ABB Drives

Application Guide
Adaptive Program

Adaptive Program

Application Guide

3AFE64527274 Rev C
EN
EFFECTIVE: 08.04.2005

Table of Contents

Introduction to the guide

Chapter overview

The chapter gives general information on the guide.

Compatibility

The guide complies with the drive application programs in which the Adaptive Programming features are included.

Safety instructions

Follow all safety instructions delivered with the drive.

- Read the **complete safety instructions** before you install, commission or use the drive. The complete safety instructions are given at the beginning of the Hardware Manual.
- Read the **software function specific warnings and notes** before changing the default settings of the function. For each function, the warnings and notes are given in the Firmware Manual in the subsection describing the related user-adjustable parameters.

Reader

The reader of the manual is expected to:

- know the standard electrical wiring practices, electronic components and electrical schematic symbols.
- have no experience or training in installing, operating or servicing of ABB drives.

Use

The guide is to be used together with the firmware manual of the drive application program. The firmware manual contains the basic information on the drive parameters including the parameters of the Adaptive Program. The guide gives more detailed information on the Adaptive Program:

- what the Adaptive Program is
- how to build a program
- how the function blocks operate
- how to document the program
- the parameters and actual signals of ACS800 Standard Application Program essential for the Adaptive Program.

Related publications

The user documentation of the drive also includes:

- Firmware manual (the appropriate manual is delivered with the unit)
- Hardware manual (the appropriate manual is delivered with the unit)
- Guides/supplements for the optional equipment and programs (appropriate manuals are included in the delivery).

Adaptive Program

Chapter overview

The chapter describes the basics of the Adaptive Program and instructs in building a program.

What is the Adaptive Program

Conventionally, the user can control the operation of the drive by parameters. Each parameter has a fixed set of choices or a setting range. The parameters make the programming easy, but the choices are limited: you cannot customise the operation any further. The Adaptive Program makes freer customising possible without the need of a special programming tool or language:

- The program is built of function blocks.
- The control panel is the programming tool.
- The user can document the program by drawing it on block diagram template sheets.

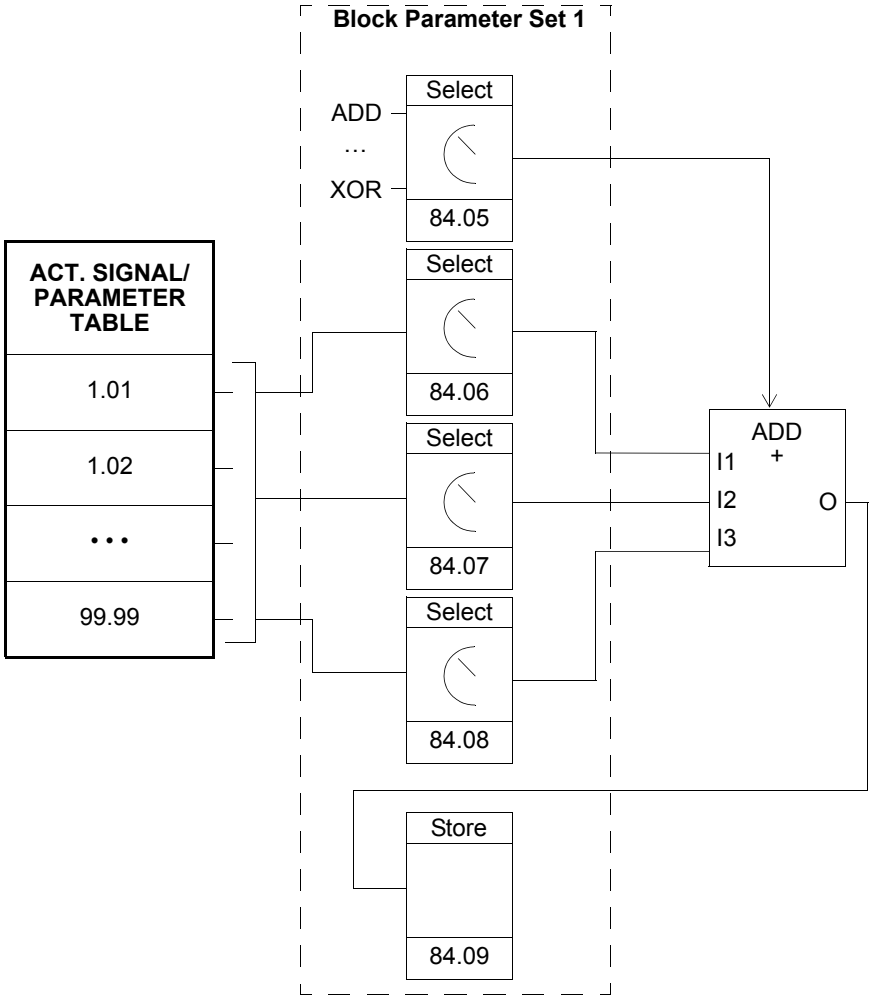
The maximum size of the Adaptive Program is 15 function blocks. The program may consist of several separate functions.

How to build the program

The programmer connects a function block to other blocks through a Block Parameter Set. The sets are also used for reading values from the drive application program and transferring data to the drive application program. Each Block Parameter Set consists of five parameters.

The figure shows the use of Block Parameter Set 1 in the ACS800 Standard Application Program (parameters 84.05 to 84.09):

- Parameter 84.05 selects the function block type.
- Parameter 84.06 selects the source that input I1 of the function block is connected to.
- Parameter 84.07 selects the source that input I2 of the function block is connected to.
- Parameter 84.08 selects the source that input I3 of the function block is connected to.
- Parameter 84.09 stores the value of the function block output. The user cannot edit the parameter value.



How to connect the program to the drive application

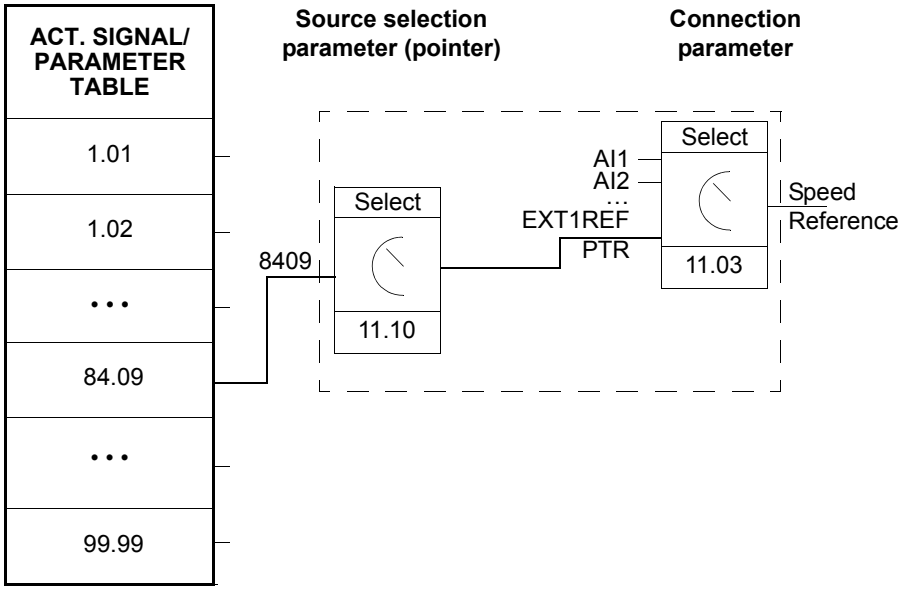
The output of the Adaptive Program needs to be connected to the drive application program. For that purpose the user needs two parameters:

- a connection parameter and
- a source selection parameter (pointer).

The figure below shows the connection principle.

Example:

The Adaptive Program output is stored in parameter 84.09. The diagram shows how to use the value as speed reference REF1 in the ACS800 Standard Application Program.



How to control the execution of the program

The Adaptive Program executes the function blocks in numerical order, all blocks on the same time level. This cannot be changed by the user. The user can:

- select the operation mode of the program (stop, start, editing)
- adjust the execution time level of the program
- delete or add blocks.

Function blocks

Chapter overview

The chapter describes the function blocks.

General rules

The use of input I1 is compulsory (it must not be left unconnected). Use of input I2, I3, etc. is voluntary for the most blocks. As a rule of thumb, a unconnected input does not affect the output of the block.

Block inputs

The blocks use three input formats:

- integer
- boolean
- text string

The used format varies depending on the block. For example, the ADD block uses integer inputs and the OR block boolean inputs. Text string format is used only by the EVENT block.

Note: The inputs of the block are read when the execution of the block starts, not simultaneously for all blocks!

Parameter value as an integer input

How the block handles the input

The block reads the selected value in as an integer.

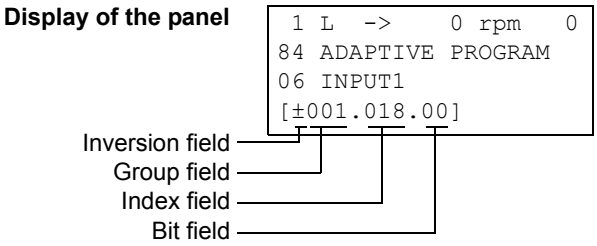
Note: The parameter selected as an input should be a real or integer value. If the value is not in integer format by default, the block converts it. The integer (fieldbus) scaling for each parameter is given in the *Firmware Manual.*

How to select the input

- Scroll to the input selection parameter of the block and switch to edit mode (Enter).
- Set the values of the inversion, group, index and bit fields according to the address from which the input value is to be read (double arrow and arrow keys).

The figure below shows the panel display when the input I1 selection parameter is in edit mode. The value is inverted if there is a minus (-) sign in the inversion field. The bit selection field is not effective for an integer or string type input.

Display of the panel



Example: Analogue input AI1 is 5.8 V in a drive equipped with the ACS800 Standard Application Program. How is the signal connected to the MAX block in the Adaptive Program? What is the value at the block input?

AI1 is connected to the block as follows:

- Scroll to the input I1 selection parameter and shift to edit mode (Enter).
- Set the value in the group field to 1 and the value in the index field to 18. (Value of AI1 is internally stored as actual signal 1.18.)

The value at the input of the block is 5800, since the integer scaling of actual signal 1.18 is: 0.001 V = 1 (given in the *Firmware Manual*).

Constant as an integer input

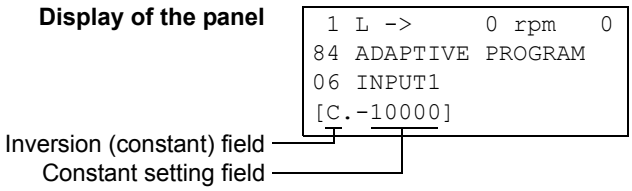
How to set and connect the input

Option 1

- Scroll to the input selection parameter of the block and switch to edit mode (Enter).
- Select C in the inversion field (double arrow and arrow keys). The appearance of the row changes. The rest of the line is now a constant field.
- Give the constant value to the constant field (double arrow and arrow keys).
- Accept by Enter.

The figure below shows the panel display when the input I1 selection parameter is in edit mode and the constant field is visible. The constant may have a value from -32768 to 32767. The constant cannot be changed while the adaptive program is running.

Display of the panel



Option 2

- Set the constant to one of the parameters reserved for the constants.
- Connect the constant value to a block as usual by the input selection parameter.

The constants can be changed while the adaptive program is running. They may have values from –8388608…8388607.

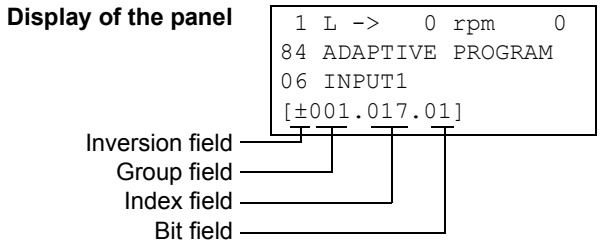
Parameter value as a boolean input

How the block handles the input

- The block reads the selected value as an integer.
- The block uses the bit defined by the bit field as the boolean input.

Bit value 1 is boolean value true and 0 is boolean value false.

Example: The figure below shows the value of input I1 selection parameter when the input is connected to a bit indicating the status of digital input DI2. (In ACS800 Standard Application Program, the digital input states are internally stored as actual signal 1.17 DI6-1 STATUS. Bit 1 corresponds to DI2, bit 0 to DI1.)



How to select the input

See the section *Parameter value as an integer input* above.

Note: The parameter selected as an input should have a packed boolean value (binary data word). See the Firmware Manual.

Constant as a boolean input

How to set and connect the input

- Scroll to the input selection parameter of the block and switch to edit mode (Enter).
- Select C in the inversion field (double arrow and arrow keys). The rest of the line changes to a constant setting field.
- Give the constant. If boolean value true is needed, set the constant to -1. If boolean value false is needed, set to 0.
- Accept by Enter.

String input

How to select the input

String input is needed only with the EVENT block.

For the input selection procedure, see the section *Parameter value as an integer input* above. The bit selection field is not effective.

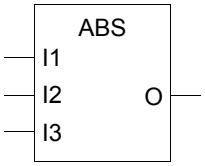
Note: The parameter selected as an input must have a string value. In the ACS800 Standard Application Program, there are parameters in group 85 USER CONSTANTS which can be used for string inputs.

Function blocks

ABS

Type Arithmetic function

Illustration



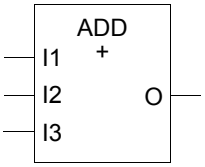
Operation The output is the absolute value of input I1 multiplied by I2 and divided by I3.
$O = |I1| \cdot I2 / I3$

Connections Input I1, I2 and I3: 24 bit integer values (23 bits + sign)
Output (O): 24 bit integer (23 bits + sign)

ADD

Type Arithmetic function

Illustration



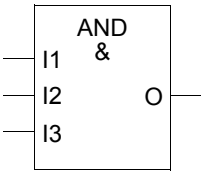
Operation The output is the sum of the inputs.
O = I1 + I2 + I3

Connections Input I1, I2 and I3: 24 bit integer values (23 bits + sign)
Output (O): 24 bit integer (23 bits + sign)

AND

Type Logical function

Illustration



Operation The output is true if all connected inputs are true. Otherwise the output is false.
Truth table:

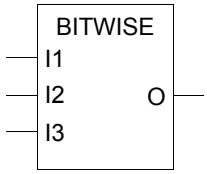
I1	**I2**	**I3**	**O** (binary)	**O** (value on display)
0	0	0	False (All bits 0)	0
0	0	1	False (All bits 0)	0
0	1	0	False (All bits 0)	0
0	1	1	False (All bits 0)	0
1	0	0	False (All bits 0)	0
1	0	1	False (All bits 0)	0
1	1	0	False (All bits 0)	0
1	1	1	True (All bits 1)	-1

Connections Input I1, I2 and I3: Boolean values
Output (O): 24 bit integer value (packed boolean)

BITWISE

Type Arithmetic function

Illustration



Operation The block compares bits of three 24 bit word inputs and forms the output bits as follows:

O = (I1 OR I2) AND I3

Example

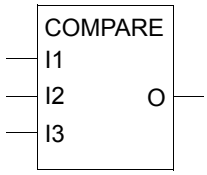
I1	I2	I3	O
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

Connections Input I1: 24 bit integer value (packed boolean)
Input I2: 24 bit integer value (packed boolean)
Input I3: 24 bit integer value (packed boolean)
Output (O): 24 bit integer value (packed boolean)

COMPARE

Type Comparative function

Illustration



Operation

Output bits 0, 1 and 2:

- If I1 > I2, O = … 001 (Output bit 0 is set.)
- If I1 = I2, O = … 010 (Output bit 1 is set.)
- If I1 < I2, O = … 100 (Output bit 2 is set.)

Output bit 3:

- If I1 > I2, O = … 1xxx (Output bit 3 is set and remains set until I1 < I2 - I3, after which bit 3 is reset.)

Output value on display:

bit 0	**bit 1**	**bit 2**	**bit 3**	**O** (value on display)
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
0	0	1	0	4
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
0	0	1	1	12

Connections

Input I1, I2 and I3: 24 bit integer values (23 bits + sign)

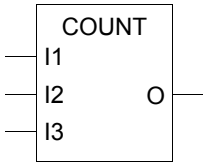
Output (O): 24 bit integer (packed boolean)

COUNT

Type

Counter function

Illustration



Operation

The counter function counts rising edges of the input I1.

The counter is reset by the rising edge of input I2 and limited to the value set with the input I3.

I1: Trigger input

I2: Reset

I3: Max limit for the counter (B0…B19 -> 0…1048575)

O: Value of the counter (B0…B19 -> 0…1048575), and status of the counter (B20). B20 = 1: Counter is at max limit or input I3 is negative.

Connections

Input I1 and I2: Boolean values

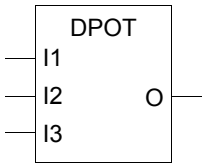
Input I3: 24 bit integer value (20 bits used by the counter)

Output (O): 24 bit integer value (20 bits for counter value and 4 indication bits)

DPOT

Type Counter function

Illustration



Operation The digitally controlled ramp function increments or decrements the output O according to control inputs I1 and I2. The input I1 ramps the output to positive direction and I2 to negative direction. If both inputs are active, nothing happens. The step is defined by the input I3.

Input I1: Count up

Input I2: Count down

Input I3: Ramp step to positive/negative direction (step/sec).

Note: The internal calculation uses 48 bits accuracy to avoid offset errors.

Connections Input I1 and I2: Boolean values

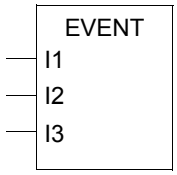
Input I3: 24 bit integer value (23 bits + sign)

Output (O): 24 bit integer value (23 bits + sign)

EVENT

Type Event function

Illustration



Operation Input I1 triggers the event. I2 selects the parameter index from which the event message (text string) is read. I3 selects the type of the event (warning or fault).

I1	I2	I3	Cause
0->1			block activates the event
0			block deactivates the event
	I2		contents of the event message
		0	type of event: warning
		1	type of event: fault
		2	type of event: event

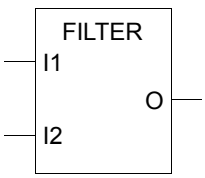
Connections Input I1, I3: 24 bit integer values (23 bits + sign)

Input I2: String (compulsory)

FILTER

Type Filtering function

Illustration



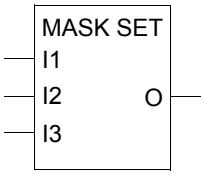
Operation The output is the filtered value of input I1. Input I2 is the filtering time.
$O = I1 \cdot (1 - e^{-t/I2})$
Note: The internal calculation uses 48 bits accuracy to avoid offset errors.

Connections Input I1: 24 bit integer value (23 bits + sign)
Input I2: 24 bit integer value (23 bits + sign). One corresponds to 1 ms.
Output (O): 24 bit integer (23 bits + sign)

MASK-SET

Type Logical function

Illustration



Operation The block function sets or resets the bits defined in I2 in I1.
Input I1: Word input
Input I2: Set word input
Input I3: Set/reset I2 in I1.

Example with SET			
I1	**I2**	**I3**	**O**
0	0	True	0
1	0	True	1
1	1	True	1
0	1	True	1

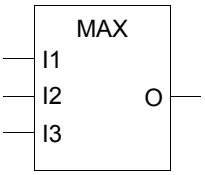
Example with RESET			
I1	**I2**	**I3**	**O**
0	0	False	0
1	0	False	1
1	1	False	0
0	1	False	0

Connections Input I1: 24 bit integer value (packed boolean)
Input I2: 24 bit integer value (packed boolean)
Input I3: Boolean
Output (O): 24 bit integer value (packed boolean)

MAX

Type Comparative function

Illustration



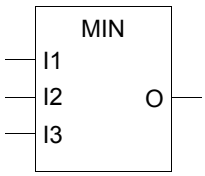
Operation The output is the highest input value.
O = MAX (I1, I2, I3)

Connections Input I1, I2 and I3: 24 bit integer values (23 bits + sign)
Output (O): 24 bit integer (23 bits + sign)

MIN

Type Comparative function

Illustration



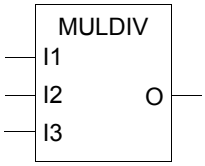
Operation The output is the lowest input value.
O = MIN (I1, I2, I3)

Connections Input I1, I2 and I3: 24 bit integer values (23 bits + sign)
Output (O): 24 bit integer (23 bits + sign)

MULDIV

Type Arithmetic function

Illustration



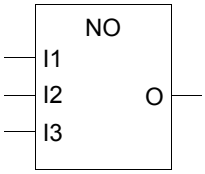
Operation The output is the product of input I1 and input I2 divided by input I3.
O = (I1 · I2) / I3

Connections Input I1, I2 and I3: 24 bit integer values (23 bits + sign)
Output (O): 24 bit integer (23 bits + sign)

NO

Type -

Illustration



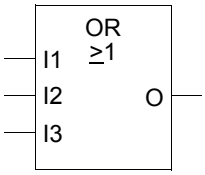
Operation The block does not do anything.

Connections -

OR

Type Logical function

Illustration



Operation The output is true if any of the inputs is true. Truth table:

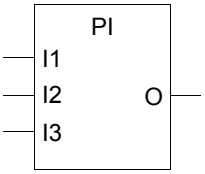
I1	**I2**	**I3**	**O** (binary)	**O** (value on display)
0	0	0	False (All bits 0)	0
0	0	1	True (All bits 1)	-1
0	1	0	True (All bits 1)	-1
0	1	1	True (All bits 1)	-1
1	0	0	True (All bits 1)	-1
1	1	0	True (All bits 1)	-1
1	1	1	True (All bits 1)	-1

Connections Input I1, I2 and I3: Boolean values
Output (O): 24 bit integer value (packed boolean)

PI

Type PI controller

Illustration



Operation The output is input I1 multiplied by I2/100 plus integrated I1 multiplied by I3/100.
$O = I1 \cdot I2/100 + (I3/100) \cdot \int I1$
Note: The internal calculation uses 48 bits accuracy to avoid offset errors.

Connections Inputs I1: 24 bit integer value (23 bit + sign)
Input I2:
- 24 bit integer value (23 bit + sign)
- Gain factor. 100 corresponds to 1. 10 000 corresponds to 100.

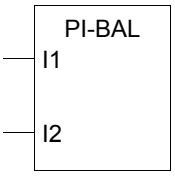
Input I3:
- Integrator coefficient. 100 corresponds to 1. 10 000 corresponds to 100.

Output (O): 24 bit integer (23 bits + sign). The range is limited to 0…10000.

PI-BAL

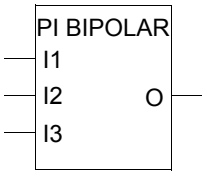
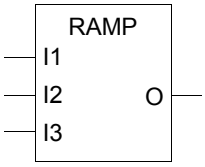
Type Initialisation block for the PI controller

Illustration



Operation The block initialises the PI block first. When input I1 becomes true, the block writes the value of I2 to the output of the PI block. When I1 becomes false, the block releases the output of the PI controller block which continues normal operation from the set output.
Note: The block may be used only with the PI block. The block must follow the PI block.

Connections Input I1: Boolean value
Input I2: 24 bit integer value (23 bits + sign)

PI-BIPOLAR	**Type**	PI controller
	Illustration	
	Operation	See the PI block. Except Output (O) range: -10000…10000.
RAMP	**Type**	Ramp function
	Illustration	

Operation

The block uses input I1 as a reference value. The step values (inputs I2 and I3) increase or decrease the output O as long as the output differs from limit I1. When O = I1, the output remains steady.

Input I1: Reference value

Input I2: Step to positive direction (step/sec). Increase the output, when O < I1.

Input I3: Step to negative direction (step/sec). Decrease the output, when O > I1.

$O_n = O_{n-1} + I2$ when I1 > O

$O_n = O_{n-1} - I3$ when I1 < O

$O_n = I1$ when I1 = O

Example:

Input I1: 0 -> 150 -> -100 ->0

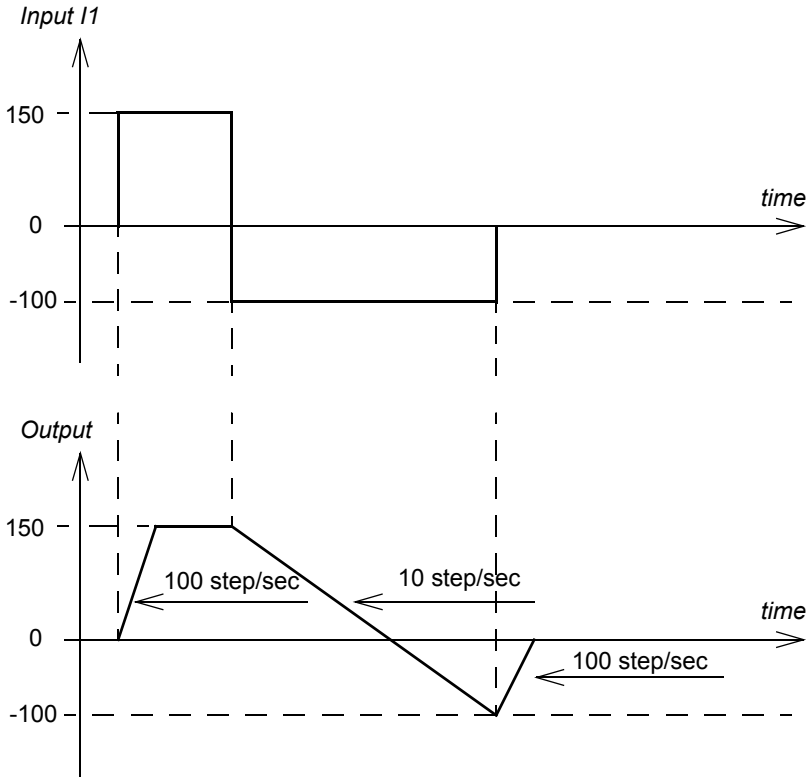
Input I2: 100 step/sec

Input I3: 10 step/sec

Output:

Going up: Ramp step from Input I2

Going down: Ramp step from Input I3



Connections

Input I1, I2 and I3: 24 bit integer values (23 bits + sign)

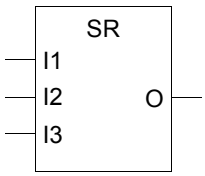
Output (O): 24 bit integer (23 bits + sign)

Note: The internal calculation uses 48 bits accuracy to avoid offset errors.

SR

Type Logical function

Illustration



Operation Set/reset block. Input I1 sets and I2 and I3 reset the output.

- If I1, I2 and I3 are false, the current value remains at the output.
- If I1 is true and I2 and I3 are false, the output is true.
- If I2 or I3 is true, the output is false.

I1	**I2**	**I3**	**O** (binary)	**O** (value on display)
0	0	0	Output	Output
0	0	1	False (All bits 0)	0
0	1	0	False (All bits 0)	0
0	1	1	False (All bits 0)	0
1	0	0	True (All bits 1)	-1
1	0	1	False (All bits 0)	0
1	1	0	False (All bits 0)	0
1	1	1	False (All bits 0)	0

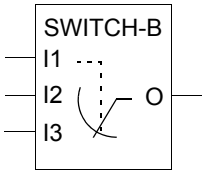
Connections Input I1, I2 and I3: Boolean values

Output (O): 24 bit integer value (23 bits + sign)

SWITCH-B

Type Logical function

Illustration



Operation The output is equal to input I2 if input I1 is true and equal to input I3 if input I1 is false.

I1	**I2**	**I3**	**O**	**O** (value on display)
0	I2	I3	I3	True = -1
1	I2	I3	I2	False = 0

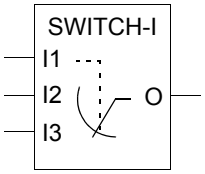
Connections Input I1, I2 and I3: Boolean values

Output (O): 24 bit integer value (packed boolean)

SWITCH-I

Type Logical function

Illustration



Operation The output is equal to input I2 if input I1 is true and equal to input I3 if input I1 is false.

I1	I2	I3	O
0	I2	I3	I3
1	I2	I3	I2

Connections

Input I1: Boolean value

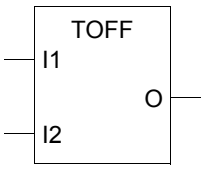
Input I2 and I3: 24 bit integer values (23 bits + sign)

Output (O): 24 bit integer value (23 bits + sign)

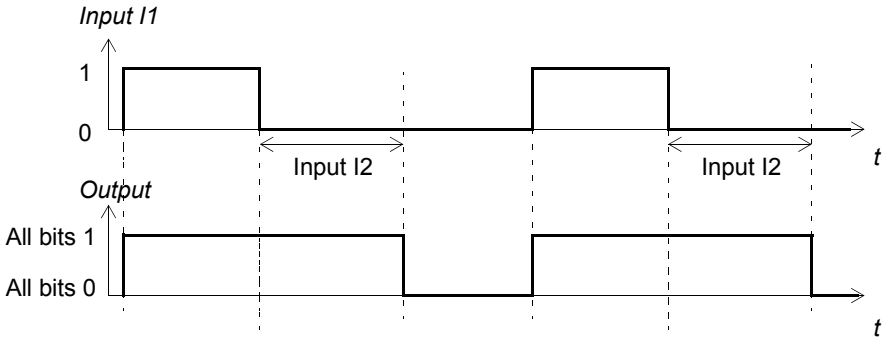
TOFF

Type Timing function

Illustration



Operation The output is true when input I1 is true. The output is false when input I1 has been false for a time equal or longer than input I2.



Values on display: True = -1, false = 0.

Connections

Input I1: Boolean value

Input I2: 24 bit integer value (23 bits + sign). One corresponds to 1 ms.

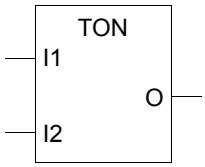
Output (O):

- 24 bit integer value (packed boolean)

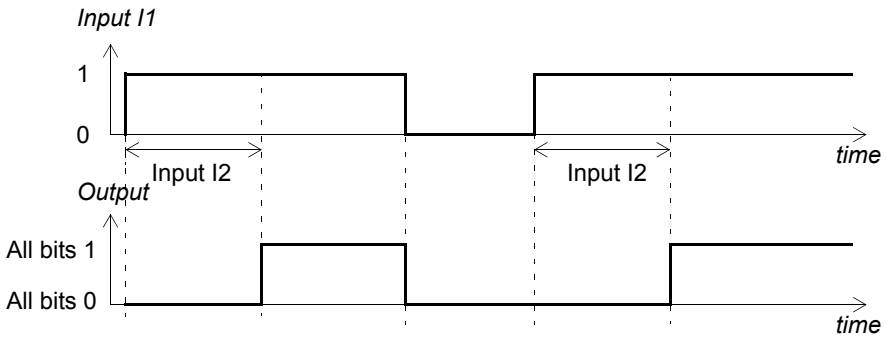
TON

Type Timing function

Illustration



Operation The output is true when input I1 has been true for a time equal or longer than input I2. The output is false when the input is false.



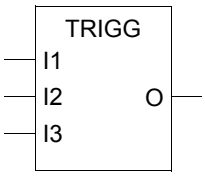
Values on display: True = -1, false = 0.

Connections Input I1: Boolean value
Input I2: 24 bit integer value (23 bits + sign). 1 corresponds to 1 ms.
Output (O): 24 bit integer value (packed boolean)

TRIGG

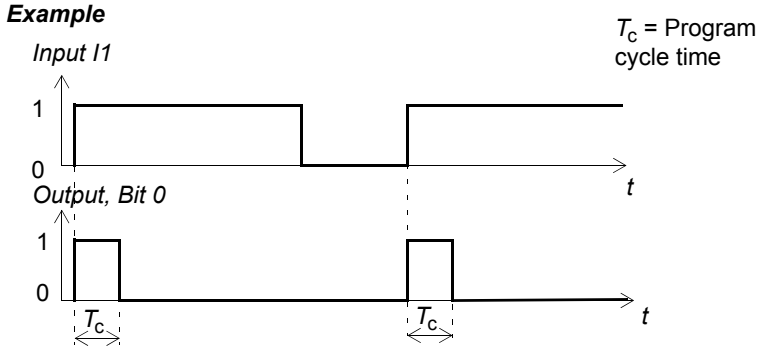
Type Timing function

Illustration



Operation The rising edge of input I1 sets the output bit 0 for one program cycle.
The rising edge of input I2 sets the output bit 1 for one program cycle.
The rising edge of input I3 sets the output bit 2 for one program cycle.

Example

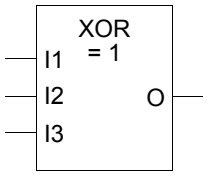


Connections Input I1, I2 and I3: Boolean values
Output (O):
- 24 bit integer value (23 bits + sign)

XOR

Type Logical function

Illustration



Operation The output is true if one input is true, otherwise the output is false. Truth table:

I1	**I2**	**I3**	**O** (binary)	**O** (value on display)
0	0	0	False (All bits 0)	0
0	0	1	True (All bits 1)	-1
0	1	0	True (All bits 1)	-1
0	1	1	False (All bits 0)	0
1	0	0	True (All bits 1)	-1
1	0	1	False (All bits 0)	0
1	1	0	False (All bits 0)	0
1	1	1	True (All bits 1)	-1

Connections Input I1, I2 and I3: Boolean values
Output (O):
- 24 bit integer value (23 bits + sign)

Actual signals and parameters in ACS800 Standard Application Program

Chapter overview

The chapter lists the actual signals, parameters and parameter values of ACS800 Application Program that are essential for the Adaptive Programming.

Actual signals

The table below lists the actual signals that are essential for the Adaptive Program. The abbreviation FbEq stands the fieldbus equivalent.

Index	Signal Name/Value	Description	FbEq.
09	ACTUAL SIGNALS	Signals for the Adaptive Program	
09.01	AI1 SCALED	Value of analogue input AI1 scaled to an integer value.	20000 = 10 V
09.02	AI2 SCALED	Value of analogue input AI2 scaled to an integer value.	20000 = 20 mA
09.03	AI3 SCALED	Value of analogue input AI3 scaled to an integer value.	20000 = 20 mA
09.04	AI5 SCALED	Value of analogue input AI5 scaled to an integer value.	20000 = 20 mA
09.05	AI6 SCALED	Value of analogue input AI6 scaled to an integer value.	20000 = 20 mA
09.06	MASTER CW	Control Word (CW) of the Main Reference Dataset received from the master station through the fieldbus interface.	-32768 …32767
09.07	MASTER REF1	Reference 1 (REF1) of the Main Reference Dataset received from the master station through the fieldbus interface	-32768 … 32767
09.08	MASTER REF2	Reference 2 (REF2) of the Main Reference Dataset received from the master station through the fieldbus interface.	-32768 … 32767
09.09	AUX DS VAL1	Reference 3 (REF3) of the Auxiliary Reference Dataset received from the master station through the fieldbus interface.	-32768 … 32767
09.10	AUX DS VAL2	Reference 4 (REF4) of the Auxiliary Reference Dataset received from the master station through the fieldbus interface.	-32768 … 32767
09.11	AUX DS VAL3	Reference 5 (REF5) of the Auxiliary Reference Dataset received from the master station through the fieldbus interface.	-32768 … 32767

Parameters

The table below lists the parameters and parameters values that are essential for the Adaptive Program. The abbreviation FbEq stands for fieldbus equivalent.

Index	Parameter name / value	Description	FbEq
10	START/STOP/DIR	Parameters through which the Adaptive Program can control the start, stop and direction of the drive.	
10.01	EXT1 STRT/STP/DIR		
	PARAM 10.04	Source selected by 10.04	17
10.02	EXT2 STRT/STP/DIR		
	PARAM 10.05	Source selected by 10.05.	17
10.04	EXT 1 STRT PTR	Selects the source for parameter 10.01.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value: - Parameter pointer: Inversion, group, index and bit fields. The bit number is effective only for blocks handling boolean inputs. - Constant value: Inversion and constant fields. The inversion field must have value C to enable the setting of the constant.	
10.05	EXT 2 STRT PTR	Selects the source for 10.02.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
11	REFERENCE SELECT	Parameters through which the Adaptive Program can control the reference of the drive.	
11.02	EXT1/EXT2 SELECT		
	PARAM 11.09	Source selected by 11.09.	16
11.03	EXT REF1 SELECT		
	PARAM 11.10	Source selected by 11.10.	37
11.06	EXT REF2 SELECT		
	PARAM 11.11	Source selected by 11.11.	38
11.09	EXT 1/2 SEL PTR	Selects the source for 11.02.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
11.10	EXT 1 REF PTR	Selects the source for 11.03.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
11.11	EXT 2 REF PTR	Selects the source for 11.06.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14	RELAY OUTPUTS	Parameters through which the Adaptive Program can control the relay outputs of the drive.	
14.01	RELAY RO1 OUTPUT		
	PARAM 14.16	Source selected by parameter 14.16.	36
14.02	RELAY RO2 OUTPUT		

Index	Parameter name / value	Description	FbEq
	PARAM 14.17	Source selected by parameter 14.17.	36
14.03	RELAY RO3 OUTPUT		
	PARAM 14.18	Source selected by parameter 14.18.	36
14.10	DIO MOD1 RO1		
	PARAM 14.19	Source selected by parameter 14.19.	7
14.11	DIO MOD1 RO2		
	PARAM 14.20	Source selected by parameter 14.20.	7
14.12	DIO MOD2 RO1		
	PARAM 14.21	Source selected by parameter 14.21.	7
14.13	DIO MOD2 RO2		
	PARAM 14.22	Source selected by parameter 14.22.	7
14.14	DIO MOD3 RO1		
	PARAM 14.23	Source selected by parameter 14.23.	7
14.15	DIO MOD3 RO2		
	PARAM 14.24	Source selected by parameter 14.24.	7
14.16	RO PTR1	Selects the source for parameter 14.01.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.17	RO PTR2	Selects the source for parameter 14.02.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.18	RO PTR3	Selects the source for parameter 14.03.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.19	RO PTR4	Selects the source for parameter 14.10.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.20	RO PTR5	Selects the source for parameter 14.11.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.21	RO PTR6	Selects the source for parameter 14.12.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.22	RO PTR7	Selects the source for parameter 14.13.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.23	RO PTR8	Selects the source for parameter 14.14.	

Index	Parameter name / value	Description	FbEq
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
14.24	RO PTR9	Selects the source for parameter 14.15.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
15	ANALOGUE OUTPUTS	Parameters through which the Adaptive Program can control the standard analogue outputs of the drive.	
15.01	ANALOGUE OUTPUT1		
	PARAM 15.11	Source selected by 15.11	17
15.06	ANALOGUE OUTPUT2		
	PARAM 15.12	Source selected by 15.12	16
15.11	AO1 PTR	Selects the source for parameter 15.01.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
15.12	AO2 PTR	Selects the source for parameter 15.06.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
16	SYSTEM CTRL INPUTS	Parameters through which the Adaptive Program can control the system control inputs of the drive.	
16.01	RUN ENABLE		
	PARAM 16.08	Source selected by parameter 16.08.	15
16.08	RUN ENA PTR	Selects the source for parameter 16.01	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
20	LIMITS	Parameters through which the Adaptive Program can control the operation limits of the drive.	
20.13	MIN TORQ SEL	Selects the torque minimum limit	
	PARAM 20.18	Limit given by 20.18	20
20.14	MAX TORQ SEL	Selects the torque maximum limit	
	PARAM 20.19	Limit given by 20.19	19
20.18	TORQ MIN PTR	Selects the source for 20.13	100 = 1%
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
20.19	TORQ MAX PTR	Selects the source for 20.14	100 = 1%
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
22	ACCEL/DECEL	Parameters through which the Adaptive Program can control the acceleration and deceleration of the drive.	
22.01	ACC/DEC 1/2 SEL		

Index	Parameter name / value	Description	FbEq
	PAR 22.08&09	Acceleration and deceleration times given by parameters 22.08 and 22.09	15
22.08	ACC PTR	Selects the source for 22.01	100 = 1 s
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
22.09	DEC PTR	Selects the source for 22.01	100 = 1 s
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
26	MOTOR CONTROL	Parameters through which the Adaptive Program can control the flux of the drive.	
26.06	FLUX REF PTR	Selects the source for the flux reference.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
40	PID CONTROL	Parameters through which the Adaptive Program can affect on the process PID control.	
40.07	ACTUAL1 INPUT SEL		
	PARAM 40.25	Source selected by parameter 40.25.	6
40.25	ACTUAL1 PTR	Selects the source for 40.07	100 = 1%
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
83	ADAPT PROG CTRL	Execution control of the Adaptive Program.	
83.01	ADAPT PROG CMD	Selects the operation mode for the Adaptive Program.	
	STOP	Stop. The program cannot be edited.	
	START	Run. The program cannot be edited.	
	EDIT	Stop to edit mode. The program can be edited.	
83.02	EDIT COMMAND	Selects the command for the block placed in the location defined by parameter 83.03. The program must be in edit mode. (See parameter 83.01.)	
	NO	Home value. The value automatically restores to NO after an editing command has been executed.	
	PUSH	Shifts the block in the location defined by parameter 83.03 and the subsequent blocks one location up. A new block can be placed in the emptied location by programming the Block Parameter Set as usual. **Example:** A new block needs to be placed in between the current block number four (parameters 84.20…84.25) and five (parameters 84.25…84.29). In order to do this: - Shift the program to the edit mode by parameter 83.01. - Select location number five as the desired location for the new block by parameter 83.03. - Shift the block in location number 5 and all subsequent blocks one location forward by parameter 83.02 (selection PUSH). - Program the emptied location number 5 by parameters 84.25 to 84.29 as usual.	
	DELETE	Deletes the block in the location defined by parameter 83.03 and shifts the subsequent blocks one step down.	

<table>
<tr><th>Index</th><th>Parameter name / value</th><th>Description</th><th>FbEq</th></tr>
<tr><td>83.03</td><td>EDIT BLOCK</td><td>Defines the block location number for the command selected by parameter 83.02.</td><td></td></tr>
<tr><td></td><td>1 … 15</td><td>Block location number.</td><td></td></tr>
<tr><td>83.04</td><td>TIMELEVEL SEL</td><td>Selects the execution cycle time for the Adaptive Program. The setting is valid for all blocks.</td><td></td></tr>
<tr><td></td><td>12 ms</td><td>12 milliseconds</td><td></td></tr>
<tr><td></td><td>100 ms</td><td>100 milliseconds</td><td></td></tr>
<tr><td></td><td>1000 ms</td><td>1000 milliseconds</td><td></td></tr>
<tr><td>84</td><td>ADAPTIVE PROGRAM</td><td>Creation and diagnostics of the Adaptive Program.</td><td></td></tr>
<tr><td>84.01</td><td>STATUS</td><td>Shows the value of the Adaptive Program status word. The table below shows the alternative bit states and the corresponding values on the panel display.
<table>
<tr><th>Bit</th><th>Display</th><th>Meaning</th></tr>
<tr><td>0</td><td>1</td><td>Stopped</td></tr>
<tr><td>1</td><td>2</td><td>Running</td></tr>
<tr><td>2</td><td>4</td><td>Faulted</td></tr>
<tr><td>3</td><td>8</td><td>Editing</td></tr>
<tr><td>4</td><td>10</td><td>Checking</td></tr>
<tr><td>5</td><td>20</td><td>Pushing</td></tr>
<tr><td>6</td><td>40</td><td>Popping</td></tr>
<tr><td>8</td><td>100</td><td>Initialising</td></tr>
</table></td><td></td></tr>
<tr><td>84.02</td><td>FAULTED PAR</td><td>Points out the faulted parameter in the Adaptive Program.</td><td></td></tr>
<tr><td>84.05</td><td>BLOCK1</td><td>Selects the function block for Block Parameter Set 1.</td><td></td></tr>
<tr><td></td><td>ABS</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>ADD</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>AND</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>COMPARE</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>EVENT</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>FILTER</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>MAX</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>MIN</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>MULDIV</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>NO</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>OR</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>PI</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>PI-BAL</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>SR</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>SWITCH-B</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>SWITCH-I</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>TOFF</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>TON</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>TRIGG</td><td>See the chapter Function blocks.</td><td></td></tr>
<tr><td></td><td>XOR</td><td>See the chapter Function blocks.</td><td></td></tr>
</table>

Index	Parameter name / value	Description	FbEq
84.06	INPUT1	Selects the source for input I1 of Block Parameter Set 1 (BPS1).	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value: - Parameter pointer: Inversion, group, index and bit fields. The bit number is effective only for blocks handling boolean inputs. - Constant value: Inversion and constant fields. The inversion field must have value C to enable the setting of the constant. **Example:** The state of digital input DI2 is connected to Input 1 as follows: - Set the source selection parameter (84.06) to +.01.17.01. (The application program stores the state of digital input DI2 to bit 1 of actual signal 01.17.) - Invert the value by switching the sign of the pointer value (-01.17.01.).	
84.07	INPUT2	See parameter 84.06.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	See parameter 84.06.	
84.08	INPUT3	See parameter 84.06.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	See parameter 84.06.	
84.09	OUTPUT	Stores and displays the output of Block Parameter Set 1.	
…	…	Stores and displays the output of Block Parameter Set 15.	
84.79	OUTPUT	Stores the output of Block Parameter Set 15. See parameter 84.09.	
85	USER CONSTANTS	Storage of the Adaptive Program constants and messages.	
85.01	CONSTANT1	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.02	CONSTANT2	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.03	CONSTANT3	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.04	CONSTANT4	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.05	CONSTANT5	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.06	CONSTANT6	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.07	CONSTANT7	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.08	CONSTANT8	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.09	CONSTANT9	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.10	CONSTANT10	Sets a constant for the Adaptive Program.	
	-8388608 to 8388607	Integer value.	
85.11	STRING1	Stores a message to be used in the Adaptive Program (EVENT block).	

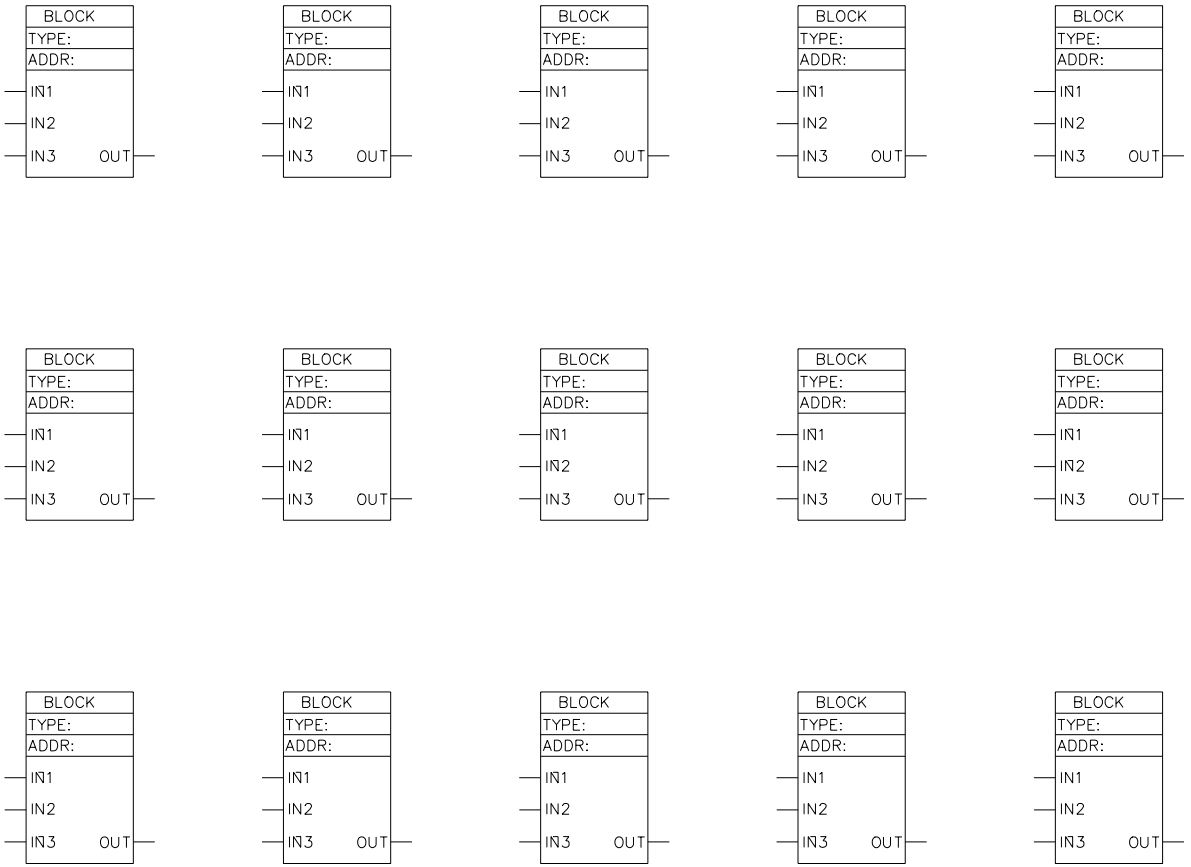
Index	Parameter name / value	Description	FbEq
	MESSAGE1	Message	
85.12	STRING2	Stores a message to be used in the Adaptive Program (EVENT block).	
	MESSAGE2	Message	
85.13	STRING3	Stores a message to be used in the Adaptive Program (EVENT block).	
	MESSAGE3	Message	
85.14	STRING4	Stores a message to be used in the Adaptive Program (EVENT block).	
	MESSAGE4	Message	
85.15	STRING5	Stores a message to be used in the Adaptive Program (EVENT block).	
	MESSAGE5	Message	
96	EXTERNAL AO	Parameters through which the Adaptive Program can control the optional analogue outputs of the drive.	
96.01	EXT AO1		
	PARAM 96.11	Source selected by parameter 96.11.	16
96.06	EXT AO2		
	PARAM 96.12	Source selected by parameter 96.11.	16
96.11	EXT AO1 PTR	Selects the source for 96.01.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	
96.12	EXT AO2 PTR	Selects the source for 96.06.	
	-255.255.31 … +255.255.31 / C.-32768 … C.32767	Parameter pointer or constant value. See Parameter 10.04.	

Customer diagrams

Chapter overview

This chapter includes three blank block diagram sheets on which the Adaptive Program can be documented.

85.01 Constant1=
85.02 Constant2=
85.03 Constant3=
85.04 Constant4=
85.05 Constant5=
85.06 Constant6=
85.07 Constant7=
85.08 Constant8=
85.09 Constant9=
85.10 Constant10=
85.11 String1=
85.12 String2=
85.13 String3=
85.14 String4=
85.15 String5=



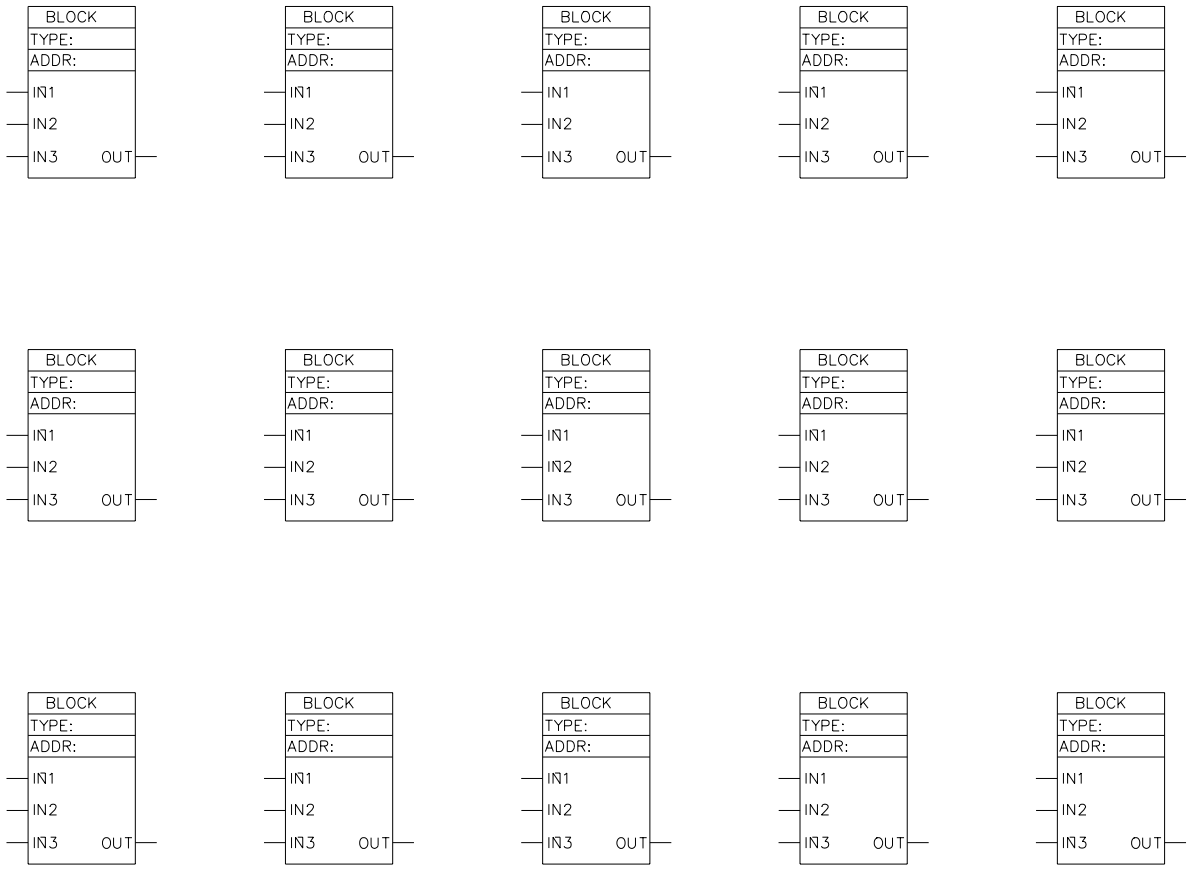
EXT1 STRT/STP/DIR	10.01
EXT1 START PTR	10.04
EXT2 STRT/STP/DIR	10.02
EXT2 START PTR	10.05
EXT1 REF SELECT	11.03
EXT1 REF PTR	11.10
EXT2 REF SELECT	11.06
EXT2 REF PTR	11.11
EXT1/EXT2 SELECT	11.02
EXT 1/2 SEL PTR	11.09
RO1	14.01
RO1 PTR	14.16
RO2	14.02
RO2 PTR	14.17
RO3	14.03
RO3 PTR	14.18
RO4	14.10
RO4 PTR	14.19
RO5	14.11
RO5 PTR	14.20
RO6	14.12
RO6 PTR	14.21
RO7	14.13
RO7 PTR	14.22
RO8	14.14
RO8 PTR	14.23
RO9	14.15
RO9 PTR	14.24
AO1	15.01
AO1 PTR	15.11
AO2	15.06
AO2 PTR	15.12
RUN ENABLE	16.01
RUN ENABLE PTR	16.08
MIN TORQ SEL	20.13
TORQ MIN PTR	20.18
MAX TORQ SEL	20.14
TORQ MAX PTR	20.19
ACC/DEC	22.01
ACC PTR	22.08
DEC PTR	22.09
FLUX REF SEL	26.06
FLUX REF PTR	26.08
ACTUAL1 INPUT SEL	40.07
ACTUAL 1 PTR	40.25
EXT AO1	96.01
EXT AO1 PTR	96.11
EXT AO2	96.02
EXT AO2 PTR	96.12

83.04	TIME LEVEL	ms
33.01	SOFTWARE VERSION	

ABS ADD AND COMPARE EVENT FILTER MAX MIN MULDIV NO OR PI PI-BAL SR SWITCH-B SWITCH-I TOFF TON TRIGG XOR

Based on	Prepared	Title	Doc. des.
Customer	Approved		
	Project name		Resp. dept.
Cust. Doc. No.		ABB Industry Oy	Doc. No.
Date			

85.01 Constant1=
85.02 Constant2=
85.03 Constant3=
85.04 Constant4=
85.05 Constant5=
85.06 Constant6=
85.07 Constant7=
85.08 Constant8=
85.09 Constant9=
85.10 Constant10=
85.11 String1=
85.12 String2=
85.13 String3=
85.14 String4=
85.15 String5=



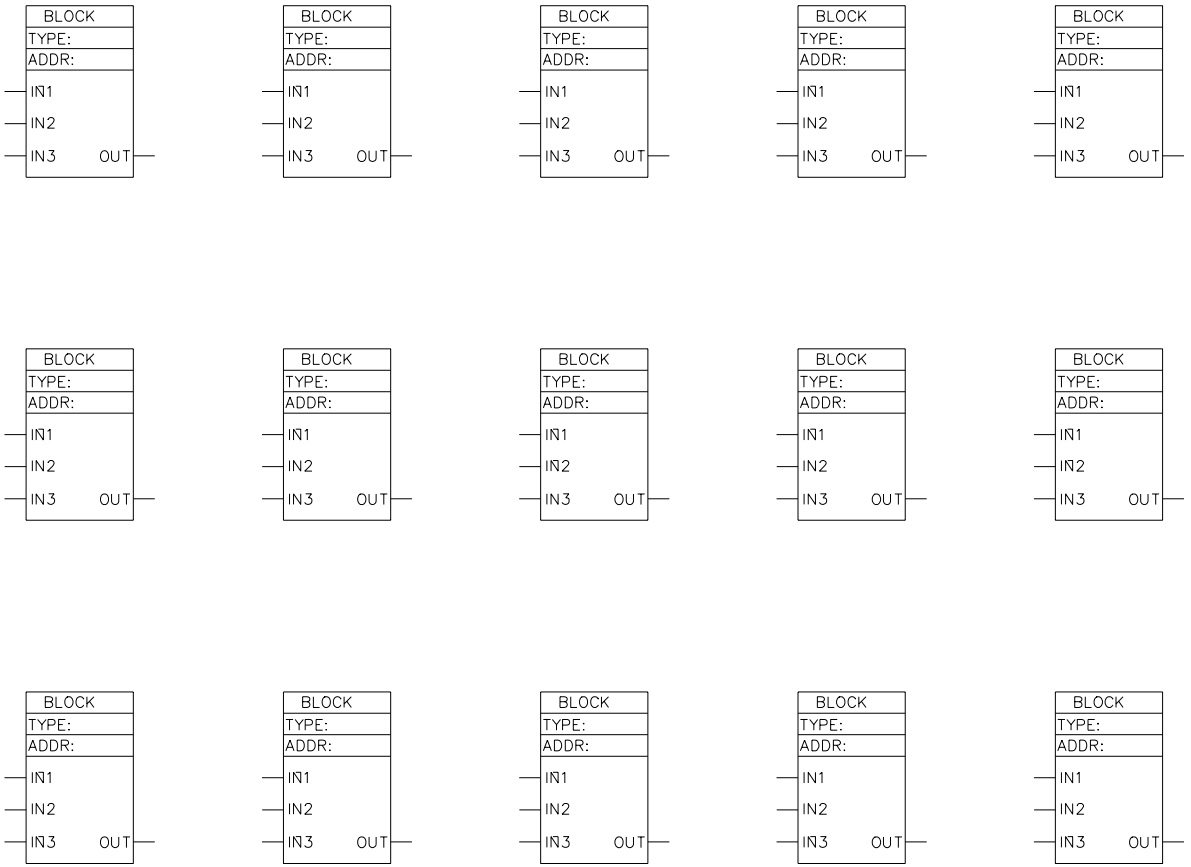
EXT1 STRT/STP/DIR	10.01
EXT1 START PTR	10.04
EXT2 STRT/STP/DIR	10.02
EXT2 START PTR	10.05
EXT1 REF SELECT	11.03
EXT1 REF PTR	11.10
EXT2 REF SELECT	11.06
EXT2 REF PTR	11.11
EXT1/EXT2 SELECT	11.02
EXT 1/2 SEL PTR	11.09
RO1	14.01
RO1 PTR	14.16
RO2	14.02
RO2 PTR	14.17
RO3	14.03
RO3 PTR	14.18
RO4	14.10
RO4 PTR	14.19
RO5	14.11
RO5 PTR	14.20
RO6	14.12
RO6 PTR	14.21
RO7	14.13
RO7 PTR	14.22
RO8	14.14
RO8 PTR	14.23
RO9	14.15
RO9 PTR	14.24
AO1	15.01
AO1 PTR	15.11
AO2	15.06
AO2 PTR	15.12
RUN ENABLE	16.01
RUN ENABLE PTR	16.08
MIN TORQ SEL	20.13
TORQ MIN PTR	20.18
MAX TORQ SEL	20.14
TORQ MAX PTR	20.19
ACC/DEC	22.01
ACC PTR	22.08
DEC PTR	22.09
FLUX REF SEL	26.06
FLUX REF PTR	26.08
ACTUAL1 INPUT SEL	40.07
ACTUAL 1 PTR	40.25
EXT AO1	96.01
EXT AO1 PTR	96.11
EXT AO2	96.02
EXT AO2 PTR	96.12

83.04	TIME LEVEL	ms
33.01	SOFTWARE VERSION	

ABS ADD AND COMPARE EVENT FILTER MAX MIN MULDIV NO OR PI PI-BAL SR SWITCH-B SWITCH-I TOFF TON TRIGG XOR

Based on	Prepared	Title	Doc. des.
Customer	Approved		
	Project name		Resp. dept.
Cust. Doc. No.		ABB Industry Oy	Doc. No.
Date			

85.01 Constant1=
85.02 Constant2=
85.03 Constant3=
85.04 Constant4=
85.05 Constant5=
85.06 Constant6=
85.07 Constant7=
85.08 Constant8=
85.09 Constant9=
85.10 Constant10=
85.11 String1=
85.12 String2=
85.13 String3=
85.14 String4=
85.15 String5=



EXT1 STRT/STP/DIR	10.01
EXT1 START PTR	10.04
EXT2 STRT/STP/DIR	10.02
EXT2 START PTR	10.05
EXT1 REF SELECT	11.03
EXT1 REF PTR	11.10
EXT2 REF SELECT	11.06
EXT2 REF PTR	11.11
EXT1/EXT2 SELECT	11.02
EXT 1/2 SEL PTR	11.09
RO1	14.01
RO1 PTR	14.16
RO2	14.02
RO2 PTR	14.17
RO3	14.03
RO3 PTR	14.18
RO4	14.10
RO4 PTR	14.19
RO5	14.11
RO5 PTR	14.20
RO6	14.12
RO6 PTR	14.21
RO7	14.13
RO7 PTR	14.22
RO8	14.14
RO8 PTR	14.23
RO9	14.15
RO9 PTR	14.24
AO1	15.01
AO1 PTR	15.11
AO2	15.06
AO2 PTR	15.12
RUN ENABLE	16.01
RUN ENABLE PTR	16.08
MIN TORQ SEL	20.13
TORQ MIN PTR	20.18
MAX TORQ SEL	20.14
TORQ MAX PTR	20.19
ACC/DEC	22.01
ACC PTR	22.08
DEC PTR	22.09
FLUX REF SEL	26.06
FLUX REF PTR	26.08
ACTUAL1 INPUT SEL	40.07
ACTUAL 1 PTR	40.25
EXT AO1	96.01
EXT AO1 PTR	96.11
EXT AO2	96.02
EXT AO2 PTR	96.12

83.04	TIME LEVEL	ms
33.01	SOFTWARE VERSION	

ABS ADD AND COMPARE EVENT FILTER MAX MIN MULDIV NO OR PI PI-BAL SR SWITCH-B SWITCH-I TOFF TON TRIGG XOR

Based on	Prepared	Title	Doc. des.
Customer	Approved		
	Project name		Resp. dept.
Cust. Doc. No.		ABB Industry Oy	Doc. No.
Date			

3AFE 64527274 Rev C / EN
EFFECTIVE: 08.04.2005

ABB Oy
AC Drives
P.O.Box 184
FI-00381 Helsinki
FINLAND
Telephone: +358 10 22 11
Fax: +358 10 22 22681
Internet: http://www.abb.com

ABB Inc.
Automation Technologies
Drives & Motors
16250 West Glendale Drive
New Berlin, WI 53151
USA
Telephone: 262 785-8378
800-HELP-365
Fax: 262 780-5135