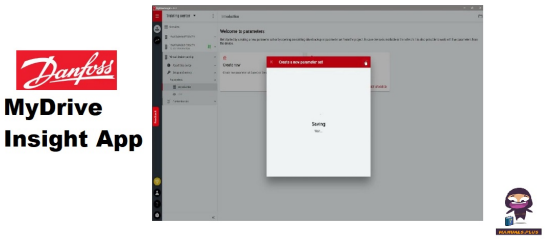
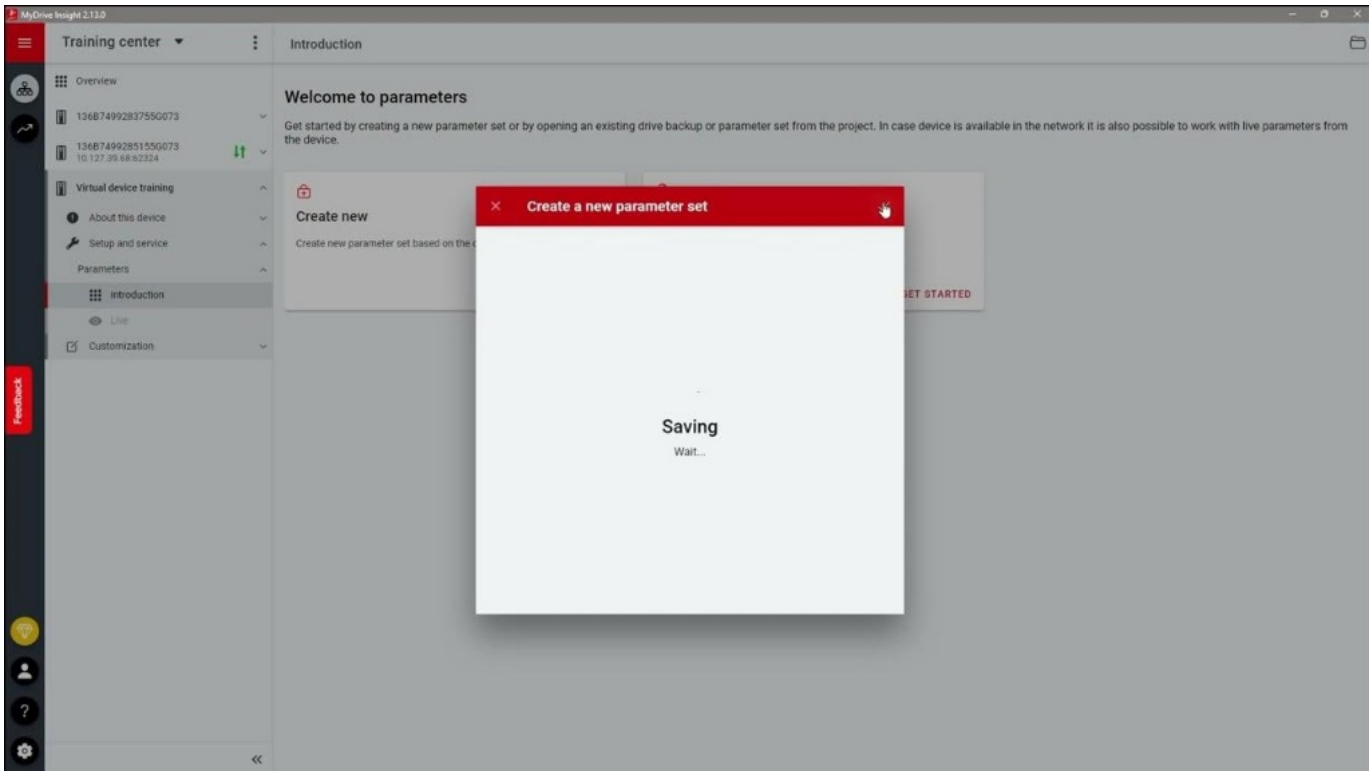


Danfoss MyDrive Insight App User Guide

Contents

1 Danfoss MyDrive Insight App
2 Product Information
3 Product Usage Instructions
4 Introduction
5 General
6 Examples
7 Function blocks
8 Error Handling
9 Documents / Resources
9.1 References
10 Related Posts

Danfoss MyDrive Insight App



Product Information

Introduction

Logic is a versatile feature that allows the customization and control of the operation of the drive without the need for a separate programming tool or language. By utilizing Logic, the operation of the drive can be freely customized using a fixed number of programmable function blocks. Each function block has three inputs and one output, and the functionality of these blocks can be selected from a comprehensive list of IEC61131-3 standard function blocks.

Specifications

- **Automation Product Version:** iC7_Automation-2024.5.43 (GR3)
- **Industry Version:** 4.2.9
- **Motion Version:** 3.1.9

Why Use Logic?

Logic can be used for a wide range of applications and purposes, providing enhanced flexibility and customization options. Some common use cases for Logic include:

1. Conditional Controls: Logic allows for the implementation of conditional controls based on various inputs or parameters.
2. Fault Detection and Diagnostics: Logic can be used to implement fault detection and diagnostics algorithms.

Configuration Tool

Logic can be easily configured using the graphical configuration tool integrated into MyDrive Insight. This tool simplifies the customization process, making it easy to tailor the drive’s operation to specific needs.

Product Usage Instructions

Setting Up Logic Feature

To utilize the Logic feature, follow these steps:

1. Access the graphical configuration tool in MyDrive Insight.
2. Select the desired function blocks from the list of IEC61131-3 standard function blocks.
3. Connect monitoring values or parameters to the block inputs.
4. Configure the output signals of each function block to control the drive’s outputs.

Customizing Logic Operations
To customize the Logic operations:

1. Define specific conditions or criteria for conditional controls.
2. Create fault detection and diagnostics algorithms by monitoring parameters.
3. Set references and control signals based on function block outputs.

FAQ

- **Q: Who is the intended audience for the application guide?**
  A: The intended audience includes trained personnel, automation engineers, and configurators with experience in operating with parameters and basic knowledge of AC drives.
- **Q: How can Logic be configured?**
  A: Logic can be easily configured using the graphical configuration tool integrated into MyDrive Insight.

Introduction

Version History
This guide is regularly reviewed and updated. All suggestions for improvement are welcome. The original language of this guide is English (US).
Table 1: Version History

Version	Automation Product Version	Industry Version	Motion Version	MyDrive® Insight
01	iC7_Automation-2024.5.43 (GR3)	4.2.9	3.1.9	2.15.0

Purpose of this Application Guide
This application guide provides an overview of the Logic feature and its integration into MyDrive® Insight. It covers the following topics:

- What is Logic and its purpose.
- How to configure Logic using MyDrive® Insight.

- Understanding the operation of function blocks within Logic.
- Example configurations to illustrate the usage of Logic.
- A comprehensive list of all available function blocks.
- Error handling within the Logic feature.

Intended Audience
The intended audience of the application guide is trained personnel, automation engineers, and configurators with experience in operating with parameters and with basic knowledge of AC drives.

Additional Resources
Additional resources are available with related information:

- The iC7 Series Motion Application Guide and iC7 Series Industry Application Guide provide information about the Automation applications that support the Logic feature.
- The MyDrive Insight Application Guide covers the general usage of the MyDrive Insight tool.

General

What is Logic?
Logic is a versatile feature that allows the customization and control of the operation of the drive without the need for a separate programming tool or language. By utilizing Logic, the operation of the drive can be freely customized using a fixed number of programmable function blocks.

Logic in MyDrive® Insight extends the features of the drive and provides increased flexibility. Logic enables applying conditional controls, implementing fault detection and diagnostics, creating sequencing, and interlocking logic.
Each function block has three inputs and one output, and the functionality of these blocks can be selected from a comprehensive list of IEC61131-3 standard function blocks. These function blocks are executed sequentially on every application cycle.
Any monitoring value or parameter can be connected to the block inputs using either the parameter name or parameter number. The output signal of each programmable function block can be used as input to another function block or to control the drive's digital or analog outputs. Moreover, the value of most parameters can be freely controlled with the Logic feature. The drive can be directly controlled by the function block outputs through setting references and control signals.
Logic can be easily configured using the graphical configuration tool integrated into MyDrive Insight. This tool simplifies the customization process, making it easy to tailor the drive's operation to specific needs.

Why use Logic?
Logic can be used for a wide range of applications and purposes, providing enhanced flexibility and customization options. Here are some common use cases for Logic:

1. Conditional Controls: Logic allows for the implementation of conditional controls based on various inputs or parameters. Logic can adjust system behavior based on specific conditions, such as drive on time, external events, or other defined criteria.
2. Fault Detection and Diagnostics: Logic can be used to implement fault detection and diagnostics algorithms. By monitoring various parameters and inputs, logic can be created that detects abnormal conditions or faults in the system, enabling proactive maintenance and troubleshooting.

These are just a few examples of what Logic can be used for. The versatility and flexibility of Logic make it a powerful tool for implementing customized functionality and adapting the system's behavior to meet specific

requirements.

Configuration
Logic can be configured inside MyDrive Insight. However, the Logic menu, which is part of the Customization menu, is only accessible if the drive supports the Logic feature and a connection to the drive has been established.

Running Mode

NOTICE

- **RUNNING MODE**
  Before utilizing the Logic feature, it is important to evaluate whether the installation is in a suitable state for making changes to parameters, digital outputs, and analog outputs. Logic can be in the following modes:
  - **Disabled:** The Logic feature is not executed. Outputs and parameters are not affected by the Logic feature.
  - **Programming:** The Logic feature is running in debug mode – blocks are executed but outputs and parameters are not changed by the Logic feature.
  - **Executing:** The outputs are actively driven and reflect the configured Logic behavior.

To configure Logic, stop the execution of Logic by setting it to Programming mode. This allows the configuration process to take place. Once the configuration is complete and ready to be utilized, select Start Executing to initiate the execution of the configured Logic. In cases where the Logic feature is not required, set it to its default Disabled mode. This helps reduce the processing load on the drive and prevents any unnecessary execution of Logic.

Debugging
In Programming and Executing mode in the MyDrive® Insight GUI, it is possible to monitor the live values of block inputs and outputs.

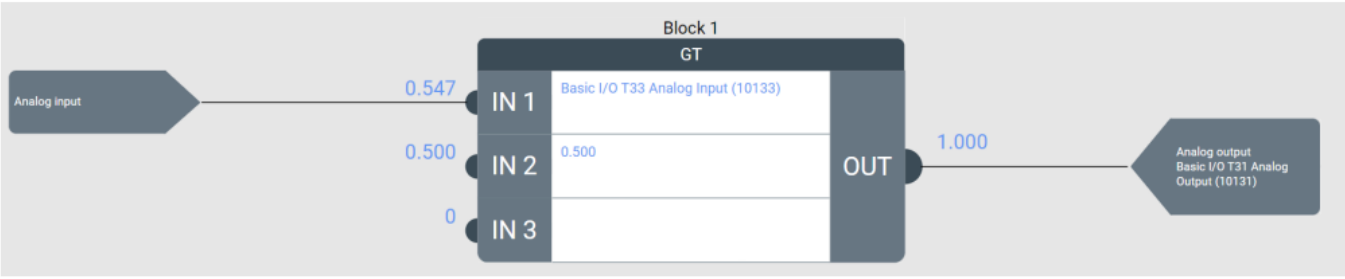


Illustration 2 : Live debugging values shown on the inputs and outputs of a (GreatherThan(GT) Function block

Function Blocks
Each function block in Logic can be configured by selecting the appropriate Function Block Type with the parameter Function Block Type Selection. This parameter provides a wide range of commonly used IEC61131-3 standard function blocks such as AND, OR, MUL, DIV, EQ, GT and more. For a complete list of available function blocks, refer to the Function blocks section at the end of this guide. Every function block consists of three inputs and one output. Each input and output can be configured individually to meet the specific requirements of the application.
Furthermore, each function block has mandatory inputs that must be configured correctly. Failure to configure these inputs correctly results in a warning. For more information on error handling, refer to the Error handling section.

Data Types
In the Logic function, all signals and values are internally handled as floating-point values. However, some of the selectable functions have inputs or outputs defined as boolean values (BOOL), such as AND, OR, RS, and others. These boolean values can have two different states: TRUE or FALSE.
In the case of boolean values, the following conversion rule applies:

- If the input or output is not equal to 0.0, it is considered TRUE.
- If the input or output is equal to 0.0, it is considered FALSE.

For example, if a value of 0.534 is routed to a digital output, the digital output is active because it is interpreted as TRUE.
Similarly, in the example shown in Illustration 3, if an analog input with a value of 0.497 is routed to an OR function, the result is TRUE. Only when the analog input is precisely 0, is it interpreted as FALSE. Therefore, it is often not a good idea to use an analog input as input to a Boolean function block operation.

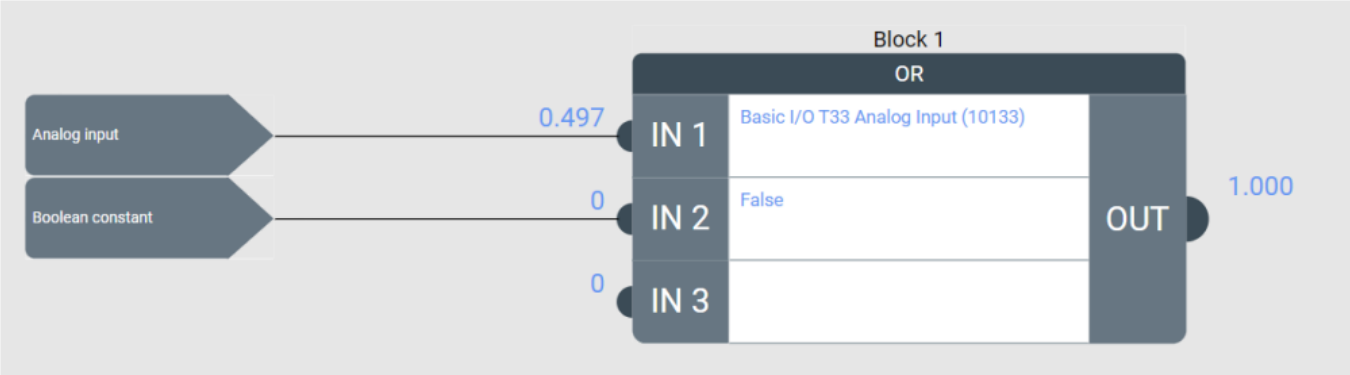


Illustration 3 : Data types

Block Inputs
In the MyDrive® Insight GUI, each input (IN1, IN2, and IN3) has a configuration selection. Click on Input Mode to select the input signal. Depending on the selected input mode, additional configuration options such as Input Value, Bit/Index, and negate/invert become visible.
It is also possible to use the output of one block as the input to another block, enabling the creation of more complex logic configurations.
Some of the input modes are limited to a fixed number of simultaneous instances. For example, a maximum of 10 different digital input terminals can be accessed. See the additional information column in Table 2.
Table 2: Input Modes

Selection name	Description	Additional information
Not used	Functionality is disabled.	Input does not fetch any values fro m any source. It returns 0.0 (FALSE). Input is treated as not configured. If the input is required for an operat or, leaving it not configured triggers a Logic Block Configuration Error E vent.
Digital input	Read the state of a digital input.	The input mode Digital input has a li mit of 10 simultaneous instances.
Parameter bit	This provides a method for fetching a specific bit from a word-type para meter value. The LSB has bit numb er 0.	The input mode Bit in parameter val ue has a limit of 5 simultaneous inst ances.

Boolean constant	Provides a method for setting a boo lean value.	Set an input to constant TRUE or F ALSE
Event active	Provides a method for reading if an event is active. Returns TRUE=1.0, if an event is active.	The input modes Event active and Event group active together have a limit of 5 simultaneous instances. All events can be selected based o n the event number in decimal form or the event name.
Event group active	Provides a method for reading if an y event in an event group is active. Returns TRUE=1.0, if an event is a ctive.	The input modes Event active and Event group active together have a limit of 5 simultaneous instances. All event groups can be selected ba sed on the event group number in h exadecimal form or the event name .
Analog input	Reads the value of an analog input terminal, returning a normalized val ue between 0.0 and 1.0.	The input mode Analog input has a limit of 5 simultaneous instances. A nalog input returns the analog input value scaled between 0.0 and 1.0 a nd not the value in physical units. U sing *Parameter Value* instead retrie ves the analog input status.
Parameter value	This provides a method for fetching a parameter value.	The input mode Parameter read ha s a limit of 10 simultaneous instanc es.
Numeric constant	This provides a method for inputting a numeric constant.	Use 0.0 for FALSE and 1.0 for TRU E if a boolean value is required, or use a Boolean constant input instea d.
Block output	Output value from a selected block i s passed on as the input value.	This makes it possible to link the ou tput of a block to the input of another block.

Block Outputs

In MyDrive® Insight's Logic GUI, the output (OUT) of a function block can be configured by clicking on the configuration field. The output signal, and other configuration options, can be defined based on the selected output mode. These options may include Output Value, Bit/Index, and negate/invert.

Some of the output modes are limited to a fixed number of simultaneous instances. For example, a maximum of 10 different digital input terminals can be accessed. See the additional information column in Table 3.

Using negation for output mode Parameter value means multiplying the value by -1. This may not provide the expected result if you are trying to set a boolean type parameter. Writing an invalid value, such as -1, to a boolean parameter that only accepts 0 or 1 results in a Logic Output Error Event that can be seen in the MyDrive Insight Events view.

Illustration 4: Logic output error event because -1 is not a valid choice for parameter number 106

NOTICE

- **PARAMETER SET LIMITATIONS**
  The parameters in the motor configuration cannot be adjusted while the motor is running. If the parameter cannot be written or if the parameter value does not meet the specified limits, a Logic Output Error Event is shown in the Events view, similar to illustration 4.
  Additionally, certain parameters are designated as ReadOnly and are not accessible for modification. Those parameters are therefore not shown in the Logic ParamOut selection list.

Table 3: Output Modes

Selection name	Description	Additional information
Not used	Block does not apply result anywhere.	
Digital output	Block output is applied to the selected digital output terminal.	Digital output has a limit of 5 simultaneous instances.
Analog output	Block output is applied to the selected analog output terminal.	Analog output has a limit of 5 simultaneous instances. The analog output is limited to values between 0.0 and 1.0 as it is giving the internal reference for the analog output. The value is then converted to physical units based on the analog output mode and min. and max. value configurations in the application parameters.
Parameter value	Block output is applied to the selected parameter.	Parameter write has a limit of 5 simultaneous instances. ReadOnly parameters cannot be written. Parameter values must be within the parameter's limits and a valid selection when limited

Selection name	Description	Additional information
		by a selection list. Otherwise, a Log ic Block Output Error event is trigge red.
Logic speed reference	Block output is applied to the Logic speed reference.	This provides a method for setting a speed reference directly from Logic. The Logic reference can be selecte d as source in the control place configuration. Speed reference cou nts as a parameter write.
Logic torque reference	Block output is applied to the Logic torque reference.	This provides a method for setting a torque reference directly from Logic . The Logic reference can be select ed as source in the control place co nfiguration. Torque reference count s as a parameter write.
Logic process reference	Block output is applied to the Logic process reference.	This provides a method for setting a process reference directly from Logi c. The Logic reference can be selec ted as source in the control place configuration. Process reference co unts as a parameter write.
Logic position reference	Block output is applied to the Logic position reference.	This provides a method for setting a position reference directly from Logi c. The Logic reference can be selec ted as source in the control place configuration. Position reference co unts as a parameter write.

Logic Virtual Terminals

Logic in MyDrive® Insight includes virtual terminals that serve as storage for signals, allowing them to be used as inputs in other application features. Data written to a virtual terminal output can be read by its respective virtual terminal input.

For example, a function block's output can set the virtual digital output Logic Digital I/O 1. The virtual terminal's state can be accessed by selecting the same virtual terminal as an input by setting parameter 4722 Advanced Start Input as Logic Digital I/O 1. The Logic virtual digital terminals can be selected wherever virtual terminals are available.

Table 4: Logic virtual terminals

Terminal name	Description
Logic Digital I/O 1	Logic virtual digital I/O terminal 1.
Logic Digital I/O 2	Logic virtual digital I/O terminal 2.
Logic Digital I/O 3	Logic virtual digital I/O terminal 3.
Logic Digital I/O 4	Logic virtual digital I/O terminal 4.

Saving the Logic configuration

There is no need for a separate save command for the Logic configuration. The Logic configuration is parameter-based, which means it is treated like any other parameter, and is saved when creating a backup and can then be restored. It is not possible to save the Logic configuration separately.

Examples

To follow the examples, an open MyDrive® Insight instance with a connected iC7 drive that supports the Logic feature is required.

Start based on Analog input T33

Description:

- The drive is controlled from I/O
- Frequency reference is given by analog input (T33)
- The drive is started when the T33 signal exceeds 50% and stopped when the signal goes below 40%.

In this example, the drive is controlled using I/O, and the frequency reference is provided by an analog input, specifically T33. The objective is to start the drive when the T33 signal exceeds a threshold level of 50% and stop it when the signal drops below a hysteresis level of 40%.
Analog Input T33 is already configured as the frequency reference in the drive's default settings. This example extends its functionality to include the start command based on the analog input level.
To implement this logic, use a GreaterThan (GT) function block to compare the analog input against the 50% threshold and a LessThan (LT) function block to compare it against the 40% hysteresis threshold. Additionally, an RS flip-flop can be used to latch the Start signal based on the results of these two comparisons.
Configure the function blocks and connect the inputs and outputs appropriately to create a logic configuration that enables the drive to start when the analog input exceeds the threshold and stop when it falls below the hysteresis level.
To configure both Logic and the drive's parameters, follow these steps:

1. Open the Logic view in MyDrive® Insight and set the running mode to Programming by clicking on Enable Logic in the top right corner. This allows the Logic configuration to be changed and stops the block program execution.
2. Use Block1 for the GreaterThan function.
    - **a.** Select Block1 and select the function GT.
    - **b.** Configure IN1 as Analog input connected to Basic I/O T33 Analog Input.
    - **c.** Configure IN2 as Numeric Constant set to 0.5.
3. Use Block2 for the LessThan function. Select Block2 and select the function LT. Configure IN1 as Analog input connected to Basic I/O T33 Analog Input. Configure IN2 as Numeric Constant set to 0.4.
4. Use Block3 for the decision. Select Block3 and select the function RS. Configure IN1 as Block output connected to Block 1. Configure IN2 as Block output set to Block 2. Configure OUT as Digital Output connected to Logic Digital I/O 1.
5. Go to the Parameters view, parameter group 5.5.6.1. Configure parameter 4722 Advanced Start Input Index 1 to use Logic Digital I/O 1. Advanced Control Place must be used, since it is the only control place that supports virtual I/Os.
6. Go back to the Logic view. Test the function by checking the debugging values. Once everything is working as expected, set Logic's running mode to Executing by selecting Start Executing. Now Logic will drive the outputs.

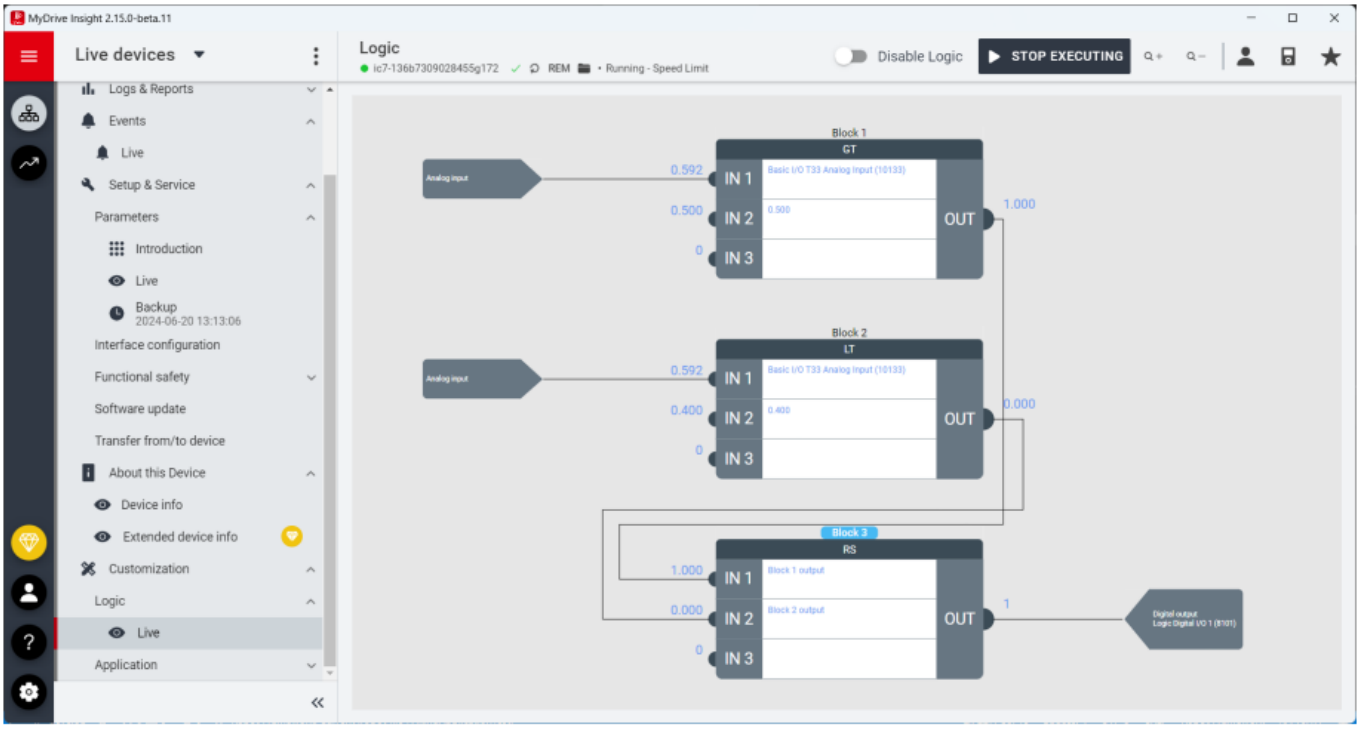


Illustration 5 : Example: Start based on Analog input T33

Scaling Motor Torque Limit by Analog input T34

- Description:
  - The drive is controlled from I/O
  - Frequency reference is given by analog input (T33)
  - Motor Torque Limit is changed linearly between 0…300% by analog input (T34)
- To change the value of parameter 1810 Positive Torque Limit using Logic, follow these steps:
  1. Open the Logic view in MyDrive® Insight and set the Running mode to Programming by clicking on Enable Logic in the top right corner.
  2. Use Block1 for the multiplication function.
     - **a.** Select Block1 and select the function MUL.
     - **b.** Configure IN1 as Analog input connected to Basic I/O T34 Analog Input.
     - **c.** Configure IN2 as Numeric constant set to 300.0.
     - **d.** Select the output mode Parameter value and select the parameter 1810 Positive Torque Limit.
  3. Test the function by checking the debugging values. Once everything is working as expected, change the running mode to Executing by selecting Start Executing. Now, the Positive Torque Limit is set based on the analog input (T34), with a scaling of 0–300%.

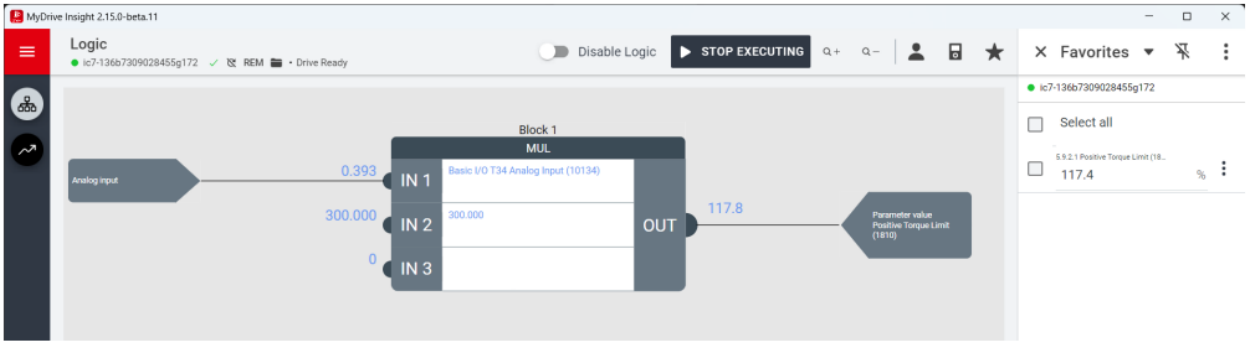


Illustration 6 : Example: Start based on Analog input T34

Delayed and conditional external fault

This example shows how to get some extra conditions added to the external fault-triggering logic. By default, the external event is just a simple on/off type of logic connected, for example, to a digital input (T15). This example shows how to allow the fault triggering from digital input (T15) while the drive is in run mode and use a 2 second ON-Delay.
The best way to solve this issue is by stripping it down into two steps. The first step is to handle the conditional rule of only triggering when both the digital input T15 is active, and the drive is running. The second step is the ON-Delay handling which can be handled by the existing implementation in the application with parameter 4592 External Event 1 Delay.

To implement delayed external fault with additional conditions, follow these steps:

1. Open the Logic view in MyDrive® Insight and set the Running mode to Programming by clicking on Enable Logic in the top right corner.
2. Use Block1 for the conditional function.
    - **a.** Select Block1 and select the function AND.
    - **b.** Configure IN1 as Digital input connected to Basic I/O T15 Digital Input.
    - **c.** Configure IN2 as Parameter Bit set to Motor Ctrl. Status Word and select Bit value 1 to retrieve the Run bit as defined in the application guideline.
    - **d.** Configure OUT as output mode: Digital output and select the virtual Logic terminal Logic Digital I/O 1.
3. Go to the Parameters view, parameter group 5.2.2. Configure parameter 4557 External Event 1 Input, as Logic Digital I/O 1. Select the desired response. By default, parameter 4559 External Event 1 Response, is set to Fault, ramp to coast. Set the desired ON-Delay by changing parameter 4592 External Event 1 Delay, to 2 s.
4. Return to the Logic view. Test the function by checking the debugging values. Once everything is working as expected, change the running mode to Executing by selecting Start Executing. Now, the virtual terminal Logic Digital I/O 1 activates when both Digital input T15 is active and the drive is running, and an external event is triggered based on it with a 2-second delay.

Illustration 7 : Example: Delayed external fault with condition

Custom scaling of status parameter to drive analog output
This example shows how to scale a signal and output it on an analog output. This is useful when the parameter or signal cannot be selected to be written to an analog output within the application or the application does not offer the desired scaling.
For example, parameter 2305 Motor Power Output offers the possibility to select an output for the motor power signal. The scaling of the signal is 0–100% of the nominal power.
The Logic feature can be used with customized scaling to handle overloads. In this example, the Motor Power Output is scaled as 0–300% of nominal power and output on Basic I/O T31 Analog Output.

1. Open the Logic view in MyDrive® Insight and set the running mode to Programming by clicking on Enable Logic in the top right corner.
2. Use Block1 for the scaling function: OUT = (Motor Shaft Power (kW) * 1/3) / (Nominal Power (kW)).
    - **a.** Select Block1 and select the function MULDIV.
    - **b.** Configure IN1 as Parameter value connected to parameter 9008 Motor Shaft Power.

- **c.** Configure IN2 as Numeric constant set to 0.3333 to scale to 300% of Nominal Power.
- **d.** Configure IN3 as Parameter value set to parameter 405 Nominal Power.
- **e.** Configure OUT as output mode: Analog output and select the analog terminal: Basic I/O T31 Analog Output.

3. Make sure that the analog output T31 is configured as desired by configuring the parameters of parameter group 9.5.1 Output T31. Once everything is working as expected, change the running mode to Executing by selecting Start Executing. Now, the analog terminal Basic I/O T31 Analog Output shows the motor power scaled as a range of 0–300% of the nominal power.

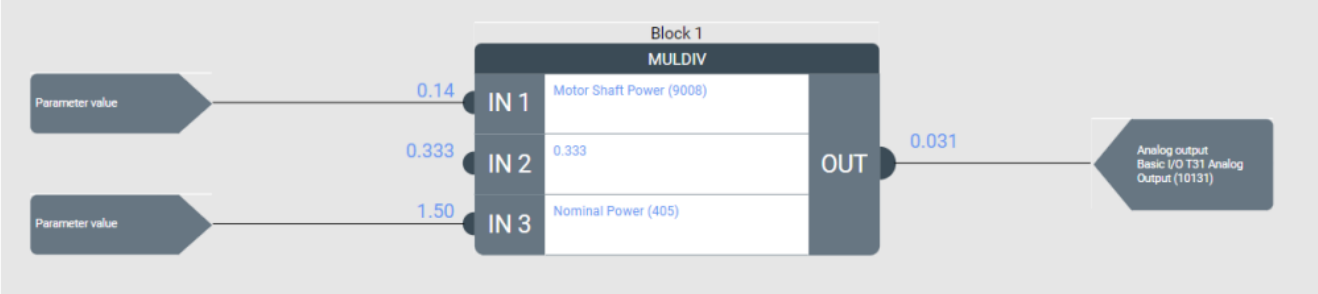


Illustration 8 : Example: Custom scaling of status parameters

Function blocks

The functions available in the logic blocks can be categorized into four groups:

1. Logic and Bit Operations: These functions provide boolean operators for common boolean algebra. They are used to perform logical operations on boolean signals.
2. Math Operations: These functions provide numeric operators for elementary arithmetic operations. They are used to perform mathematical calculations on numeric values.
3. Comparators: These functions provide comparative logic for numeric values. They are used to compare two values and determine their relationship, such as equality, inequality, or order.
4. Special Operators: These functions can combine logic and arithmetic operations. They are used for advanced or specialized operations.

Table 5: List of available function blocks

Logic operations	Math operations	Comparators	Special operations
AND	ADD	EQ	DELAY
OR	SUB	NE	SEL
NAND	MUL	GE	
NOR	DIV	GT	
NOT	MULDIV	LE	
XOR	NEG	LT	
R_TRIG	ABS	GT_LT	
F_TRIG	SQRT		
RS	NORMALIZE		
SR	LIMIT		
	MIN		
	MAX		
	MEAN		
	FILTER		
	MODULUS		

Logic and Bit Operations
In logic operations, all inputs are converted to booleans. A numeric value of 0.0 is converted to boolean FALSE, while all other values are treated as boolean TRUE.
If an input is not configured, it is considered FALSE. Optional inputs are ignored if they are not configured.
Table 6: List of available logic operators

Operator	Description	Detailed description
AND	Logical AND-function. Output = Input1 AND Input2 AND Input3(optional).	
OR	Logical OR-function. Output = Input1 OR Input2 OR Input3(optional).	
NAND	Logical NAND-function. Output = Input1 NAND Input2 NAND Input3(optional).	
NOR	Logical NOR-function. Output = Input1	

Operator	Description	Detailed description
	NOR Input2 NOR Input3(optional).	
NOT	Logical NOT-function. Output = NOT Input1.	
XOR	Logical XOR-function. Output = Input1 XOR Input2 XOR Input3(optional).	
R_TRIG	Detects a rising edge on any of the inputs, setting the output TRUE for one cycle. Output = Rising edge on Input1(optional) OR Input2(optional) OR Input3(optional). At least one of the inputs must be configured.	The rising edge trigger block can detect a rising edge in a boolean signal and switches its output from FALSE to TRUE. The output stays active for one execution cycle (application dependent, for example Industry: 5 ms) in case of a detected rising edge.
F_TRIG	Detects a falling edge on any of the inputs, setting the output TRUE for one cycle. Output = Falling edge on Input1(optional) OR Input2(optional) OR Input3(optional). At least one of the inputs must be configured.	The falling edge trigger block can detect a falling edge in a boolean signal and switches its output from FALSE to TRUE. The output stays active for one execution cycle (application dependent, for example Industry: 5 ms) in case of a detected falling edge. At least one of the inputs must be configured.
RS	RS flipflop – RESET has priority if both TRUE. SET = Input1, RESET = Input2, Output = FF State.	The output is reset (OUT=0) if the RESET input is TRUE (≠0), regardless of the state of the SET input. If the SET input is TRUE(≠0) and RESET is FALSE(=0), the output pin is set (OUT=1). If both the inputs are FALSE (=0), the output preserves its previous value.
SR	SR flipflop – SET has priority if both TRUE. SET = Input1, RESET = Input2, Output = FF State.	The output is set (OUT=1) if the SET input is TRUE(≠0), regardless of the state of the RESET input. If the RESET input is TRUE and SET is FALSE(=0), the output pin is cleared (OUT=0). If both the inputs are FALSE, the output preserves its previous value.

Math Operations

Arithmetic operations are working with numeric values. All inputs are converted to numeric values. Non-configured inputs are always considered 0.0, and optional inputs are ignored if not configured.

Table 7: List of available math operators

Operator	Description	Detailed description
ADD	Summarize function. Output = Input1 + Input2 + Input3 (optional).	Sum of all inputs.
SUB	Subtract function. Output = Input1 – Input2 – Input3 (optional).	Input1 minus Input2 and Input3.
MUL	Multiply function. Output = Input1 x Input2 x Input3 (optional).	Input1 multiplied with Input2 (optionally multiplied a gain with Input3).
DIV	Divide function. Output = Input1 / Input2 / Input3 (optional).	Input1 divided with Input2 (optionally divided again with Input3).
MULDIV	Combined multiply and divide function. Output = Input1 x Input2 / Input3.	Combines multiplication and division of inputs 1…3.
NEG	Negate function. Output = (-1)* Input1.	Negated value of Input1.

Operator	Description	Detailed description
ABS	Absolute value function – removes the sign from a value. Output = ABS (Input1).	Absolute value of Input1.
SQRT	Square root function. Output = SQRT (Input1).	Square root of Input1.
NORMALIZE	Normalize function. Scale Input1 between Input2 (min) and Input3 (max). Input 2 must be smaller than Input3. Output is limited between 0.0 and 1.0. Output = ((MIN(MAX(Input1, Input2),Input3)) – Input2) / (Input3 – Input2).	Rescale (min-max normalization) Input between Input2 (min) and Input3 (max). Output is limited between 0.0 and 1.0. Input2 must be smaller than Input3. [illegible] Example: Dynamically rescale Analog Input T33 between T34 (equivalent to 0%) and 10 V (100%). NORMALIZE(T33 Analog Input Value(1611), T34 Analog Input Value(1612), 10.0) For example: IN1: T33=5.75, IN2: T34= 2.5, IN3: 10.0 OUT = Normalize(IN1,IN2,IN3) = (5.75-2.5)/(10.0-2.5)=0.433

LIMIT	Limit function. Input2 must be smaller than Input3. Output = MIN(MAX(Input1, Input2), Input3).	Returns Input1 if it lies between min=Input2 and max=Input3, otherwise it returns the violated limit. Input2 must be smaller than Input3. Example: Return the analog input value but only within allowed limits of 2.0-5.0 V. LIMIT(T33 Analog Input Value(1611),2.0, 5.0) For example: IN1:T33=6.1V, IN2(Min)=2.0, IN3(Max)=5.0 OUT= LIMIT(IN1,IN2, IN3) = LIMIT(6.1V>5.0) = 5.0
MIN	Minimum function. Output = MIN(MIN(Input1, Input2), Input3(optional)).	Returns the smallest value of Input1, Input2, and Input3. If Input3 is not configured, it's ignored. Inputs 1 and 2 are always considered, so if they are not configured, the value is 0.0.
MAX	Maximum function. Output = MAX(MAX(Input1, Input2), Input3(optional)).	Returns the largest value of Input1, Input2, and Input3. If Input3 is not configured, it's ignored. Inputs 1 and 2 are always considered, so if they are not configured, the value is 0.0.
MEAN	Mean function. At least one of the inputs must be configured. Output = (Input1(optional) + Input2(optional) + Input3 (optional)) / 'number of configured inputs'.	Returns the average value of the selected inputs – if an input is not configured, it is not considered.
FILTER	First-order low-pass filter function. Output = LowPass(Input1), Filter time = Input 2 [s].	The low-pass filter returns a filtered value of Input1.
MODULUS	Integer modulus division function. Output = DINT(Input1) MOD DINT(Input2).	This arithmetic function divides the operand connected to Input1 by the operand connected to Input2 and returns the remainder of the division. The modulus operation treats the inputs as double integers and returns a double integer value. Example: Return how many sets of 5 revolutions the encoder has detected. MOD(Channel 1 Revolutions (4066), 5) e.g. IN1:17 IN2:5 OUT=MOD(17,5)=3

Comparators

Comparative logic for numeric values – all inputs are converted to numerics. Non-configured inputs are always considered 0.0. By default, the optional tolerance value is 10^-6 if not configured. Output is a boolean that can be used for boolean operators.

Table 8: List of available comparator blocks

Operator	Description	Detailed description
EQ	Equal function. Output = Input1 == Input2, Input3(optional) = tolerance.	Input1 and Input2 have the same values.
NE	Not Equal function. Output = Input1 != Input2, Input3(optional) = tolerance.	Input1 and Input2 have different values.
GE	Greater or Equal function. Output = Input1 >= Input2, Input3(optional) = tolerance.	Input1 is Greater or Equal to Input2.
GT	Greater Than function. Output = Input1 > Input2.	Input1 is Greater Than Input2.
LE	Less or Equal function. Output = Input1 <= Input2, Input3(optional) = tolerance.	Input1 is Less or Equal than Input2.
LT	Less Than function. Output = Input1 < Input2.	Input1 is Less Than Input2.
GT_LT	Between but not equal to limits. Output = Input2 < Input1 < Input3.	Input1 is Greater Than Input2 and Less Than Input3.

Special Operators
Non-configured inputs are always considered 0.0 / FALSE. Output depends on the operator.
Table 9: List of available special operators

Operator	Description	Detailed description
DELAY	Delay boolean signal with a separate delay for ON and OFF. Output = delayed state of BOOL(Input1), ON-delay time [s] = Input2(optional), OFF-delay time [s] = Input3(optional). Input1 and at least one of the delays must be configured.	
SEL	Select/relay function. Input2 and Input3 can be either numeric or boolean values. Output = SEL(BOOL(Input1), Input2 (input1=FALSE), Input3 (input1=TRUE)).	OUT = SEL(IN1, IN2, IN3) OUT = IN2, if IN1 = FALSE OUT = IN2, if IN1 = TRUE

Error Handling

When configuring or executing Logic, it is important to check the active Events Log to see if any configuration errors have occurred. Logic can detect some configuration errors and issue a warning.

LOGIC ERROR HANDLING
If a Logic Input Error or Logic Block Configuration Error is detected, the Logic outputs are not set and remain at their last value. This prevents incorrect or unintended outputs from being generated. If more than one function block output is configured to drive the same output (DigOut, AnOut or Parameter). Logic sets the output to the last value assigned. Therefore, the output from the block with the highest number drives that output signal since the function blocks are executed sequentially.

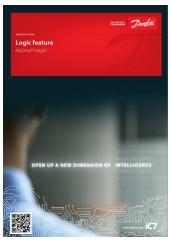
Table 10: Logic error handling

Name	Event number	Event group	Description	Example
Logic Input Error	5901	0xFF06	Configured input function reports an error.	Input mode= Digital input is selected, but no terminal is selected, leaving the default *None*. Since *None* is not a valid Digital input terminal, a Logic Input Error is triggered.
Logic Output Error	5902	0xFF06	Configured output function reports an error.	Output mode= Analog output is selected, but no terminal is selected, leaving the default *None*. Since *None* is not a valid Analog output terminal, a Logic Output Error is triggered once Logic is set into *Executing mode*.
Logic Block Configuration Error	5903	0xFF06	Logic block configuration is incorrect.	Not all required inputs are configured, or the inputs given to the function do not follow the input requirements. For example, NORMALIZE requires all three inputs to be configured and IN2<IN3.
Logic Input Mode Instances Occupied	5904	0xFF06	The logic input instances of an input mode are all occupied. Use fewer of the same input mode configurations.	Trying to read more than 5 unique Analog inputs inside Logic by configuring 6 input modes to read from 6 different terminals triggers an error, since only 5 simultaneous instances are supported.
Logic Output Mode Instances Occupied	5905	0xFF06	The logic output instances of an output mode are all occupied. Use fewer of the same output mode configurations.	Trying to set more than 5 unique Analog outputs inside Logic by configuring 6 output modes to set 6 different analog terminals triggers an error, since only 5 simultaneous instances are supported.

Danfoss A/S Ulsnaes 1
DK-6300 Graasten
drives.danfoss.com

Any information, including, but not limited to information on selection of product, its application or use, product design, weight, dimensions, capacity or any other technical data in product manuals, catalog descriptions, advertisements, etc. and whether made available in writing, orally, electronically, online or via download, shall be considered informative, and is only binding if and to the extent, explicit reference is made in a quotation or order confirmation. Danfoss cannot accept any responsibility for possible errors in catalogs, brochures, videos and other material. Danfoss reserves the right to alter its products without notice. This also applies to products ordered but not delivered provided that such alterations can be made without changes to form, fit or function of the product. All trademarks in this material are property of Danfoss A/S or Danfoss group companies. Danfoss and the Danfoss logo are trademarks of Danfoss A/S. All rights reserved.

Documents / Resources

	**Danfoss MyDrive Insight App** [pdf] User Guide MyDrive Insight App, App

References

- **Global AC drive manufacturer - Danfoss Drives | Danfoss**
- **User Manual**

Manuals+, **Privacy Policy**

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.