ICs2On	ICs.IC2 On
JogEnblOff	JogEnbl Off
JogEnblOn	JogEnbl On
LEnblOff	LEnbl Off
LEnblOn	LEnbl On
MV_LEnblOff	MV_LEnbl Off
MV_LEnblOn	MV_LEnbl On
SeqE1	SeqE1
SeqE2	SeqE2
SeqExtSP	SeqExtSP
SeqMan	SeqMan
SeqManSP	SeqManSP