AKD BASIC™
User Guide

Edition March 30, 2012, Revision A

Valid for Hardware Revision C

Patents Pending

Part Number 903-2000014-00

c UL us LISTED IND. CONT. EQ. 86F2

Keep all manuals as a product component during the life span of the product.
Pass all manuals to future users/owners of the product.

Record of Document Revisions:

Revision	Remarks
06/2011, Rev A	Launch Version for M_01-06-00-000

Current patents:

US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)

US Patent 5,162,798 (used in control card R/D)

US Patent 6,118,241 (used in control card simple dynamic braking)

Technical changes which improve the performance of the device may be made without prior notice.

Printed in the United States of America

This page intentionally left blank.

Table of Contents

1 About this User Guide

This guide describes the operation and use of the AKD drive. Each section details a specific topic related to the use of the product in simple terms which will help you get the most from the product. Each section includes examples to help guide you in setting up and using the various features available in the drive.

This guide is for users who have installed and tested the drive according to the *AKD Installation Manual*. The *AKD Installation Manual* is included on the product CD and contains critical safety information.

2 Program View

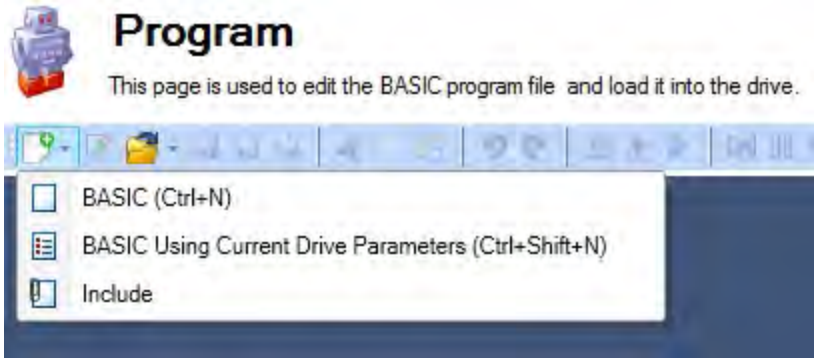
The program view in AKD WorkBench is specific to AKD BASIC drives and can only be accessed using an AKD BASIC drive type in either online or offline mode. Programs can be compiled in offline mode, but an AKD BASIC drive must be connected to download and run a program.

2.1 Toolbar Options

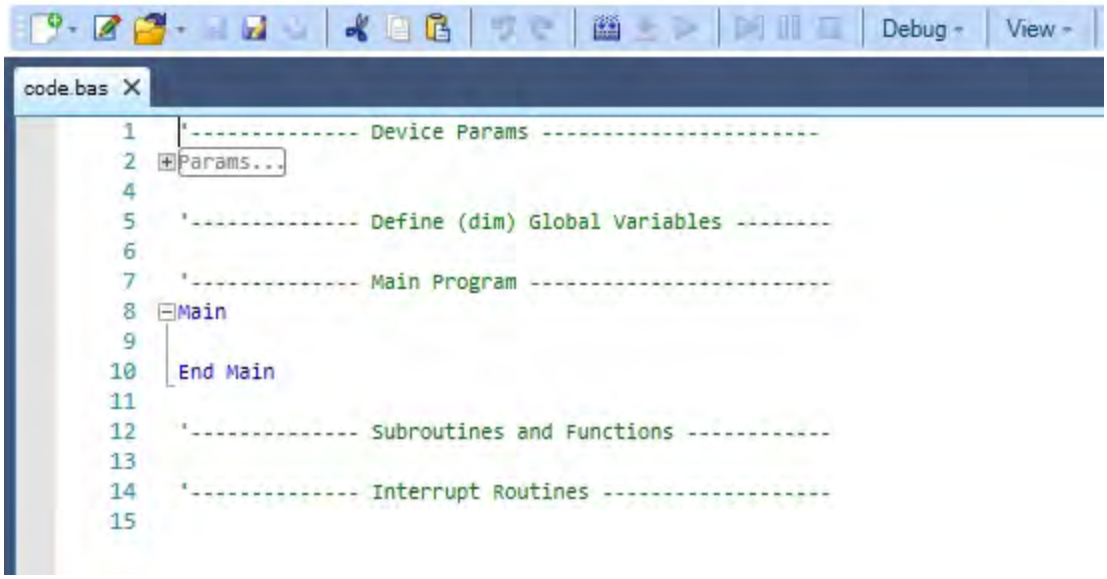
The toolbar at the top of the program view contains a number of tools to assist you in programming.

2.1.1 New ...

The Program view loads as an empty window. Create a new program by clicking the new icon in the upper-left corner and selecting either **BASIC** or **BASIC Using Current Drive Params**. The **BASIC** selection will load an empty template. The **BASIC Using Current Drive Params** selection will populate the template with all current drive values. Then select the location and name of your new project file, and click save.



This will load the BASIC template from which you can begin programming.



2.1.2 Save / Save As

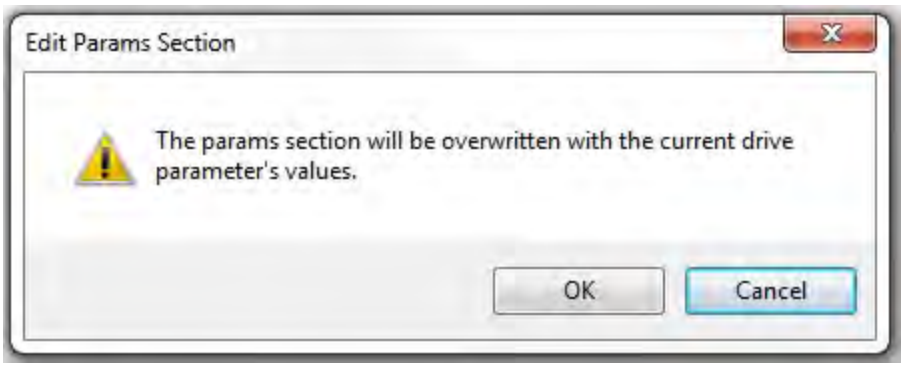
Once you have written your program, save it using the "Save" or "Save As" function on the toolbar .

2.1.3 Open

To open a previous project file click the "Open" function and locate your file in the explorer.

2.1.4 Edit Parameters Section

Clicking on the Edit Params Section tool will prompt you to populate the Params section of your code with all the current drive settings. Clicking yes will overwrite all current param defintions with the current drive values.



2.1.5 Compile / Download / Run

The Compile button will compile your source code to binary so that you may download the program to the drive.

The Download button will download your compiled code and source code to the drive by default. If you do not wish to download the source code to the drive, go to the options menu, choose download and de-select download source with binary.

The run button will issue a VM.START to start the BASIC program in the drive. If the active BASIC program in the editor is different than the program in the drive, the run button will compile and download the program before running.

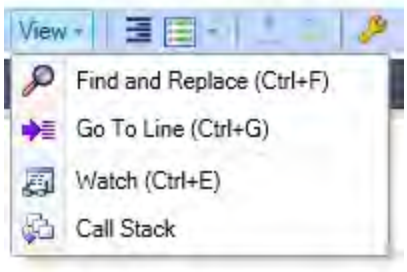
2.1.6 Continue / Pause / Stop

The Continue, Pause, and Stop options allow you to control your program while it is running.

2.1.7 Debug

The Debugger allows you to set break points and step through your program.

2.1.8 View



Use Find and Replace to find keywords in your program and replace them with a new keyword

Use the Go To Line to go to a specific line of code

Watch will open the expression evaluator tab. The expression evaluator is only valid while using the debugger. You can evaluate any drive parameter or user defined variable when the program is paused by the debugger.

Call Stack will open the call stack tab

2.1.9 Format Document

The format document tool implements correct line indentation across all code.

2.1.10 Insert Code Snippet

The Insert Code Snippet tool presents a list of existing code snippets, which can be selected and inserted into the program at the point where the cursor is currently placed.

2.1.11 Upload?

The upload tool uploads the source code from the drive. If the source code has not been downloaded, this option is not available. The tools menu allows you to select if the source code is to be downloaded with the binary.

2.1.12 Lock

The lock tool password protects the source code in the drive. If a new program is downloaded to a drive, the original source code and password protection is erased.

3 AKD BASIC Language

This chapter describes the overall structure of an AKD BASIC program and the elements of the AKD BASIC language. Topics covered are:

- scope
- program structure
    - setup parameters
    - global variables, constants and aliases
    - 'main' program, subroutines, functions and interrupt handlers
- language description
    - lexical conventions
    - identifiers
    - data types
    - constants
    - statements
    - built-in functions
    - parameters
    - expressions
    - function invocation
    - $include
    - arrays and parameter lists
    - optimizations

3.1 AKD BASIC Program Structure

3.1.1 Local Variables

The notion of 'scope' is a key concept in AKD BASIC programs. By 'scope', we mean those parts of the program in which a particular name is 'visible'. There are two levels of scope in AKD BASIC — global and local. Variables (and constant definitions, aliases, etc.) defined inside a 'main' definition, or a subroutine, function, or interrupt handler definition, are considered to be 'local' in scope (visible only within that function).

3.1.2 Global Variables

All other definitions (those occurring outside functions) are considered 'global' in scope (visible inside main, and inside any subroutine, function, or interrupt handler). For example, consider the following simple AKD BASIC program:

```
'-------------- Define (dim) Global Variables -
-------
dim i as integer
'-------------- Main Program -----------------
-------
Main
       dim i as integer
       for i = 1 to 10
       print   "the cube of "; i; "is "; cube(i)
       call increment
       next i
End Main
```

```
'-------------- Subroutines and Functions -----
-------
function cube(i as integer) as integer
       cube = i * i * i
end function
sub increment
       i = i + 1
end sub
```

This program prints a table of the cubes of the integers from 1 to 10. The first (global) definition of 'i' is visible inside subroutine 'increment', but 'shadowed' by the 'i' in main and function 'cube'. The definition of 'i' inside 'main' is local to 'main', and is NOT the same variable as the 'i' inside the function 'cube', or inside the subroutine 'increment'. These same scope rules apply to constant definitions and aliases, as well.

3.2 Program Sections

The major sections of an AKD BASIC program are:

- setup parameter definitions
- global variables, constants, and aliases
- 'main' program, subroutines, functions, and interrupt handlers

Although these sections may appear in any order, we recommend that you keep them in the order shown, or at least, choose a single layout style and use it consistently.

3.2.1 Program Template

The program below is an example of the template generated automatically by WorkBench:

```
'-------------- Device Params -----------------
------
Params


End Params
'-------------- Define (dim) Global Variables -
-------
'-------------- Main Program ------------------
-------
Main

End Main
 '-------------- Subroutines and Functions ----
--------
 '-------------- Interrupt Routines -----------
--------
```

These sections are described in greater detail in the following paragraphs.

3.2.2 Setup Parameter Definitions

This section of the program defines the setup parameters for drive tuning and configuration. It is executed immediately upon power-up (if VM.AUTOSTART = 1), before entering main, and before any interrupts are enabled. The section begins with Param and ends with End Param

(this is similar to the format used to define a subroutine or function). The only statements permitted in this section are assignment statements of the form:

```
<parameter> = <constant expression>
```

This section can be automatically generated by WorkBench when a New Program is created. Choose "current drive parameters" to include the connected drive's setup parameters in your program. If you choose to not include the setup parameters in your program, be sure to save the setup parameters to the drive to ensure proper drive setup upon power up.

3.2.3 Global Variables, Constants, and Aliases

This section contains variables, constant definitions, and global alias expressions — they apply everywhere in the program, unless specifically overridden by another declaration at local scope (inside a subroutine, function, or interrupt handler). Global definitions may be placed almost anywhere in the program text — between subroutines, before or after 'main', and so on.

Global variables, constants, and aliases do not need to be defined before use — the only requirement is that they be defined at some point in the program text. You may have multiple instances of the global variables section throughout your program. However, as a matter of good programming style, we recommend that you keep all global definitions in one place, preferably at or near the beginning of your program.

3.2.4 Variable Definitions

The format of a global variable definition is:

```
dim a,b as integer, x,y,z as float
dim ia(3,4) as integer
dim s1, s2 as string*80
dim sa(5,2) as string
dim j, k, l as integer NV
```

Line 1 declares a and b as integers, x,y, and z as floats. Line 2 declares a 3 x 4 array of integers. Line 3 declares s1 and s2 as strings, each of length 80. Line 4 declares sa as a 5 x 2 array of strings, each with the default length of 32 characters. Line 5 defines 3 integers, j, k, and l as NV.

In addition, global variables can be specified as 'NV' to indicate their values are retained when power is turned off. All other global variables are automatically initialized when the program begins (strings are set to empty, and floats and integers are set to 0). There are no restrictions on the ordering of volatile vs. non-volatile user-variables. For ease of program maintenance, place all non-volatile variables definitions in a single section at the beginning of the program, and add new variables to the end of that section.

3.2.5 Constant Definitions

The format of a constant declaration is:

```
const <name> = <constant_expression>
```

as in

```
const ARRAY_SIZE = 4 * NUMBER_OF_ENTRIES
const PI_SQUARE = 3.1415926535 ^ 2
const GREETING = "Hello"
const GREETING = "Hello"
const NUMBER_OF_ENTRIES = 5
```

Names for constants follow the same rules as variable names. 'Forward definitions' are allowed. Circular definitions are detected and reported at compile-time. Although it is not required, it is convenient to adopt a convention of keeping all constants in UPPER_CASE, so you can easily distinguish between constants and variables in the program.

Constant definitions are entirely 'folded' at compile-time. Feel free to write maintainable constant expressions such as:

```
const LENGTH = 3
const WIDTH = 10
const AREA = LENGTH * WIDTH
```

The value of AREA is computed at compile-time, so the program does NOT need to compute this at run-time and the program is easier to maintain if LENGTH changes at some future date.

3.2.6 Alias Definitions

Aliases allow you to define your own names for system resources, such as input / output pins. The intention is to make it possible for you to use names that are meaningful to you in your particular application. The format of an alias expression is:

```
alias <name> = <expression>
```

For example, the following alias defines application-specific uses of input # 1:

```
alias CONVEYOR_IS_RUNNING = (DIN1.STATE = 0)
alias CONVEYOR_IS_STOPPED = (DIN1.STATE = 1)
if CONVEYOR_IS_RUNNING then
       print"running"
else
       print"stopped"
```

An alias is much more powerful than a constant. Constant expressions are computable at compile-time, while an alias has a value that is only known (in general) at the time it is used. Use aliases with care — too much aliasing can make it very difficult for you to understand the program.

3.3 Main Program, Subroutines, Functions & Interrupt Handlers

These sections share the same fundamental structure:

<section>

<declarations>

<statements>

<section end>

An example of each of these sections follows, with an explanation of key points.

3.3.1 Main Definitions

For main, a typical definition is:

```
Main
       dim i as integer
       i = 1
       print i
End Main
```

The variable 'i' defined above in the 'dim' statement is a local variable — it is not accessible to other functions, and inside 'main', its definition overrides any other variable named 'i' that might exist at global scope.

Unlike global variables, local variables MUST be defined at the beginning of the section — they must appear before any executable statement in main. For example, the following is illegal:

```
Main
        dim i as integer
        i = 1
        dim j as integer        'this is an error!
        j = i
End Main
```

You may also define local constant definitions and aliases, provided that like local variables, they appear before any executable statement. Local constant definitions override global definitions of the same name. For example, given the following global definitions,

```
const N = 1
Main
        const N = "Hello, world!"
        print N
        call sub1
End Main
'-------------- Subroutines and Functions -----
-------
sub sub1
        print N
end sub
```

The program prints to the console:

```
Hello world!
1
```

Because the N visible inside main is the constant defined there, while the N visible to sub1 is the global constant N, whose value is 1.

The main program is the section of your program that is executed immediately after the'params section, regardless of its position in the program text. Other functions, subroutines, and interrupt handlers are executed according to the flow of control defined in the program.

main does not accept arguments, and cannot be called from any other subroutine, function, or interrupt handler.

3.3.2 Subroutine Definition

For a subroutine such as print_sum, a typical definition is:

```
sub print_sum(i,j as integer)
        print i+j
end sub
```

The arguments to this subroutine are specified as integer variables, and are passed by value - any assignments to these variables has no effect on the arguments supplied by the caller. Subroutines are invoked by 'call' instructions, as in call print_sum(3,4).

3.3.3 Function Definition

For a function such as sum_squares, a typical definition is:

```
function sum_squares(i,j as integer) as integer
       sum_squares = i^2 + j^2
end function
```

The function above returns a value of type integer. The value of the function is assigned by assigning to the name of the function, as if it were a variable.

NOTE It is not legal to use the function name as a variable name on the left-hand-side of an assignment - a function name on the leftt-hand-side is always an INVOCATION of that function.

There must be at least one statement in the function that assigns a value to the function. It is not possible to detect at compile-time if the statement will actually execute. Functions are invoked by name, as in print sum_squares(3,4). This is syntactically identical to an array reference.

3.3.4 Interrupt Handler Definition

The Interrupt statment marks the beginning of an Interrupt Service Routine. The Interrupt Service Routine is defined by a program structure resembling a subroutine. The interrupt feature permits execution of a user-defined subroutine upon receipt of a hardware interrupt signal or a predefined interrupt event.

Interrupts are triggered by pre-defined events or external hardware sources. The interrupt-source-name and interrupt enable flag are unique for each interrupt source. For a list of interrupt sources see Interrupt {Source}.

Receiving an interrupt will suspend program execution and the interrupt service routine will be executed. Then program execution will resume at the point that it was interrupted.

Interrupts are enabled (or disabled) by setting (or clearing) the associated interrupt enable flag. Interrupts are disabled until explicitly enabled. After an interrupt is triggered it is automatically disabled until it is enabled again in your program.

```
Main
        DRV.TIME = 0
       INTR.DIN1LO = 1 'enables interrupt
       while 1
               pause(0.5)
               DOUT1.STATEU=0 : Pause(0.005):DOUT1.STATEU=1
       wend
end main
'the following interrupt defines what occurs
for interrupt DIN1LO
 Interrupt      DIN1LO
       print    "I'm awake"
       If    DRV.TIME > 10 then
               print    "OK. That's it."
       else
               INTR.DIN1LO = 1 're-enables interrupt
       end if
 End Interrupt
```

The interrupt is re-enabled by the statement INTR.DIN1LO = 1. A similar statement must be executed once before the interrupt is serviced. It is a run-time error to attempt to enable an interrupt for which no handler is defined.

Interrupt handlers do not return values and cannot have arguments. They declare local variables, constants, and aliases. Interrupt handlers are invoked when the AKD hardware detects that the designated interrupt condition is satisfied (provided that the interrupt is enabled).

Interrupts are triggered by pre-defined events or external hardware sources. The interrupt-source-name and interrupt enable flag are unique for each interrupt source.

Receiving an interrupt will suspend program execution and the interrupt service routine will be executed. Then program execution will resume at the point that it was interrupted.

Interrupts are enabled (or disabled) by setting (or clearing) the associated interrupt enable flag. Interrupts are disabled until explicitly enabled. After an interrupt is triggered it is automatically disabled until it is enabled again in your program.

3.4 Language Definition

3.4.1 Lexical Conventions

AKD BASIC is case-insensitive. String literals are not modified, but all other text is treated as if it were entered in upper case. This means that the identifiers spin, Spin, and SPIN all refer to the same entity.

3.4.2 Identifiers

Identifiers are alphanumeric and must start with an alphabetic character or underscore. In addition, they may include the underscore character ('_') and dollar sign ('$'). Identifiers denote variables, functions, subroutines, and statement labels, symbolic constants, and aliases. Identifiers are a maximum of 40 characters. User-defined identifiers may not include the period ('.'). Use of a longer identifier is a compile-time error.

Although both forms are accepted for compatibility, the latter form is preferred. Although AKD BASIC is case-insensitive, we recommend that you adopt a consistent naming convention, such as Move.Acc, and avoid having Move.acc, move.acc, and move.Acc in the same program.

3.4.3 Data Types

The pre-defined types are INTEGER, FLOAT, and STRING. LONG is used for INTEGER. INTEGER variables are 32-bit signed integers. FLOAT variables are IEEE single-precision floating point numbers. STRING variables are represented internally as a maximum length, a current length, and an array of ASCII characters (can contain null characters).

When a FLOAT result is assigned to an INTEGER variable, or when a FLOAT argument is used where an INTEGER is expected, the value is coerced to an integer before use. Coercion from FLOAT to INT always rounds to the nearest integer. For example:

1.2 rounds to 1

1.7 rounds to 2

-1.2 rounds to –1

-1.7 rounds to -2

Scalar INTEGER and FLOAT coercion is automatically provided for function arguments. When passing ARRAYS as arguments, the types must match exactly because coercion is prohibitively expensive at run-time.

String assignment is checked at run-time. An attempt to copy a string to a destination too small results in a run-time error. String indexing is 1-origin. For example, mid$("abc",1,1) returns the string, a.

STRING variables have a firmware-imposed maximum length of 230 characters and a default maximum length of 32 characters. They may be assigned a different maximum length by declaring them to be of type STRING*n where n is a positive integer between 1 and 230 (inclusive).

Declare arrays of the pre-defined types. Arrays have a maximum rank of four dimensions. The upper-bound of each dimension has no compiler-defined limit. However, because of the limited data space of the controller, there is a logical upper-bound that depends on the controller model.

Array indexing is 1-origin. The indices in each dimension range from 1 to the upper-bound of the dimension. Every reference to an array element is checked at run-time. Any attempt to reference beyond the bounds of the array causes a run-time error. New types cannot be defined.

3.4.4 Literal Constants

String constants begin and end with the double-quotes (""). They cannot extend past the end of the input line. Any printable ASCII character appears in a string constant. An attempt to generate a string literal with non-ASCII characters causes a compile-time error. No check is made to verify that non-ASCII strings are not created at run-time, so avoid doing so.

3.4.5 Decimal Integer Constants

Decimal integer constants are a string of decimal digits with no decimal point. A leading '-' sign is optional and is parsed as a unary minus. For example:

1

-1

314159

are all valid decimal constants.

3.4.6 Hexadecimal Constants

Hexadecimal constants are denoted by a leading &H or &h, and cannot have a sign or decimal point. Hexadecimal constants are composed from the set [0-9A-Fa-f]. Upper- and lower-case may be mixed.

For example:

&h00ff

&HFF00

&H1234abcd

are all valid hexadecimal constants. Octal and binary constants are not supported.

3.4.7 Floating-Point Constants

Floating-point constants are specified in fixed-point or mantissa-exponent notation. A floating-point constant consists of one of the following.

digit	[0-9]
optsign	'+' | '-' | /* nothing */
fixed	optsign {digit}+ '.' {digit}*optsign '.' {digit}+
exp	fixed 'e' optsign {digit}+
float	fixed | exp

For example:

0.1
.1
-.1
-0.1
3.14159E-6
-1.0E6

are all valid floating point constants. By design, "." is not a legal floating-point constant.

3.5 Statements

Statements are separated by a new line (CR-LF) or a colon (':'). The statements of the language are:

3.5.1 Alias

Create an alias for an identifier (not just any identifier). Alias is either a parameter or another alias. ID must be a legal variable name. You cannot create an alias for an array element.

Like Const definitions, Alias definitions can be made to identifiers not yet defined. Circular definitions are not allowed.

Any duplicate definition of an identifier in the same scope is illegal. However, a local definition can shadow a definition from the global scope. Using a single identifier to denote two different objects is NOT allowed (i.e., you cannot have both a label and a variable named all_done).

Like constant, variable, and function declarations, Alias declarations made in the global scope are imported into all functions (including the main function).

Example:

```
Alias speed = MOTOR.SPEED       'save some key-
strokes
```

3.5.2 Call

```
Call sub[(arg1, arg2, ...)]
```

sub is the name of a subroutine. The current program counter is saved and sub is invoked. When sub finishes (by reaching either an exit sub or end sub statement, control is returned to the statement logically following Call.

A subroutine is essentially a function with no return value. The parameter passing conventions followed by subroutines are the same as those followed by functions.

3.5.3 Cls

This statement transmits 40 line-feed characters (ASCII code = 10) to the serial port. Cls clears the display of the console.

3.5.4 Const

```
const name = x
```

Declares symbolic constants to be used instead of numeric values. Forward references are allowed, but circular references are not supported.

```
'supported
const x = y + 2
```

```
const y = 17
'unsupported
const x = y + 2
const y = x - 2
```

Like alias, variable, and function declarations, Const declarations made in the global scope are imported into all functions (including the main function).

3.5.5 Dim

```
Dim var1 [, var2 [...]] as type [NV]
```

All variables must be declared. Local variables must be declared in the function before use.

The NV specifier is used on a Dim statement in the global scope.

Variables in the global scope are automatically imported into functions and subroutines. Variables in function scope (including inside the main function) are not accessible in other functions.

Arrays cannot be assigned directly.

```
'This is not allowed
dim x(5), y(5) as integer
x = y
'Instead, a loop is needed:
dim x(5), y(5), i as integer
for i = 1 to 5
       x(i) = y(i)
next i
```

3.5.6 Exit

Exit {{Sub|Function|Interrupt|For|While}]

Exits the closest enclosing context of the specified type. It is a compile-time error to EXIT a construct not currently in scope.

3.5.7 For…Next

For loop_counter = Start_Value To End_Value [Step increment]

...statements...

Next

If step increment is not specified, uses 1 as the step increment. If step increment is positive, continues to the value of End_Value. If step increment is negative, continues to the value of var = limit.

The loop index variable must be a simple identifier, not an array element or a parameter and must be a numeric variable (integer or float).

```
for var = init to limit step delta
       stlist
next var
```

Substantially more efficient code is generated if delta is a constant (i.e., the default value of 1 is used, or specified as an expression that is evaluated at compile-time).

3.5.8 Function

Function function-name [(argument-list)] as function-type

...statements...

End Function

On function entry, all local variable strings are “” and all numeric locals are zero (including all elements of local arrays). If the function takes no arguments, omit the argument-list. An empty argument-list is illegal. The value returned from the function is specified by assigning an identifier with the name of the function.

Example:

```
function cube(x as float) as float
       cube = x * x * x
end function
```

Arguments are passed by value. Arrays can not be returned by a function. Arrays passed to a function are passed by value.

If the return value is not set, a runtime error condition is generated (caught with ON ERROR).

Array actuals must conform with formals to the extent that they have the same number of dimensions, and EXACTLY the same type. The size of each dimension is available to the function through the use of local constants that are bound on function entry.

Example:

```
function sum(x(n) as integer) as integer
       dim i, total as integer
       sum = 0
       for i = 1 to n
                total = total + x(i)
       next
       sum = total
end function
```

This function exploits the fact that the variable N is automatically assigned a value when the function is called and the value is the extent of the array passed on invocation. N is a read-only variable in this context. Attempts to write to N cause compile-time errors.

The local variable, total is automatically initialized to 0 upon function entry.

3.5.9 GoTo

GoTo label

A program can only GoTo a label in the same scope. A GoTo may jump out of a For or While loop, but not INTO one.

3.5.10 If…Then…Else

```
if condition1 then
       ...statement block1...
elseif condition2 then
       ...statement block2...
else
       ...statement block3...
end if
```

IF...THEN...ELSE statements control program execution based on the evaluation of numeric expressions. The IF...THEN...ELSE decision structure permits the execution of program statements or allows branching to other parts of the program based on the evaluation of the expression.

There are two structures of IF... THEN...ELSE statements, single line and block formats.

3.5.11 $Include

```
$include inclfile
$include include-file-name
```

Textually include inclfile at this point in the compilation. There can be no space between $ and include. The $include directive must start at the beginning of the line.

3.5.12 Input

```
input [prompt-string][,|;]input-variable
```

Input reads a character string received by the console tab in the program view, terminated by a carriage return.

As an option, the prompt message is transmitted when the Input statement is encountered. If the prompt string is followed by a semicolon, a question mark is printed at the end of the prompt string. If a comma follows the prompt string, no question mark is printed. This input statement is typically used for debugging purposes.

3.5.13 Interrupt … End Interrupt

```
interrupt {Interrupt-Source-Name}
...program statements...
end interrupt
```

Interrupt handlers can be located anywhere in the program text (e.g., before main).

3.5.14 MOVE.ABORT

MOVE.ABORT stops motor motion and allows continued program execution. Deceleration is determined by the controlled stop deceleration rate (CS.DEC).

3.5.15 MOVE.GOABS

MOVE.GOABS (Go Absolute) moves the motor to the position specified by MOVE.TARGETPOS. This position is based on a zero position at electrical home.

The motor speed follows a velocity profile as specified by MOVE.ACC, MOVE.RUNSPEED, and MOVE.DEC . Direction of travel depends on current position and target position only (MOVE.DIR has no effect). After the program initiates MOVE.GOABS, it immediately goes to the next instruction.

Change MOVE.ACC, MOVE.RUNSPEED, and MOVE.DEC during a move using MOVE.GOUPDATE.

3.5.16 MOVE.GOHOME

MOVE.GOHOME moves the motor shaft to the electrical home position (PL.FB = 0).

The motor speed follows a trapezoidal velocity profile as specified by MOVE.ACC, MOVE.RUNSPEED, and MOVE.DEC. After the program initiates MOVE.GOHOME, it immediately goes to the next instruction.

MOVE.GOHOME performs the same action as setting MOVE.TARGETPOS to zero and executing a MOVE.GOABS function. Change MOVE.ACC, MOVE.DEC and MOVE.RUNSPEED during a move using MOVE.GOUPDATE

3.5.17 MOVE.GOREL

MOVE.GOREL (Go Relative) moves the motor shaft a relative distance from the current position.

Distance, as specified in MOVE.RELATIVEDIST, is either positive or negative. The motor speed follows a trapezoidal velocity profile as specified by MOVE.ACC, MOVE.RUNSPEED, and MOVE.DEC.

The program does not wait for motion completion. After the program initiates this move it immediately goes to the next instruction.

Change MOVE.ACC, MOVE.RUNSPEED, and MOVE.DEC during a move using MOVE.GOUPDATE.

3.5.18 MOVE.GOUPDATE

MOVE.GOUPDATE (Update Move) updates a move in process with new variables. This allows you to change motion “on the fly” without having to stop and restart the motion function with new variables.

3.5.19 MOVE.GOVEL

MOVE.GOVEL (Go Velocity) moves the motor shaft at a constant speed.

The motor accelerates and reaches maximum speed as specified by MOVE.ACC and MOVE.RUNSPEED, with direction determined by MOVE.DIR. Stop motion by:

- Programming MOVE.ABORT for maximum deceleration allowed by current limits.
- Programming MOVE.RUNSPEED = 0 for deceleration at rate set by MOVE.DEC.

After the program initiates MOVE.GOVEL, it immediately goes to the ne xt instruction.

Change variables during a move using MOVE.GOUPDATE.

3.5.20 On Error GoTo

```
On Error Goto Error-Handler-Name
```

or

```
On Error Goto 0
```

When a firmware runtime error condition occurs, Error-Handler-Name is called, the error handler is de-installed, and an internal flag (inerror-handler) is set. Any subsequent runtime error (including attempting to set the error handler, or return from the On Error handler) causes an immediate Stop.

On Error Goto 0 disables the current On Error handler. If an error occurs when no error handler is installed, Stop is invoked.

3.5.21 Pause( )

Pause(Pause_Time) causes the program to pause the amount of time specified by the Pause_Time argument. The motion of the motor is not affected.

3.5.22 Print

```
print expression1 [ [,;] expression2 ] [;]
```

Print a list of expressions, separated by delimiters to the console. Any number of delimiters (including zero) can appear before or after the list of expressions. At least one delimiter must appear between each pair of expressions in the print list. The print statement is primarily used for debugging purposes.

3.5.23 Example

```
print      ' print a newline
print , ' advance a single tab stop
print a,b ' print a and b, tab between
print a,b, ' print a and b, tab between and at
end
print ,,,x,,, ' tab tab tab x tab tab tab
```

3.5.24 VM.RESTART

VM.RESTART clears the run time error variables and causes program execution to start again from the beginning of the program. Any Interrupts, Subroutines, WHEN statements or loops in process are aborted. This statement is used to continue program execution after a Run Time Error Handler or to abort from WHEN statements without satisfying the condition.

VM.RESTART does not clear the data area or change any program or motion variables.

3.5.25 Select Case

```
Select Case test-expression
       Case expression-list1
              ...statement block1...
       Case expression-list2
              ...statement block1...
       Case expression-list3
              ...statement block1...
       Case Else
              ...else block...
End Select
```

test-expression must evaluate to an INTEGER or FLOAT value.

expression-list1 is a non-empty list of case-defn, separated by commas.

There can be only one Case Else and, if present, it must appear as the last case. It is selected only if all other tests fail.

case-defn can be any of the following:

```
expr
expr to expr (tests inclusive (closed range))
```

```
is relop expr (<, =, =, =, > )
is expr (equiv to      "is = expr")
```

Select-case statements where the case-defn expressions are composed solely of integer constants are evaluated much quicker at run-time. (Cases involving variables must be transformed to logically equivalent if-then-else statements.)

3.5.26 Static

Static var1 [, var2[...]] as type

where type is:

INTEGER 32 bit integer

FLOAT IEEE single precision float

STRING default length is 32 characters

Static is used for declaring variables before use. All variables (except parameters) must be declared before they can be used. The Static statement is used in a Function, Sub or Interrupt to specify that the specified variable's value be remembered even when the Function or Sub is finished. The next time that the Function, Sub or Interrupt is executed, the value will be available.

Example:

```
Main
       while 1
               call MySub
               pause(1)
       wend
End Main
'-------------- Subroutines and Functions -----
-------
sub MySub
       dim x as integer      'value is forgotten
       static y as integer      'value is remembered
       x= x + 1
       y = y + 1
       print x,y
end sub
```

3.5.27 Stop

Stops the execution of the program.

3.5.28 Sub…End Sub

```
Sub [argument-list]
       ...body of the sub-procedure...
End Sub
```

Declare a subroutine. Invoked via Call. Optionally takes arguments. As with Function, it is illegal to provide an empty parameter list (‘()’) if the subroutine takes no parameters.

3.5.29 Swap

```
Swap x, y
```

Swaps the values of the variables. The variable types must be the same. Does not work on arrays or strings.

3.5.30 When

```
When when-condition , when-action
```

When is used for very fast output response to certain input conditions. You specify the condition and action. Upon encountering When, program execution waits until the defined condition is satisfied. The program immediately executes the action and continues with the next line of the program.

The When statement provides latching of several variables when the When condition is satisfied. These variables are:

WHEN.DRVHANDWHEEL	WHEN.FB1MECHPOS
WHEN.PLCMD	WHEN.DRVTIME
WHEN.PLFB	

The software checks for the defined condition at the 4Khz rate. The when action is queued up and executed immediately. The when action will be executed within 25 microseconds of the when condition being met.

3.5.31 While...Wend

```
While condition
       ...statement block...
Wend
```

While...Wend tells the program to execute a series of statements as long as an expression after the While statement is true.

If the expression is true, the loop statements between While and Wend are executed. The expression is evaluated again and if the expression is still true, the loop statements are executed again. This continues until the expression is no longer true. If the expression is not true, the statement immediately following the Wend statement is executed.

3.6 Built-in Functions

A function that takes a numeric argument (either FLOAT or INTEGER) returns the same type. Coercion between INTEGER and FLOAT is not performed unless necessary. (notation - the arguments n and m refer to INTEGER types, as in the definition of the MID$ function, whose signature is MID$(string, integer, integer).

Name	Args	Return	Semantics
ABS	numeric	numeric	absolute value
ATAN	float	float	arc tangent (radians)
CINT	numeric	int	truncate (round to nearest int)
COS	float	float	cosine

Name	Args	Return	Semantics
EXP	float	float	e ^ arg, arg 88.02969 (o/w overflow)
FIX	numeric	int	truncate (round toward zero)
INT	numeric	int	truncate (round towards -INFINITY)
LOG	float	float	natural log
LOG10	float	float	log base 10
SGN	numeric	integer	sign of argument: -1, 0, 1
SIN	float	float	sine (radians)
SQR	float	float	square root of arg
TAN	float	float	tangent (radians)

String function			Description
ASC	string	int	ASCII code for 1st char
CHR$	int	string	One-character string containing the character with the ASCII code of arg. If arg 255, returns CHR$(arg % 256).
HEX$	int	string	Printable hexadecimal rep of arg (without leading &H).
INKEY$		string	One-character string, read from serial port.Returns “” if no char available.
INSTR	[pos],str1,str2	int	Index of str2 in str1, or 0 if not found. Optional first arg specifies where to start search (defaults to position 1).
LCASE$	str	str	Returns lower-case copy of arg.
LEFT$	str,n	str	Returns n leftmost chars of str.
LEN	str	int	Returns length of str in bytes.
LTRIM$	str	str	Trim leading spaces.
MID$	str,n[,m]	str	Returns substring starting at position n [for up to to m bytes].
OCT$	n	str	Octal string representation of arg.
RIGHT$	str,n	str	Rightmost n chars of str.
RTRIM$	str	str	Trim trailing spaces.
SPACE$	n	str	Returns a string of n spaces.
STR$	n	str	Decimal string representation of str.
STRING$	n,str	str	Return n copies of first char of str.
TRIM$	str	str	Trim leading AND trailing spaces.
UCASE$	str	str	Returns upper-case copy of arg.
VAL	str	numeric	Returns numeric value of str.

3.6.1 Parameters and Commands

The AKD BASIC language is augmented by a set of parameters, input\output parameters, and pre-defined commands., The parameters set motor-specific control parameters, and the pre-defined commands control the motor.

For example, MOVE.ACC, MOVE.DEC, and MOVE.RUNSPEED are used to set the acceleration rate, deceleration rate, and commanded motor speed for the next commanded move:

```
MOVE.ACC = 1000.0
MOVE.DEC = 1000.0
MOVE.RUNSPEED = 500.0
MOVE.GOVEL
```

The program fragment above sets up the relevant motion parameters, and commands the motor to move at the specified velocity.

You cannot create variables (or function names, etc.) that shadow pre-defined ones. For a complete list of Parameters and commands, refer to the Parameter Guide section in this manual.

3.7 Expressions

3.7.1 Arithmetic Expressions

Arithmetic expressions (expressions involving INTEGER and FLOAT values) use the following operators.

3.7.2 Numeric Operators

Operators higher in the table have greater precedence than those below.

Operator	Assoc	Name
^	right	exponentiation
-	right	unary minus
*	left	multiply
/	left	divide
MOD	left	modulo
+	left	add
-	left	subtract

3.7.3 Logical Operators

Operator	Assoc	Explanation
=, < >, ≥, ≤, <, >	left	the usual
NOT, BITNOT	right	not, boolean not
AND, BITAND	left	and, boolean and
OR, BITOR, XOR, BITXOR	left	or, boolean or, xor, boolean xor

Logical expressions (as, for example, in the condition of an 'if' statement) also use these operators. Strings are concatenated with the '+' operator. Logical expressions are formed from strings, using the comparison operators, NOT, AND, OR, and XOR, with the meaning of an empty string being FALSE, and a non-empty string being TRUE.

Integer values are coerced to floating point values as needed. Floating-point values are rounded when coerced to integer values.

Logical operators are NOT short-circuiting (i.e., when executing the code).

if a(x) or b(y) or c(z) then ...

if a(x) is true, b(y) and c(z) are still invoked.

BITxxx boolean operators are provided to support bitwise operations on integer values. They operate quite differently from their logical equivalents. For example:

2 and 1 has the value –1 (TRUE, since each operand is 'true'), but 2 bitand 1 has the value 0 since no matching bits are 1).

Similarly, 3 or 4 has the value –1(TRUE since at least one operand is not FALSE), while 3 bitor 4 has the value 7 (the three lsb's are set).

Remember that relational and logical operators return numeric values - 0 for FALSE and -1 for TRUE. Any value not equal to FALSE is considered to be logically equivalent to TRUE for purposes of the logical operators.

It is syntactically incorrect to code:

```
dim a, b, c, x as integer
x = a < b < c
```

3.7.4 String Operators

Operator	Assoc	Name
<, >, ≤, ≥	nonassoc	string comparisons
=, <>	nonassoc	string comparisons
	left	string concatenation

There is no implicit coercion between strings and numeric types. String comparison is case-sensitive. Relative comparisons are made using ASCII lexical ordering. The empty string sorts before all other strings. String comparison operators are non-associative because they evaluate to a numeric value.

3.7.5 Example

It makes no sense to say a$ = b$ = c$.

It is sensible to say x = a$ = b$

x is assigned the value TRUE if a$ is the same as b$, and FALSE otherwise.

3.8 Function Invocation

A function invocation is denoted as:

```
var = func(arg1, arg2, ..., argn)
```

The arguments are passed by value (i.e., modifications made to the formal parameters inside a function are not reflected in the actuals). Arrays are also passed by value to functions. Arrays cannot be returned by a function. A function of no arguments is invoked by using the function name alone. For example, if func_none takes no arguments, then func_none is correct and func_none() is invalid.

The return value of a function may not be ignored by the caller. If the return value of a function is regularly ignored, the function should be rewritten as a subroutine (a function with no return value).

3.8.1 $INCLUDE

Use $INCLUDE to textually include one file in another. The $INCLUDE facility is a simple, powerful way to create a consistent family of applications. By including source files containing commonly used functions, subroutines, constant definitions, aliases, etc., you have control over the source for each application. When you change the source, you update each application simply by recompiling (see Optimizations).

A file cannot include itself, either directly or indirectly. Include file nesting is allowed, but limited to a pre-defined maximum depth (currently 16).

The path of an include file is relative to the directory of the included file, not the current working directory of the compiler. Suppose, for example, the source program is in directory C:\WORK,

and includes the file .C\H\HEADER, and the file HEADER includes COMMON. The compiler looks for COMMON in C:\H, not in C:\WORK.

```
C:\WORK
   A.BAS
      $INCLUDE "..\H\HEADER"
C:\H
   HEADER
      $INCLUDE "COMMON"
```

Compilation errors occur when a file is included multiple times. For example, if B.BAS includes files MATH and INCL, and INCL also includes MATH, MATH is included twice, causing a compile-time error.

```
B.BAS
   $INCLUDE "MATH"
   $INCLUDE "INCL"
INCL
   $INCLUDE "MATH"
```

3.9 Arrays and Function Parameter Lists

When an array parameter (formal) of a function or subroutine is declared, the number of dimensions is specified, but the extent of (number of elements in) each dimension is not specified. This allows the programmer some freedom when invoking such a function. For example, a function may be defined to take a one-dimensional array and compute the sum of the elements in the array. A single function can be written to take a one-dimensional array of any size and correctly compute the sum. (Because AKD BASIC checks array bounds at run time on each access, there is no risk that a function will read or write outside the bounds of the array.)

When a formal parameter to a function is an array, instead of specifying the extent of each dimension, a list of variables is used to both implicitly specify the number of dimensions and to hold the extent of each dimension. These variables are read-only and cannot be modified within the function.

Adopt a convention for assigning names to placeholders. One such convention is to use the name of the array with a numerical suffix. For example,

```
function f(a(a1,a2,a3) as integer) as integer
```

where a1, a2, and a3 are the variables that get the extents of the array, a.

The function f above would be called as follows:

```
dim x_array(3,4,5) as integer
dim y_array(1,2,10) as integer
print f(x_array()) + f(y_array())
```

In both invocations of f, the function correctly determines the extent of each dimension of the passed array.

Remember that when passing an array to a function, the type of the array must match EXACTLY with the type expected by the function. Unlike scalar arguments (implicitly coerced from float to int or int to float), arrays are NOT coerced. An attempt to pass an integer array to a function that expects a float array results in a compile-time error.

3.9.1 Optimizations

As mentioned in an earlier section, constant definitions are completely 'folded' at the point of definition. This is efficient code. Constant expressions inside AKD BASIC statements are also

folded under certain conditions. For example, in the statement:

```
const PI = 3.1415926535
Main
        Print PI^2
End Main
```

The value of PI^2 is not computed at run-time. It is detected as a constant value and pre-computed by the compiler as a single literal constant to be printed.

Similarly, the literal constant 3*4*PI in

```
x = 3 * 4 * PI * x
```

is folded at compile-time, leaving only one multiplication to be performed at run-time.

However, certain constant expressions are not folded. For example:

```
x = 3 * PI * x * 4
```

is computed at run-time, involving 3 multiplications because the analysis of constant expressions does not attempt to exploit algebraic commutativity laws. Since the basic arithmetic operators are 'left associative', you can ensure the best performance by grouping constant factors together towards the left (or using a new constant definition).

If a function is not referenced (transitively from MAIN, plus any interrupt handlers), the compiler does not generate code for it. So, you can freely $include libraries with unused code (e.g., a comprehensive library containing functions supporting several possible axis configurations). Although the compiler parses and type-checks all the included source, it does not generate code into the downloaded program.

If select-case cases are all constants, more efficient code is generated. If a case is a variable, the generated code is equivalent to a string of if-then-else statements for all cases.

If any of the cases is an open-ended range (e.g., is 10), or covers a large range (e.g., 1 to 1000), a fast table-lookup is generated.

If all of the cases are constant, and can be grouped into locally dense subsets, the fastest possible code is generated — a binary search of dispatch tables, followed by an indirect jump through the table. If speed is a consideration, keep your cases constant and close together. (values form a reasonably dense set.)

The compiler performs limited dead-code elimination based on simple constant analysis. For example:

```
const DEBUGGING = FALSE
Main
        dim i, sum as integer
        for i = 1 to 10
                sum = sum + i
                if DEBUGGING then print   "partial sum is
";sum
        next i
End Main
```

Since the value of DEBUGGING is FALSE, the compiler recognizes that the printing of the partial sum never happens and does not generate the print statement. This allows you to place debugging code in strategic locations in your programs and effectively disable it when shipping a production version (shrinks the size of the generated code).

This dead-code elimination also applies to functions whose only point of reference lies in eliminated code. The functions themselves become dead-code and no code is generated for their definitions.

The compiler does not eliminate the print statement from the following program:

```
dim DEBUGGING as integer
Main
        dim i, sum as integer
        DEBUGGING = FALSE
        for i = 1 to 10
                sum = sum + i
                if DEBUGGING print    "partial sum is ";sum
        next i
End Main
```

In this case, the print statement never executes, but the code to implement is generated because the value of the integer DEBUGGING could be changed by the AKD's Integrated Development Environment Debugger at runtime, causing the print statement to be executed!

3.10 ModBus TCP/IP

Modbus TCP/IP, or Modbus TCP, is a Modbus variant used for communications over TCP/IP networks, connecting over port 502. A max of 3 masters can be connected to one drive at any time. Modbus standard limits one master to 256 slaves. Modbus and Workbench (telnet) can be connected to the same drive at the same time. Because Modbus and Telnet are processed in the AKD's background task, 5~10mS delay between messages will prevent over running the back ground task stack limits.

3.10.1 ModBus Parameter Table

For a list of parameters and their Modbus addresses, visit the Modbus Parameter Table.

For 64 bit to 32 bit mapping, visit Modbus 64-bit Parameters to 32-bit Mapping.

3.10.2 ModBus Register and Data Types

All predefined AKD parameters are Modbus 32 bit or 64 bit (some with and some without sign).

Because most HMI products don't support 64 bit numbers, the AKD also has a series of matching 32 bit parameters. Ex: PL.FB is a signed 64 bit integer with address number 588. There is also PL.FB_32 as a signed 32 bit integer with address number 2072. PL.FB_32 is the lower two 16 bit registers of PL.FB.

Typical HMI tag addressing will need the additional "40000" be added to the Modbus address number. Ex: To create a tag in an HMI to read PL.FB_32, the address number 42072 should be used.

All AKD command parameter will require writing a "1" to trigger the command. Ex: DRV.EN is used to enable the drive. To trigger the command, send "1" as a 32 bit integer to address 254.

3.10.3 User Created Variables with Assigned Modbus Address Numbers

User variables can be assigned an Modbus address number. The range of available numbers is from 5000 to 5999. An example program:

```
Dim int2 as integer
Dimflt1 as float
Dim long1 as long
```

```
MBInfo
$MBMap32(5001, int2)
$MBMap64(5003, long1)
$MBMapfloat(5007, flt1)
End

'-------------- Main Program ------------------
-------
Main
        'setup some data to be read
        int2 = 262144
        flt1 = 1.234
        Long1 = 17179869184
End Main
```

Notice that address numbers have to be skipped for mapped varables larger than 16 bit. Ex: In the program above, 5001 was assigned for a 32 bit integer and then next number available would be 5003.

3.10.4 Drive Fault Table

The parameter MODBUS.FAULT1 to MODBUS.FAULT10 is used to read any drive fault condition. The fault parameters are loaded with the AKD fault code, starting with parameter MODOBUS.FAULT1. DRV.CLRFAULT will reset the drive and clear out any data in MODBUS.FAULT1 to MODBUS.FAULT10.

HMI fault tables only need to monitor MODBUS.FAULT1, but report faults, if present, in 1 to 10.

3.10.5 Drive Parameter Scaling Over Modbus

The predefined AKD velocity, position, acceleration, and deceleration parameters use the Modbus scaling and do not use the “user units” configured in Workbench.

MODBUS.PSCALE

MODBUS.PIN

MODBUS.POUT

The default is 2^20 counts/ rev, 2^20 counts/sec, and 2^20 counts/sec^2

User defined variables, which have assigned Modbus address numbers, are not effected by this scaling.

3.10.6 Special Modbus AKD Parameters

MODBUS.DIO

Bit 0 to 6	DIN.STATES
Bit 16 and 17	DOUT.STATES

MODBUS.DRVSTAT

Bit 0	Drive Active
Bit 1	STO Status
Bit 2	Pos Hw Limit
Bit 3	Neg Hw Limit
Bit 4	Pos Sw Limit
Bit 5	Neg Sw Limit

MODBUS.DRV

Bit 0	Stop
Bit 1	Enable

MODBUS.MOTOR

Bit 0	Has Brake
Bit 1	Brake Release

MODBUS.HOME

Bit 0	Start Homing
Bit 1	Set (current position as home position)

MODBUS.MT

Bit 0	Clear (clears motion task MT.NUM)
Bit 1	Continue
Bit 2	Load (load motion task MT.NUM)
Bit 3	Set
Bit 4	Move (start move of MT.NUM)

MODBUS.SM

Bit 0	One Direction (sets SM.MODE to either 0 or 1)
Bit 1	Start Move: Edge triggered 0 → 1 : Start motion (execution of SM.MOVE) 1 → 0 : Stop motion (execution of DRV.STOP)

3.10.7 ModBus Dynamic Mapping

For information on Modbus Dynamic Mapping visit Modbus Dynamic Mapping under Fieldbus Manuals.

3.11 Cam Profiling

In the AKD, a cam is a cyclic, generally non-linear relationship between master encoder position and slave (motor) position. The relationship between slave and master counts is no longer a constant ratio, but changes as a function of master counts. As in electronic gearing, once a cam is active, the program no longer needs to do anything special to maintain it - the motion profile is repeated indefinitely until the cam is deactivated.

In camming terminology, a master is typically an external encoder. The encoder is wired into the AKD BASIC encoder input port on X9. It is also possible to use the AKD BASIC's virtual (internal) encoder.

3.11.1 Procedure

To use a cam profile on the AKD BASIC, you must:

1. Create the cam profile (CAM.CREATE).
2. Activate the cam profile (CAM.ACTIVATE).

3.11.2 Related Variables

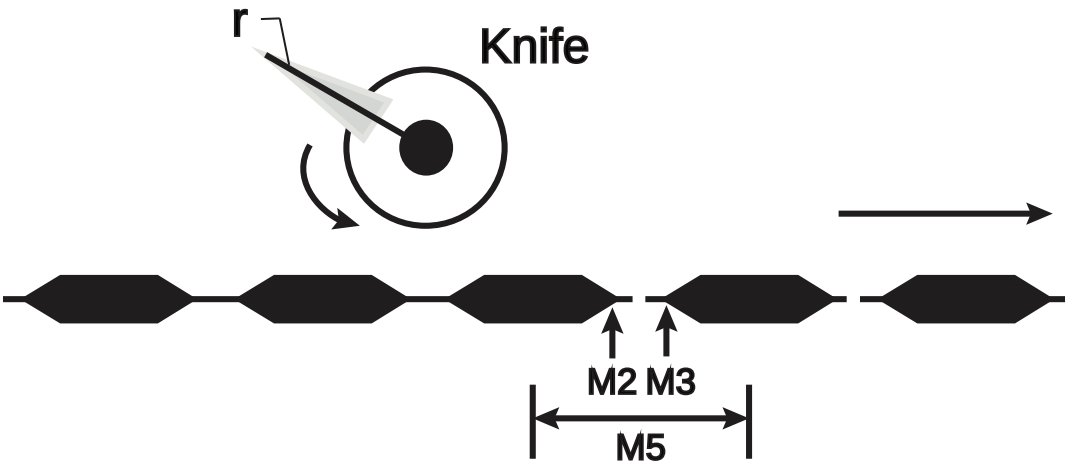
CAM.MASTER: Specifies the source of the input to the cam table for cam profiling.

CAM.CORRECTDIR: Specifies the direction of the correction move that is done when a new cam table is activated (by setting CAM.ACTIVATE = n).

CAM.ADDPOINT: Adds the specified “point” (master position and corresponding slave position) to the cam table being created.

3.11.3 Cam Wizard

The Cam Wizard is designed to solve cut to length applications. The picture below shows a typical setup:



In this application, material is being fed beneath a rotary knife. The master encoder measures forward movement of the material under the knife. The slave motor controls rotation of the knife. In order for this to work properly, the slave motor must be controlled (as a function of master encoder counts) so the blade of the rotary knife:

1. Stays out of the way until the proper amount of material has passed
2. Accelerates so the speed of the knife matches the speed of the material during the cut
3. Decelerates back to the original speed until the material is almost in position for the next cut

NOTE The rotary knife either accelerates or decelerates to match the speed of the material in the cut phase, depending on whether or not the circumference of the rotary knife is less than or greater than the length of the piece to be cut. You may need to interchange the terms ‘accelerate’ and ‘decelerate’, or simply think of them as signed quantities.

AKD BASIC’s CAM.ADDPOINT statement specifies a cam profile as a mapping from master position to slave position. The problem refers to relative velocities and accelerations. It is not always clear how to get from velocity and acceleration to position.

The Cam Wizard was designed to make such applications easy to implement. You provide:

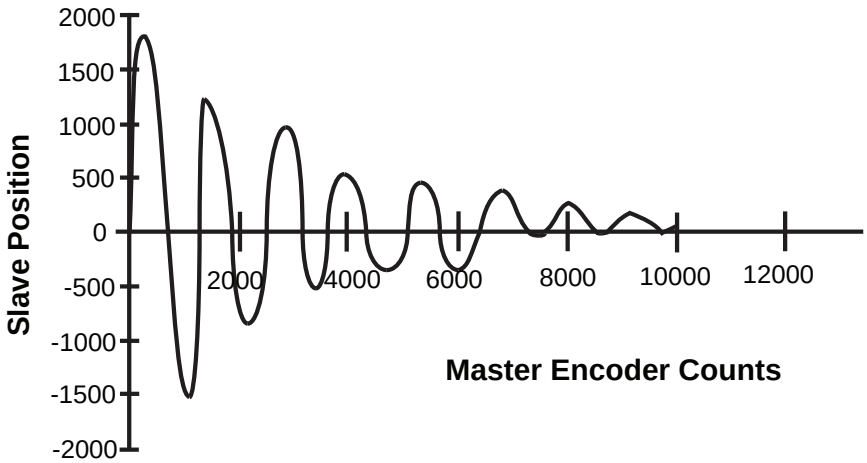
1. The master positions corresponding to the length of material to be cut
2. The slave positions corresponding to one complete rotation of the knife with respect to the master positions

Once you have provided these three pieces of information, the Cam Wizard automatically:

1. Generates a subroutine to create the cam table
2. Generates a subroutine to activate the cam

3.11.4 Example

You can create a cam to approximate any continuous function, but the Cam Wizard cannot help you with it. The basic technique is to develop an AKD BASIC expression (or function)defining the slave position as a function of master position and use it to generate a series of CAM.ADDPOINT statements at appropriate master position intervals, such as the one shown in the next figure.



3.11.5 Program

```
'-------------- Device Params -----------------
------
Params
End Params
'-------------- Define (dim) Global Variables -
-------
const MC = 10000 'master counts in total cycle
const NPOINTS = 501 'number of points in cam
profile
const pi = 3.1415926535
const k = 0.69314718 / 100
const w = 1 / (7.5 * pi)
'-------------- Main Program -----------------
-------
Main
       CAM.ACTIVATE = 0 'Turn off any active cams
       CAM.MASTER = 1 'Master = virtual encoder only
       CAMVM.DIR = 0 'set direction of virtual master

       CAMVM.FREQ = 1000 'set virtual master speed
       'Generate a cam that does exponentially-damped
sinusoidal
       'motion and activate it. Please note that
since we are computing
       '500 points of slave profile here several sec-
onds will elapse
       'during the calculation of the cam table.
       call CamCreate_2
```

```
        call ActivateCam_2
        CAMVM.GOVEL    'virtual master to run at
CAMVM.FREQ
        While 1 : Wend
End Main

'--------------- Subroutines and Functions -----
-------
Sub CamCreate_2'This code creates a cam whose
profile is an exponentially damped sine wave.
        dim m, s as float
        dim i as integer
        CAM.CREATE (2,501)
                for i = 0 to NPOINTS-1
                        m = i * (MC / (NPOINTS-1))'master position
                        s = (1 / exp (1.5 * k * i)) * sin (2 * pi *
w * i)'computed slave position
                        CAM.ADDPOINT(m, 65536 * s)
                Next i
        End   CAM.CREATE
End Sub

Sub ActivateCam_2
        DRV.SWENABLE = 0 'Need to disable the drive
before changing positions
        MOVE.POSCOMMAND = 0 'Zero out slave position
        EXTENCODER.POSITION = 0 'Zero out real master
encoder position
        CAMVM.POSITION = 0 'Zero out virtual master
encoder position
        EXTENCODER.POSMODULO = MC 'Set master modulo
value
        PL.MODPEN = 1 'Enable slave modulo
        PL.MODP2 = 65536 'Set slave modulo value
        DRV.SWENABLE = 1 'Enable drive
        CAM.ACTIVATE = 2 'Start the cam
End Sub
```

3.11.6 Virtual encoder (virtual master)

The virtual encoder is an internal count generator that is used as the input to the cam. It is controlled much like the profile generator used to control the motion of the motor. The Parameters and statements associated with the virtual encoder are listed below:

CAMVM.DIR

CAMVM.FREQ

CAMVM.GOREL

CAMVM.GOUPDATE

CAMVM.GOVEL

CAMVM.MOVING

CAMVM.POSITION

CAMVM.RELATIVEDIST
CAMVM.STOP

3.11.7 Move Parameters

CAMVM.DIR specifies direction for CAMVM.GOVEL
CAMVM.RELATIVEDIST specifies distance for CAMVM.GOREL
CAMVM.FREQ specifies speed (frequency) for CAMVM.GOREL and CAMVM.GOVEL

3.11.8 Move Statements

CAMVM.GOREL executes incremental move
CAMVM.GOVEL executes velocity move
CAMVM.GOUPDATE updates move parameters on move in progress
CAMVM.STOP stops motion

3.11.9 Other Variables

CAMVM.POSITION gives the value of the internal counter
CAMVM.MOVING indicates whether a move is in progress
The virtual encoder is used as the input to the cam, either alone (as a virtual master) or in combination with the actual encoder (EXTENCODER.POSITION), to add an offset to the master position. This functionality is controlled by the variable, CAM.MASTER.

4 Quick Reference: Parameters, Functions, Operators

This section contains the functions, parameters, statements and variables available for AKD BASIC. The tables below list all the parameters that can be called in the program view.

4.1 AKD Parameters and Commands

This table lists the parameters and commands from the AKD Base Drive documentation that are available in AKD BASIC. For detailed descriptions of each parameter click the links below or visit Appendix A of the AKD User Manual.

Parameter or Command	Type	Description
Analog Input (AIN)		
AIN.CUTOFF	NV	Sets the analog input low-pass filter cutoff frequency.
AIN.DEADBAND	NV	Sets the analog input signal deadband.
AIN.DEADBANDMODE	NV	Sets the analog input deadband mode.
AIN.ISCALE	NV	Sets the analog current scale factor.
AIN.MODE	NV	Analog input mode
AIN.OFFSET	NV	Sets the analog input offset.
AIN.PSCALE	NV	Sets the analog position scale factor.
AIN.VALUE	R/O	Reads the value of the analog input signal.
AIN.VSCALE	NV	Sets analog velocity scale factor.
AIN.ZERO	Command	Zeroes the analog input signal.
Analog Output (AOUT)		
AOUT.CUTOFF	NV	Sets the analog output low-pass filter cutoff frequency.
AOUT.ISCALE	NV	Sets the analog current scale factor.
AOUT.MODE	NV	Sets the analog output mode.
AOUT.OFFSET	NV	Sets the analog input offset.
AOUT.PSCALE	NV	Sets the analog position scale factor.
AOUT.VALUE	NV	Reads the analog output value.
AOUT.VALUEU	R/W	Sets the analog output value.
AOUT.VSCALE	NV	Sets the velocity scale factor for analog output.
Capture (CAP)		
CAP0.EDGE, CAP1.EDGE	NV	Selects the capture edge.
CAP0.EN, CAP1.EN	NV	Enables or disables the related capture engine.
CAP0.EVENT, CAP1.EVENT	NV	Controls the precondition logic.
CAP0.FILTER, CAP1.FILTER	R/W	Controls the precondition logic.
CAP0.MODE, CAP1.MODE	NV	Selects the captured value.
CAP0.PLFB, CAP1.PLFB	R/O	Reads captured position value.
CAP0.PREEDGE, CAP1.PREEDGE	NV	Selects the capture precondition edge.

Parameter or Command	Type	Description
CAP0.PREFILTER, CAP1.PREFILTER	NV	Sets the filter for the precondition input source.
CAP0.PRESELECT, CAP1.PRESELECT	NV	Sets the precondition trigger.
CAP0.STATE, CAP1.STATE	R/O	Indicates whether or not trigger source was captured.
CAP0.T, CAP1.T	R/O	Reads time capture (if time capture was configured).
CAP0.TRIGGER, CAP1.TRIGGER	NV	Specifies the trigger source for the position capture.
Controlled Stop (CS)		
CS.DEC	NV	Sets the deceleration value for the controlled stop process.
CS.STATE	NV	Returns the internal status of the controlled stop process.
CS.TO	NV	Sets the time value for the drive velocity to be within CS.VTHRESH .
CS.VTHRESH	NV	Sets the velocity threshold for the controlled stop.
Digital Input (DIN)		
DIN.ROTARY	R/O	Reads the rotary knob value.
DIN.STATES	R/O	Reads the digital input states.
DIN1.FILTER TO DIN7.FILTER	R/W	Filter mode for digital inputs 1 to 7.
DIO9.INV to DIO11.INV	R/W	DIO9.INV to DIO11.INV
DIN1.MODE to DIN19.MODE	NV	Sets the digital input modes.
DIN1.STATE TO DIN7.STATE	R/O	Reads a specific digital input state.
DIN9.STATE to DIN11.STATE	NV	Shows on selected pin if signal is high or low.
DIO		
DIO9.INV to DIO11.INV	NV	Inverting the output voltage of the IO, when in the output direction.
DIO9.DIR to DIO11.DIR	NV	Changing direction of the IOs from the X9 connector.
Digital Output (DOUT)		
DOUT.RELAYMODE	R/W	Indicates faults relay mode.
DOUT.STATES	R/O	Reads the state of the two digital outputs.
DOUT8.MODE to DOUT11.MODE	NV	Sets the digital output mode.
DOUT1.PARAM AND DOUT2.PARAM	NV	Sets extra parameters for the digital outputs.
DOUT1.STATE AND DOUT2.STATE	R/O	Reads the digital output state.
DOUT1.STATEU AND DOUT2.STATEU	R/W	Sets the state of the digital output node.
DOUT9.STATE to DOUT11.STATE	NV	Shows on selected pin if signal is high or low.
DOUT9.STATEU to DOUT11.STATEU	NV	Allows user to set level of selected pin to high or low.
Drive (DRV)		
DRV.ACC	NV	Describes the acceleration ramp for the velocity loop.

Parameter or Command	Type	Description
DRV.ACTIVE	R/O	Reads the enable status of an axis.
DRV.BLINKDISPLAY	Command	Causes the display to blink for 10 seconds.
DRV.CLRFAULTHIST	Command	Clears the fault history log in the NV.
DRV.CLRFAULTS	Command	Tries to clear all active faults in the drive.
DRV.CMDSOURCE	NV	Sets the command source (service, fieldbus, analog input, gearing, digital, or Bode).
DRV.DBILIMIT	NV	Sets the maximum amplitude of the current for dynamic braking.
DRV.DEC	NV	Sets the deceleration value for the velocity loop.
DRV.DIR	R/W	Changes drive direction.
DRV.DIS	Command	Disables the axis (software).
DRV.DISSOURCES	R/O	Returns the possible reason for a drive disable.
DRV.DISTO	R/W	Sets the emergency timeout.
DRV.EMUEDIR	R/W	Sets the direction of the emulated encoder output (EEO) signal.
DRV.EMUEMODE	R/W	Sets the mode of the emulated encoder output (EEO) connector.
DRV.EMUEMTURN	R/W	Defines the location of the index pulse on the EEO (emulated encoder output) when DRV.EMUEMODE=2.
DRV.EMUERES	R/W	Sets the resolution of the EEO (emulated encoder output).
DRV.EMUEZOFFSET	R/W	Sets the location of the EEO (emulated encoder output) index pulse (when DRV.EMUEMODE=1).
DRV.EN	Command	Enables the axis (software).
DRV.FAULT1 to DRV.FAULT10	R/O	Location of fault codes for any active fault conditions.
DRV.HANDWHEEL	R/O	Reads the EEO input value.
DRV.HANDWHEELSRC	NV	Selects the feedback for handwheel operation.
DRV.HWENABLE	R/O	Status of the hardware enable .
DRV.ICONT	R/O	Reads the continuous rated current value.
DRV.IPEAK	R/O	Reads the peak rated current value.
DRV.NAME	NV	Sets and reads the name of the drive.
DRV.NVLOAD	W/O	Loads all data from the NV memory of the drive into the RAM parameters.
DRV.NVSAVE	Command	Saves the drive parameters from the RAM to the NV memory.
DRV.OPMODE	NV	Sets the drive operation mode (current, velocity, or position).
DRV.RSTVAR	Command	Sets default values in the drive without re-booting the drive and without resetting the NV memory.
DRV.SETUPREQBITS	R/O	Reads the bitwise set status of parameters that must be set before the drive can be enabled
DRV.STOP	Command	This command stops all drive motion.
DRV.TIME	R/W	A continuous time counter in the drive.
DRV.WARNING1 to DRV.WARNING10	R/O	Location of fault codes for any active warning conditions.
Fault (FAULT)		
FAULTx.ACTION	R/W	Gets/Sets the Fault Action for Fault 130, 131, 132, 134, 139, 451, and 702.
Feedback 1 (FB1)		
FB1.BISSBITS	NV	Specifies the number of Biss Sensor (Position) Bits for the BiSS Mode C encoder in use.

Parameter or Command	Type	Description
FB1.ENCRES	NV	Sets the resolution of the motor encoder.
FB1.HALLSTATE	R/O	Reads the Hall switch values (encoder feedback
FB1.HALLSTATEU	R/O	Reads the state of Hall switch U.
FB1.HALLSTATEV	R/O	Reads the state of Hall switch V.
FB1.HALLSTATEW	R/O	Reads the state of Hall switch W.
FB1.IDENTIFIED	R/O	Reads the type of feedback device used by the drive/motor.
FB1.INITSIGNED	NV	Sets initial feedback value as signed or unsigned.
FB1.MECHPOS	R/O	Reads the mechanical position.
FB1.MEMVER	R/O	Returns the memory feedback version.
FB1.ORIGIN	NV	Adds to the initial feedback position.
FB1.P	R/O	Reads position from the primary feedback.
FB1.PDIR	NV	Sets the counting direction for feedback channel 1.
FB1.POFFSET	NV	FB1.POFFSET
FB1.POLES	R/O	Reads the number of feedback poles.
FB1.PSCALE	R/W	Sets position scaling value for fieldbus transferred position objects.
FB1.PUNIT	NV	Sets the unit for FB1.P.
FB1.SELECT	NV	Sets user entered type or identified type (–1).
Feedback 2 (FB2)		
FB2P.DIR	R/W	FB2P.DIR
FB2.ENCRES	NV	Sets the secondary feedback (FB2) resolution.
FB2.MODE	R/W	Sets the mode for the second feedback inputs, EEO connector (X9) and high speed opto inputs (pins 9 and 10 on X7).
FB2.P	R/O	Reads position from the secondary feedback.
FB2P.DIR	NV	Sets the counting direction for feedback channel 2.
FB2.POFFSET	NV	Sets the offset for secondary feedback.
FB2.PUNIT	NV	Sets the unit for FB2.P.
FB2.SOURCE	R/W	Sets the source for the second feedback input. Choices are the EEO connectors (X9) which are RS485 inputs, or the X7 connector's high speed opto inputs (pins 9 and 10).
Feedback 3 (FB3)		
FB3.MODE	NV	Selects the type of feedback connected to X9.
FB3.POFFSET	NV	Sets the offset for tertiary feedback.
FB3.PUNIT	NV	Sets the unit for FB3.P.
Hardware Limit Switch (HWLS)		
HWLS.NEGSTATE	R/O	Reads the status of the negative hardware limit switch.
HWLS.POSSTATE	R/O	Reads the status of the positive hardware limit switch.
Current Loop (IL)		
IL.BUSFF	R/O	Displays the current feedforward value injected by the fieldbus.
IL.CMD	R/O	Reads the value of the q-component current command.
IL.CMDU	R/W	Sets the user current command.
IL.DIFOLD	R/O	Reads the drive foldback current limit.
IL.FB	R/O	Reads the actual value of the d-component current.
IL.FF	R/O	Displays the current loop overall feedforward value.
IL.FOLDFTHRESH	NV	Reads the foldback fault level.

Parameter or Command	Type	Description
IL.FOLDWTHRESH	NV	Sets the foldback warning level.
IL.IFOLD	R/O	Reads the overall foldback current limit.
IL.IUFB	R/O	Reads the sigma-delta measured current in the u-winding of the motor.
IL.IVFB	R/W	Sets the sigma-delta measured current in the u-winding of the motor.
IL.KP	NV	Sets the proportional gain of the q-component of the PI regulator.
IL.KPDRATIO	NV	Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP
IL.LIMITN	NV	Sets the negative user (application-specific) current limit.
IL.LIMITP	NV	Sets the positive user (application-specific) current limit.
IL.MFOLDD	NV	Sets the motor foldback maximum time at motor peak current.
IL.MFOLDR	R/O	Sets the motor foldback recovery time.
IL.MFOLDT	NV	Sets the motor foldback time constant of the exponential current drop (foldback).
IL.MIFOLD	R/O	Sets the motor foldback current limit.
IL.VCMD	R/O	Sets the output of the q-component PI regulator.
IL.VUFB	R/O	Reads the measured voltage on the u-winding of the motor.
IL.VVFB	R/O	Reads the measured voltage on the v-winding of the motor.
LOAD Parameters		
LOAD.INERTIA	NV	Sets the load inertia.

Motor Parameters		
MOTOR.BRAKE	NV	Sets the presence or absence of a motor brake.
MOTOR.BRAKERLS	Command	Allows a user to release the motor brake.
MOTOR.ICONT	NV	Sets the motor continuous current.
MOTOR.INERTIA	NV	Sets the motor inertia.
MOTOR.IPEAK	NV	Sets the motor peak current.
MOTOR.KE	NV	Sets the motor back EMF constant.
MOTOR.KT	NV	Sets the torque constant of the motor.
MOTOR.LQLL	NV	Sets the line-to-line motor Lq.
MOTOR.NAME	NV	Sets the motor name.
MOTOR.PHASE	NV	Sets the motor phase.
MOTOR.PITCH	NV	Sets the motor pitch.
MOTOR.POLES	NV	Sets the number of motor poles.
MOTOR.R	NV	Sets the stator winding resistance phase-phase in ohms.
MOTOR.TBRAKEAPP	NV	The delay time used for applying the motor brake.
MOTOR.TBRAKERLS	NV	The delay time used for releasing the motor brake.
MOTOR.TEMP	R/O	Reads the motor temperature represented as the resistance of the motor PTC.
MOTOR.TEMPFAULT	NV	Sets the motor temperature fault level.
MOTOR.TEMPWARN	NV	Sets the motor temperature warning level.
MOTOR.TYPE	NV	Sets the motor type.
MOTOR.VOLTMAX	NV	Sets the motor maximum voltage.
Position Loop (PL)		
PL.CMD	NV	Reads the position command directly from the entry to the position loop.

PL.ERR	NV	Reads the position error present when the drive is controlling the position loop.
PL.ERRFTHRESH	NV	Sets the maximum position error.
PL.ERRMODE	R/W	Sets the type of following error warning and fault usage.
PL.ERRWTHRESH	NV	Sets the position error warning level.
PL.FB	R/O	Reads the position feedback value.
PL.FBSOURCE	NV	Sets the feedback source for the position loop.
PL.INTINMAX	NV	Limits the input of the position loop integrator by setting the input saturation.
PL.INTOUTMAX	NV	Limits the output of the position loop integrator by setting the output saturation.
PL.KI	NV	Sets the integral gain of the position loop.
PL.KP	NV	Sets the proportional gain of the position regulator PID loop.
PL.MODP1	R/W	Sets modulo range parameter.
PL.MODP2	R/W	Sets the beginning or end modulo range parameter.
Programmable Limit Switch (PLS)		
PLS.EN	R/W	Enables programmable limit switch (PLS).
PLS.MODE	NV	Selects programmable limit switch mode.
PLS.P1 TO PLS.P8	NV	Sets the trigger point for programmable limit switches.
PLS.RESET	W/O	Resets programmable limit switch.
PLS.STATE	R/O	Reads the programmable limit switch state.
PLS.T1 TO PLS.T8	R/W	Sets programmable limit switch time.
PLS.UNITS	R/W	Sets programmable limit switch (PLS) units.
PLS.WIDTH1 TO PLS.WIDTH8	R/W	Programmable Limit Switch Width
Recorder (REC)		
REC.ACTIVE	R/O	Indicates if data recording is in progress (active).
REC.DONE	R/O	Checks whether or not the recorder has finished recording.
REC.OFF	R/W	Turns the recorder OFF.
REC.TRIG	Command	Triggers the recorder.
Regen Resistor (REGEN)		
REGEN.POWER	R/O	READS REGEN RESISTOR'S CALCULATED POWER.
REGEN.REXT	N/V	SETS THE EXTERNAL, USER-DEFINED REGEN RESISTOR RESISTANCE.
REGEN.TEXT	R/W	Sets the external regen resistor thermal protection time constant.
REGEN.TYPE	N/V	SETS THE REGEN RESISTOR TYPE.
REGEN.WATTEXT	R/W	SETS THE REGEN RESISTOR'S POWER FAULT LEVEL FOR AN EXTERNAL REGEN RESISTOR.
STO		
STO.STATE	R/O	Returns the status of the safe torque off.
SWLS		
SWLS.EN	NV	Enables and disables software travel limit switches.
SWLS.LIMIT0	NV	Sets the position of the software travel limit switch 0.
SWLS.LIMIT1	NV	Sets the position of the software travel limit switch 0.

SWLS.STATE	R/O	Reads the actual status of software limit switches.
Units (UNIT)		
UNIT.ACCLINEAR	NV	Sets the linear acceleration/deceleration units.
UNIT.ACCROTARY	NV	Sets the rotary acceleration/deceleration units.
UNIT.LABEL	NV	Sets user-defined name for user-defined position units.
UNIT.PIN	NV	Sets gear IN for the unit conversion.
UNIT.PLINEAR	NV	Sets the linear position units.
UNIT.POUT	NV	Sets gear out for the unit conversion.
UNIT.PROTARY	NV	Sets the position units when the motor type (MOTOR.TYPE) is rotary.
UNIT.VLINEAR	NV	Sets the linear velocity units.
UNIT.VROTARY	NV	Sets the velocity units when the motor type (MOTOR.TYPE) is rotary.
Bus Voltage (VBUS)		
VBUS.OVFTHRESH	R/O	Reads the over voltage fault level.
VBUS.OVWTHRESH	NV	Sets voltage level for over voltage warning.
VBUS.RMSLIMIT	R/O	Reads the limit for the bus capacitors load.
VBUS.UVFTHRESH	R/W	Sets the under voltage fault level.
VBUS.UVMODE	NV	Indicates undervoltage (UV) mode.
VBUS.UVWTHRESH	NV	Sets voltage level for undervoltage warning.
VBUS.VALUE	R/O	Reads DC bus voltage.
Velocity Loop (VL)		
VL.ARPF1 TO VL.ARPF4	R/W	Sets the natural frequency of the pole (denominator) of anti-resonance (AR) filters 1, 2, 3, and 4; active in opmodes 1 (velocity) and 2 (position) only.
VL.ARPQ1 TO VL.ARPQ4	R/W	Sets the Q of the pole (denominator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only.
VL.ARTYPE1 TO VL.ARTYPE4	NV	Indicates the method used to calculate BiQuad coefficients; active in opmodes 1 (velocity) and 2 (position) only.
VL.ARZF1 TO VL.ARZF4	R/W	Sets the natural frequency of the zero (numerator) of anti-resonance (AR)filter 1; active in opmodes 1 (velocity) and 2 (position) only.
VL.ARZQ1 TO VL.ARZQ4	R/W	Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only.
VL.BUSFF	R/O	Displays the velocity loop feedforward value injected by the field-bus; active in opmodes 1 (velocity) and 2 (position) only.
VL.CMD	R/O	Reads the actual velocity command; active in opmodes 1 (velocity) and 2 (position) only.
VL.CMDU	R/W	Sets the user velocity command; active in opmodes 1 (velocity) and 2 (position) only.
VL.ERR	R/O	Sets the velocity error; active in opmodes 1 (velocity) and 2 (position) only.
VL.FB	R/O	Reads the velocity feedback; active in opmodes 1 (velocity) and 2 (position) only.
VL.FBFILTER	R/O	Filters VL.FB value; active in opmodes 1 (velocity) and 2 (position) only.
VL.FBSOURCE	NV	Sets feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only.
VL.FBUNFILTERED	R/O	Reads the velocity feedback.

VL.FF	R/O	Displays the velocity loop overall feedforward value; active in opmodes 1 (velocity) and 2 (position) only.
VL.GENMODE	NV	Selects mode of velocity generation (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only.
VL.KBUSFF	R/W	Sets the velocity loop acceleration feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only.
VL.KI	NV	Sets the velocity loop integral gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only.
VL.KP	NV	Sets velocity loop proportional gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only.
VL.KVFF	R/W	Sets the velocity loop velocity feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only.
VL.LIMITN	NV	Sets the velocity lower limit; active in opmodes 1 (velocity) and 2 (position) only.
VL.LIMITP	NV	Sets the velocity high limit; active in opmodes 1 (velocity) and 2 (position) only.
VL.LMJR	R/W	Sets the ratio of the estimated load moment of inertia relative to the motor moment of inertia; active in opmodes 1 (velocity) and 2 (position) only.
VL.THRESH	NV	Sets the over speed fault value; active in opmodes 1 (velocity) and 2 (position) only.
Wake and Shake (WS)		
WS.ARM	Command	Sets wake and shake to start at the next drive enable.
WS.DISARM	Command	Cancels ARM requests and resets wake and shake to the IDLE state.
WS.DISTMAX	R/W	Sets maximum movement allowed for wake and shake.
WS.DISTMIN	R/W	Sets the minimum movement required for wake and shake.
WS.IMAX	R/W	Sets maximum current used for wake and shake.
WS.MODE	R/W	Sets the method used for wake and shake.
WS.NUMLOOPS	R/W	Sets the number of repetitions for wake and shake.
WS.STATE	R/O	Reads wake and shake status
WS.T	R/W	Sets wake and shake current-vector appliance time
WS.TDELAY1	NV	Delay for wake and shake timing
WS.TDELAY2	NV	Sets the delay for wake and shake timing.
WS.TDELAY3	NV	Sets the delay for wake and shake between loops in mode 0.
WS.VTHRESH	NV	Defines the maximum allowed velocity for Wake & Shake

4.2 AKD BASIC Parameters and Commands

The following table lists parameters, commands, functions, and operators unique to AKD BASIC. For detailed descriptions of each parameter click the links below or visit the AKD BASIC Parameters chapter of the AKD BASIC User Manual.

Parameter	Type	Description
Additional Statements		
$Include	Statement	The $Include statement allows you to textually include multiple separate files in a single source file.
Alias	Statement	Allows you to define your own names for system resources, such as Input or Output pins.

Parameter	Type	Description
Call	Statement	Transfer program control to a subroutine. When the subroutine is finished then control is transferred to the line following the CALL.
Cls	Statement	This statement transmits 40 line feed characters (ASCII code = 10) to the serial port. Cls clears the display of a terminal.
Const	Statement	Declares symbolic constants to be used instead of numeric values.
Dim	Statement	Used for declaring variables before use. All variables (except predefined variables) must be declared before they can be used.
Exit	Statement	The Exit statement is used to exit from a subroutine, a function, an interrupt, a For...Next or a While...Wend.
For...Next	Statement	Allows a series of lines to be executed in a loop a specified number of times.
Function	Statement	The Function statement is used to declare and define the name, arguments and type of a user defined function.
GoTo	Statement	GOTO causes the software to jump to the specified label and continue executing from there.
If...Then...Else	Statement	If...Then...Else statements controls program execution based on the evaluation of numeric or string expressions.
Input	Statement	The Input statement reads a character string received from the console tab screen below the program editor window, terminated by a carriage-return.
On Error GoTo	Statement	On Error Goto allows you to define a run-time error handler to prevent run-time errors from halting program execution.
Pause()	Statement	Causes the program execution to pause for a specified amount of time. The motion of the motor is not affected.
Print	Statement	Displays formatted output through the console while the program is running.
Restart	Statement	Causes program execution to begin again from the beginning of the program.
Select Case	Statement	Select Case executes one of several statement blocks depending upon the value of an expression.
Static	Statement	Static
Stop	Statement	Stops execution of the user program.
Sub...End Sub	Statement	The Sub statement declares a sub procedure and defines the sub procedures format.
Swap	Statement	Exchanges the value of two variables.
While...Wend	Statement	Executes a series of lines for as long as the condition after the WHILE is True.
BASIC Operators		
MOD	Operator	This is the modulus or "remainder" operator. It divides one number by another and returns the remainder.
BASIC Functions		
ABS()	Function	ABS(x) converts the associated value (x) to an absolute value. If the value is negative, it is converted to a positive value. If the value is positive, it is not changed.

Parameter	Type	Description
ASC()	Function	ASC(string expression)returns a decimal numeric value that is the ASCII code for the first character of the string expression(x$).
ATAN()	Function	ATAN() (arc tangent) returns the arctangent of its argument in radians.
CHR$()	Function	Returns a one character string whose ASCII value is the argument.
CINT()	Function	Converts a numeric expression to the closest integer number.
COS()	Function	COS(x) returns the cosine of x, where x is in radians.
EXP()	Function	Returns e (the base of natural logarithms) raised to a power.
FIX()	Function	Fix() returns the truncated integer part of x.
HEX$()	Function	HEX$() converts an integer number to its equivalent hexadecimal ASCII string.
INKEY$()	Function	Returns a 1 character string corresponding to the character in the serial port receive buffer. If there is no character waiting the INKEY$ will be the Null string (""). If several characters are pending only the first one is returned.
INSTR()	Function	Returns the starting location of a substring within a string.
INT()	Function	INT() (convert to largest integer) truncates an expression to a whole number.
LCASE$()	Function	Converts a string expression to lowercase characters.
LEFT$()	Function	Returns a string of the n leftmost characters in a string expression.
LEN()	Function	Returns the number of characters in a string expression.
LOG()	Function	Returns the natural logarithm of a numeric expression.
LOG10()	Function	Returns the base 10 logarithm of a numeric expression.
LTRIM$()	Function	Returns a copy of the original string with leading blanks removed.
MID$	Function	Returns a substring of the original string that begins at the specified offset location and is of the specified (optional) length.
OCT$()	Function	OCT$() converts an integer number to its equivalent octal ASCII string.
RIGHT$()	Function	Returns a string of the n rightmost characters in a string expression.
RTRIM$()	Function	Returns a copy of the original string with trailing blanks removed.
SGN()	Function	Returns the sign of a numeric expression.
SIN()	Function	SIN(x) returns the sine of x, where x is in radians.
SPACE$()	Function	Returns a string of n spaces.
SQR()	Function	Returns the square root of a numeric expression.
STR$()	Function	Returns a string representing the value of a numeric expression.
STRING$()	Function	Returns a string containing the specified number of occurrences of the specified character.

Parameter	Type	Description
TAN()	Function	TAN(x) returns the tangent of x, where x is in radians.
TRIM$()	Function	Returns a copy of the original string with leading and trailing blanks removed.
UCASE$()	Function	Converts a string expression to uppercase characters.
VAL()	Function	Returns the numerical value of a string.
Camming (CAM)		
CAM.ACTIVATE	Statement	Activates the specified cam table.
CAM.ADDPOINT	Statement	CAM.ADDPOINT(Master Position, Slave Position) Adds the specified "point" to the cam table being created.
CAM.CORRECTDIR	R/W	Specifies the direction of the correction move when a new cam table is activated (set CAM.ACTIVATE = n) or when speed synchronization is achieved.
CAM.CREATE	Statement	CAM.CREATE (x, y) Initiates the creation of a cam table.
CAM.MASTER	R/W	Specifies the source of the input to the cam table for cam profiling.
CAM.MASTERPOS	R/O	Gives the value of the master position presently being used as the input to the cam table.
CAM.SLAVEOFFSET	R/O	CAM.SLAVEOFFSET indicates the offset (or difference) between MOVE.POSCOMMAND and the position command that is calculated from the active cam table based upon the present value of EXTENCODER.POSITION and/or CAMVM.POSITION.
CAMVM.DIR	R/W	Specifies the direction the virtual encoder goes when CAMVM.GOVEL is executed.
CAMVM.FREQ	R/W	CAMVM.FREQ sets the maximum frequency allowed during a relative (CAMVM.GOREL) move, and sets the commanded speed during a velocity move (CAMVM.GOVEL)
CAMVM.GOREL	Statement	Makes the virtual master move the distance specified by CAMVM.RELATIVEDIST.
CAMVM.GOUPDATE	Statement	Updates a move in progress with new move parameters.
CAMVM.GOVEL	Statement	CAMVM.GOVEL (Go at Velocity) causes the virtual master to move continuously at the frequency specified by CAMVM.FREQ in the direction (positive or negative) specified by CAMVM.DIR. The frequency or direction is modified during the move using CAMVM.GOUPDATE.
CAMVM.MOVING	R/O	Indicates if the virtual encoder is moving.
CAMVM.POSITION	R/W	Contains the current value of the virtual encoder counter.
CAMVM.RELATIVEDIST	R/W	Specifies the number or counts that the virtual encoder (virtual master) will put out during an incremental move (CAMVM.GOREL).
CAMVM.STOP	Statement	CAMVM.STOP stops the virtual encoder
Drive (DRV)		
DRV.SWENABLE	R/W	Controls whether power can flow to the motor.
Electronic Gearing (EGEAR)		
EGEAR.ACCLIMIT	R/W	EGEAR.ACCLIMIT sets the maximum acceleration.

Parameter	Type	Description
EGEAR.DECLIMIT	R/W	EGEAR.DECLIMIT sets the maximum deceleration that will be commanded on the follower when EGEAR.ON is turned OFF or the electronic gearing ratio (EGEAR.RATIO or EGEAR.PULSESOUT / EGEAR.PUSLESIN) is decreased.
EGEAR.ERROR	R/W	Indicates the amount of position deviation that has accumulated on the slave axis (in an electronic gearing application) as a result of the slave axis limiting its acceleration or deceleration while achieving velocity synchronization.
EGEAR.LOCK	R/O	EGEAR.LOCK indicates when the slave axis (follower axis) in an electronic gearing application has achieved velocity synchronization with the electronic gearing master.
EGEAR.ON	Command	Starts the electronic gearing; active in opmode 2 (position) only.
EGEAR.PULSESIN	R/W	Specifies the number of encoder counts used when specifying an exact electronic gearing ratio.
EGEAR.PULSEOUT	R/W	Specifies the number of position counts used in an exact electronic gearing ratio.
EGEAR.RATIO	R/W	Sets the electronic gearing ratio (rev to rev) between the encoder shaft (master) and the motor shaft (slave).
EGEAR.TYPE	R/W	Sets the allowed direction of motion for electronic gearing.
External Encoder (EXTENCODER)		
EXTENCODER.FREQ	R/O	Gets the external encoder (EEO) velocity.
EXTENCODER.POSITION	R/O	Gets the external encoder (EEO) position.
EXTENCODER.POSMODULO	R/W	Sets/gets the external encoder (EEO) modulo position.
Interrupt (INTR)		
Interrupt...End Interrupt	Statement	The interrupt feature permits execution of a user-defined subroutine upon receipt of a hardware interrupt signal or a pre-defined interrupt event.
INTR.DIN1HI		Enables interrupt for when DIN1.STATE to DIN7.STATE goes from 0 to 1, respectively.
INTR.DIN1LO		Enables interrupt for when DIN1.STATE to DIN7.STATE goes from 1 to 0, respectively.
INTR.DISABLE		Enables interrupt for when the drive gets disabled.
INTR.DRV.FAULTS		Enables interrupt for when the drive faults.
INTR.DRV.HWENABLE		Enables interrupt for when DRV.HWENABLE goes from 0 to 1.
INTR.DRV.WARNINGS		Enables interrupt for when the drive produces a warning.
INTR.HWLS.NEGSTATE		Enables interrupt for when HWLS.NEGSTATE goes from 0 to 1.
INTR.HWLS.POSSTATE		Enables interrupt for when HWLS.POSSTATE goes from 0 to 1.
INTR.MODBUS		Enables interrupt for when a Modbus User Parameter changes.
INTR.PL.ERR		Enables interrupt for when PL.ERR = PL.ERRFTHRESH.

Parameter	Type	Description
INTR.PLS.P1 to INTR.PLS.P8		Enables interrupt for when PLS1 to PLS8 is enabled and goes high, respectively.
INTR.SWLS.LIMIT0		Enables interrupt for when PL.FB > SWLS.LIMIT0 (if SWLS.LIMIT0 is the upper limit)
INTR.SWLS.LIMIT1		Enables interrupt for when PL.FB < SWLS.LIMIT1 (if SWLS.LIMIT1 is the lower limit)
INTR.TIMER		Executes interrupt after a number of milliseconds specified by VM.INTRTIMER.
MODBUS Parameters		
MODBUS.READFLOAT	R/W	This function reads a floating-point value from the specified ModBus slave and returns the value read.
MODBUS.WRITEFLOAT	R/W	This statement writes a floating-point value to the specified ModBus slave.
Move Parameters (MOVE)		
MOVE.ABORT	Command	MOVE.ABORT stops motor motion and allows continued program execution.
MOVE.ACC	R/W	Sets the maximum commanded acceleration rate when the speed is increased.
MOVE.DEC	R/W	Sets the maximum commanded deceleration rate when the speed is decreased.
MOVE.DIR	R/W	MOVE.DIR specifies the direction the motor turns when a MOVE.GOVEL statement is executed.
MOVE.GOABS	Command	MOVE.GOABS moves the motor to the position specified by MOVE.TARGETPOS.
MOVE.GOHOME	Command	MOVE.GOHOME causes the motor to move to the position specified where PL.FB = 0.
MOVE.GOREL	Command	MOVE.GOREL moves the motor a distance specified by MOVE.RELATIVEDIST.
MOVE.GOUPDATE	Command	MOVE.GOUPDATE updates a move in progress with new move parameters.
MOVE.GOVEL	Command	MOVE.GOVEL moves the motor at a constant speed specified by MOVE.RUNSPEED and direction specified by MOVE.DIR.
MOVE.INPOSITION	R/O	Indicates whether or not the motor has achieved command position.
MOVE.INPOSLIMIT	R/W	Specifies the tolerance of Position Error (PL.ERR) within which the MOVE.INPOSITION flag will be set to 1 (True).
MOVE.MOVING	R/O	Indicates whether or not the commanded motion profile is complete.
MOVE.POSCOMMAND	R/W	Current Position Command from Trajectory Generator.
MOVE.RELATIVEDIST	R/W	Specifies the distance the motor turns during a relative move (MOVE.GOREL).
MOVE.RUNSPEED	R/W	Sets the maximum speed allowed during a relative (MOVE.GOREL) or absolute (MOVE.GOABS) move, and sets the commanded speed during a velocity move (MOVE.GOVEL).
MOVE.SCURVETIME	R/W	Sets the amount of S-curve smoothing applied to all velocity profiles.

Parameter	Type	Description
MOVE.TARGETPOS	R/W	MOVE.TARGETPOS specifies the target position for an absolute (MOVE.GOABS) move.
Virtual Machine (VM)		
VM.AUTOSTART	R/W	VM.AUTOSTART specifies whether or not the program in the AKD BASIC starts executing automatically when AC power is applied.
VM.ERR	R/W	Indicates what caused the most recent Runtime Error.
VM.INTRTIMER	R/W	Sets a number of milliseconds before INTR.TIMER executes after it is called.
VM.RESTART	Command	Restart AKD BASIC virtual machine.
WHEN		
When	Statement	The WHEN statement is used for very fast response to certain input conditions.
When Conditons		
PL.FB < value		
PL.FB > value		
PL.CMD < value		
PL.CMD > value		
DRV.HANDWHEEL < value		
DRV.HANDWHEEL > value		
DRV.TIME > value		
DINx.STATE = 0 or 1		
FB3.P < value		
FB3.P > value		
MOVE.MOVING = 0 or 1		
MOVE.INPOSITION = 0 or 1		
When Actions		
Continue		
DOUT1.STATEU = 0 or 1		
DOUT2.STATEU = 0 or 1		
EGEAR.RATIO = value		
MOVE.ABORT		
MOVE.GOABS		
MOVE.GOREL		
MOVE.GOVEL		
MOVE.GOABSREG		
MOVE.GOHOME		
MOVE.GORELREG		
MOVE.GOUPDATE		
WHEN.DRVHANDWHEEL	R/O	Records the value of DRV.HANDWHEEL when the when-condition is satisfied.
WHEN.DRVTIME	R/O	Records the value of Time when the when-condition is satisfied.

Parameter	Type	Description
WHEN.FB1MECHPOS	R/O	Records the value of FB1.MECHPOS when the when-condition is satisfied.
WHEN.PLCMD	R/O	Records the value of PL.CMD when the when-condition is satisfied.
WHEN.PLFB	R/O	Records the value of Position when the when-condition is satisfied.

5 AKD BASIC Functions

This section describes the functions available in AKD BASIC.

5.1 ABS()

General Information	
Type	Function
Description	ABS(x) converts the associated value (x) to an absolute value. If the value is negative, it is converted to a positive value. If the value is positive, it is not changed.

Instructions

Enter the argument (the value) in parentheses immediately following the term ABS.

Example

```
For x = -10 To 10
       Print          ABS(x)
Next
```

5.2 ASC()

General Information	
Type	Function
Description	ASC(string expression)returns a decimal numeric value that is the ASCII code for the first character of the string expression(x$).

Syntax

x = ASC(s$)

Instructions

If the string begins with an uppercase letter, the value of ASC() will be between 65 and 90.

If the string begins with a lowercase letter, the value of ASC() will be between 97 and 122.

Values "0" to "9" return 48 to 57.

5.3 ATAN()

General Information	
Type	Function
Description	ATAN() (arc tangent) returns the arctangent of its argument in radians.

Instructions

The result is always between -pi/2 and pi/2.

The value of x may be any numeric type.

To convert from degrees to radians, multiply by 0.01745329

5.4 CHR$()

General Information	
Type	Function
Description	Returns a one character string whose ASCII value is the argument.

Syntax

s$ = CHR$(x)

Instructions

The argument to Chr$() must be a numeric value in the range 0 to 255.

Example

```
'this example will print an uppercase B
Dim a$ as string
a$ = CHR$(66)
Print a$
```

5.5 CINT()

General Information	
Type	Function
Description	Converts a numeric expression to the closest integer number.

Instructions

X = CINT(numeric-expression)

Related Topics

INT() | FIX()

5.6 COS()

General Information	
Type	Function
Description	COS(x) returns the cosine of x, where x is in radians.

Instructions

X must be in radians. To convert from degrees to radians, multiply by 0.017453.

5.7 EXP()

General Information	
Type	Function
Description	Returns *e* (the base of natural logarithms) raised to a power.

Instructions

The Exp() function complements the action of the Log() function. For those of you keeping score, the value of *e* is 2.71828182.

Related Topics

LOG() | LOG10()

5.8 FIX()

General Information	
Type	Function
Description	Fix() returns the truncated integer part of x.

Instructions

Fix() does not round off numbers, it simply eliminates the decimal pint and all digits to the right of the decimal point.

Related Topics

ABS() | CINT() | INT()

5.9 HEX$()

General Information	
Type	Function
Description	HEX$() converts an integer number to its equivalent hexadecimal ASCII string.

Syntax

result$ = HEX$(x)

Instructions

Hexadecimal numbers are numbers to the base 16 (rather than base 10).

The argument to HEX$() is rounded to an integer before HEX$(x) is evaluated.

Example

```
Dim x,y as integer
Dim result1$, result2$ as string
x = 20
y = $H6A
result1$ = HEX$(x)
result2$ = HEX$(y)
Print result1$, result2$
'prints 14 6A
```

Related Topics

OCT$() | STR$()

5.10 INKEY$()

General Information	
Type	Function
Description	Returns a 1 character string corresponding to the character in the serial port receive buffer. If there is no character waiting the INKEY$ will be the Null string (""). If several characters are pending only the first one is returned.

Syntax

x$ = INKEY$

Instructions

Assigning a string from INKEY$ removes the character from the serial port's receive buffer.

Example

```
' remove all characters from the receive buffer
and put them into A$.
'--------------------------------------
new$ = INKEY$
While new$ <> ""
       A$ = A$ + new$
       new$ = INKEY$
Wend
```

5.11 INSTR()

General Information	
Type	Function
Description	Returns the starting location of a substring within a string.

Syntax

result = INSTR([n], x$, y$)

x$ is the string

y$ is the substring

n optionally sets the start of the search

Instructions

n must be in the range 1 to 255

INSTR() returns 0 if:

n > LEN(x$)

y$ cannot be found in x$

If y$ iis null (empty, ""), INSTR() returns n

Related Topics

LEN()

5.12 INT()

General Information	
Type	Function
Description	INT() (convert to largest integer) truncates an expression to a whole number.

Instructions

INT() behaves the same as FIX() for positive numbers. They behave differently for negative numbers.

Example

```
Print INT(12.34) 'prints the value 12
Print INT(-12.34) 'prints the value -13
```

Related Topics

CINT() | FIX()

5.13 LCASE$()

General Information	
Type	Function
Description	Converts a string expression to lowercase characters.

Syntax

result$ = LCASE$(string-expression)

Instructions

LCASE$() affects only letters in the string expression. Other characters (such as numbers) are not changed.

Example

```
Dim x$ as string
x$ = "U.S.A"
Print       LCASE$(x$) 'prints: u.s.a
```

Related Topics

UCASE$()

5.14 LEFT$()

General Information	
Type	Function
Description	Returns a string of the n leftmost characters in a string expression.

Syntax

result$ = LEFT$(x$, n)

Instructions

If n is greater than LEN(x$) then the entire string will be returned.

Example

```
a$ = "Mississippi"
Print         LEFT$(a$, 5)        'prints: Missi
```

Related Topics

LEN() | MID$ | RIGHT$()

5.15 LEN()

General Information	
Type	Function
Description	Returns the number of characters in a string expression.

Syntax

result = Len(x$)

Instructions

Non-printing characters and blanks are included.

Example

```
x$ = "New York, New York"
Print          LEN(x$) 'prints 18
```

5.16 LOG()

General Information	
Type	Function
Description	Returns the natural logarithm of a numeric expression.

Instructions

X must be greater than 0.

Example

```
Print         LOG (45.0 / 7.0) 'prints 1.860752

Print         LOG(1) 'prints 0
```

Related Topics

EXP() | LOG10()

5.17 LOG10()

General Information	
Type	Function
Description	Returns the base 10 logarithm of a numeric expression.

Instructions

X must be greater than 0.

Example

```
Print         LOG10(100) 'prints 2
Print         LOG10(1) 'prints 0
```

Related Topics

EXP() | LOG()

5.18 LTRIM$()

General Information	
Type	Function
Description	Returns a copy of the original string with leading blanks removed.

Syntax

result$ = LTRIM$(x$)

Instructions

x$ can be any string-expression

Example

```
x$ = "    Hello    "
Print        "(" + LTRIM$(x$) + ")"
'prints: (Hello    )
```

Related Topics

RTRIM$() | TRIM$()

5.19 MID$

General Information	
Type	Function
Description	Returns a substring of the original string that begins at the specified offset location and is of the specified (optional) length.

Syntax

result = MID$(x$, start, [length])

Instructions

Start and Length must both be numeric expressions.

If Length is omitted then MID$() returns a substring that starts at start and goes to the end of x$.

Example

```
x$ = "abcdefghi"
Print        MID$ (x$, 1, 5) 'prints: abcde
Print        MID$ (x$, 6) 'prints: fghi
```

Related Topics

INSTR() | LEFT$() | LEN() | RIGHT$()

5.20 OCT$()

General Information	
Type	Function
Description	OCT$() converts an integer number to its equivalent octal ASCII string.

Syntax

result$ = OCT$(x)

Instructions

Octal numbers are numbers to the base 8 (rather than base 10).

The argument to HEX$() is rounded to an integer before OCT$(x) is evaluated.

Example

```
Dim x, y as integer
Dim result1$, result2$ as string

x = 20
y = &H6A
result1$ = OCT$(x)
result2$ = OCT$(y)
print result1$, result2$ 'prints: 24    152
```

Related Topics

HEX$() | STR$()

5.21 RIGHT$()

General Information	
Type	Function
Description	Returns a string of the n rightmost characters in a string expression.

Syntax

result$ = RIGHT$(x$, n)

Instructions

If n is greater than Len(x$) then the entire string will be returned.

Example

```
a$ = "Mississippi"
Print         RIGHT$(a$, 5) 'prints: sippi
```

Related Topics

LEN() | MID$ | LEFT$()

5.22 RTRIM$()

General Information	
Type	Function
Description	Returns a copy of the original string with trailing blanks removed.

Syntax

result$ = RTRIM$(x$)

Instructions

x$ can be any string-expression.

Example

```
x$ = "    Hello    "
Print        "(" + RTRIM$(x$) + ")"
'prints: (    Hello)
```

Related Topics

LTRIM$() | TRIM$()

5.23 SGN()

General Information	
Type	Function
Description	Returns the sign of a numeric expression.

Instructions

X is any numeric expression

Example

```
Print     SGN(-33) 'prints -1
Print     SGN(0) 'prints 0
Print     SGN(45.77) 'prints 1
```

5.24 SIN()

General Information	
Type	Function
Description	SIN(x) returns the sine of x, where x is in radians.

Instructions

X must be in radians. To convert from degrees to radians, multiply by 0.017453.

5.25 SPACE$()

General Information	
Type	Function
Description	Returns a string of n spaces.

Syntax

result$ = SPACE$(n)

n is 0 to 255

Instructions

N is rounded to an integer before SPACE$() is evaluated.

Example

```
x$ = "(" + SPACE$(1) + "hello" + SPACE$(4) +
")"
Print x$ 'prints: ( hello    )
```

Related Topics

STRING$()

5.26 SQR()

General Information	
Type	Function
Description	Returns the square root of a numeric expression.

Instructions

X must be greater than or equal to zero.

Example

```
x = 10
print      SQR(x) 'prints 3.162278
```

5.27 STR$()

General Information	
Type	Function
Description	Returns a string representing the value of a numeric expression.

Syntax

result$ = STR$(x)

Instructions

Enter a numeric expression as x and STR$ will return its result as a string.

Example

```
x = 45.2 / 7
Print          STR$(x) 'prints: 6.457
```

Related Topics

HEX$() | OCT$()

5.28 STRING$()

General Information	
Type	Function
Description	Returns a string containing the specified number of occurrences of the specified character.

Syntax

1) x$ = STRING$(n, a$)

or

2) x$ = STRING$(n, m)

Instructions

n is the number of occurrences of the desired character (the length of the returned string).

In 1), the returned string will consist of the first character in a$

In 2), the returned string will consist of the ASCII value of m.

Example

```
Print       String$(5, 45) 'prints: -----
Print       String$(5, "A") 'prints: AAAAA
```

Related Topics

SPACE$()

5.29 TAN()

General Information	
Type	Function
Description	TAN(x) returns the tangent of x, where x is in radians.

Instructions

X must be in radians. To convert from degrees to radians, multiply by 0.017453.

5.30 TRIM$()

General Information	
Type	Function
Description	Returns a copy of the original string with leading and trailing blanks removed.

Syntax

result$ = TRIM$(x$)

Instructions

x$ can be any string-expression

Example

```
x$ = "      Hello      "
Print         "(" + TRIM$(x$) + ")"
'prints:   (Hello)
```

Related Topics

LTRIM$() | RTRIM$()

5.31 UCASE$()

General Information	
Type	Function
Description	Converts a string expression to uppercase characters.

Syntax

result$ = UCASE$(string-expression)

Instructions

UCASE$() affects only letters in the string expression. Other characters (such as numbers) are not changed.

Example

```
Dim x$ as string
x$ = "u.s.a"
Print       UCASE$( x$) 'prints:  U.S.A
```

Related Topics

LCASE$()

5.32 VAL()

General Information	
Type	Function
Description	Returns the numerical value of a string.

Syntax

result = VAL(a$)

Instructions

If the first character of a$ is not numeric then Val() will return 0.

Related Topics

STR$()

6 AKD BASIC Parameters, Operators, Statements

This section is an alphabetical reference to AKD BASIC parameters.

General information for each parameter is listed at the top of each page. The parameter is then described, and in some cases examples are given along with references to related parameters.

6.1 Additional Statements

This section describes statements not directly related to a parameter set.

6.1.1 $Include

General Information	
Type	Statement
Description	The $Include statement allows you to textually include multiple separate files in a single source file.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The $Include statement allows you to textually include multiple separate files in a single source file. A file cannot include itself, either directly or indirectly. Include file nesting is allowed to a depth of 16. Relative paths in a nested include file are relative to the directory location of the include file, not the current working directory of the compiler.

Example

This example shows two file, myinc.inc and myfile.bas. The file myinc.inc has a sub-procedure for doing and incremental move that is used by the main program in myfile.bas.

MyInc.Inc:

```
Sub DoIndexMove( Distance as integer)
       MOVE.RELATIVEDIST = Distance
       MOVE.GOREL
       while     MOVE.MOVING : wend
End Sub
```

MyFile.Bas"

```
$Include     "myinc.inc"
Main
       while 1
                call DoIndexMove(4096)
                Pause(0.5)
       wend
End Main
```

Related Topics

Statement Table(1)

6.1.2 Alias

General Information	
Type	Statement
Description	Allows you to define your own names for system resources, such as Input or Output pins.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	tbd

Description

Allows you to define your own names for system resources, such as Input or Output pins. ALIAS is much more powerful than CONST. Constant expressions are computable at compile-time, whereas an alias has a value that may only be known at the time that it is being used. For this reason aliases should be used with care – over-use of aliases can make it very difficult to read a program.

Example

```
Alias CONVEYOR_IS_RUNNING = (DIN1.STATE = 0)
if CONVEYOR_IS_RUNNING then
       print       "The conveyor is running"
end if
```

Related Topics

Const

6.1.3 Call

General Information	
Type	Statement
Description	Transfer program control to a subroutine. When the subroutine is finished then control is transferred to the line following the CALL.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	tbd

Description

Transfer program control to a subroutine. When the subroutine is finished then control is transferred to the line following the CALL. A subroutine is essentially a function with no return value. Arguments to subroutines are passed "by value". This means that the subroutine receives a copy of these arguments. Any assignments to these arguments made by the subroutine will have no effect on these variables in the calling function or subroutine.

Example

```
Call PrintSum(3,4)
'-------------- Subroutines and Functions -----
-------
Sub PrintSum(i,j,as integer)
       print i+j
End Sub
```

Related Topics

Sub...End Sub

6.1.4 Cls

General Information	
Type	Statement
Description	This statement transmits 40 line feed characters (ASCII code = 10) to the serial port. Cls clears the display of a terminal.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	tbd

Description

This statement transmits 40 line feed characters (ASCII code = 10) to the serial port. Cls clears the display of a terminal.

Example

```
Print     "Take a good look now..."
pause (2)
cls
```

Related Topics

Statement Table(1)

6.1.5 Const

General Information	
Type	Statement
Description	Declares symbolic constants to be used instead of numeric values.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Declares symbolic constants to be used instead of numeric values. Using the CONST Statement can make your program much more readable and self-documenting.

Unlike variables, CONSTANTS can assume only one value in a program.

Example

```
Const SLEW_SPEED = 2500
Const WORK_SPEED = 100

MOVE.RUNSPEED = SLEW_SPEED : MOVE.GOVEL
Pause (0.5)
MOVE.RUNSPEED = WORK_SPEED : MOVE.GOVEL
```

Related Topics

Alias

6.1.6 Dim

General Information	
Type	Statement
Description	Used for declaring variables before use. All variables (except predefined variables) must be declared before they can be used.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Description

Used for declaring variables before use. All variables (except predefined variables) must be declared before they can be used. The DIM statement may also be used to specify that a global variable is non-volatile. When the controller is power-cycled non-volatile variables retain the value present when the controller was powered down. All other user variables are initialized to zero.

The default length for strings is 32 characters. This default can be overridden by following the STRING type designator with a * (see example).

See the examples for how to use DIM to dimension an array.

Example

```
Dim x,y,z as Integer NV '3 non-volatile
integers
Dim q as float  '1 floating point
Dim Array1(4,5) as Integer  'a 4x5 array
Dim A$ as String*50 'a 50 character string
```

Related Topics

Static

6.1.7 Exit

General Information	
Type	Statement
Description	The Exit statement is used to exit from a subroutine, a function, an interrupt, a For...Next or a While...Wend.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The Exit statement is used to exit from a subroutine, a function, an interrupt, a For...Next or a While...Wend. Do not confuse the Exit statement with the End statement. The Exit statement causes program control to pass to the end of the block structure whereas the End statement defines the end of the structure.

Related Topics

Sub...End Sub | Function | Interrupt...End Interrupt | For...Next | While...Wend

Statement Table(1)

6.1.8 For...Next

General Information	
Type	Statement
Description	Allows a series of lines to be executed in a loop a specified number of times.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Allows a series of lines to be executed in a loop a specified number of times. You can exit from a For...Next loop using the Exit statement. If Step increment is omitted then increment defaults to 1. The loop_counter can be floating point or integer. The Step increment can be positive or negative, integer or floating point.

Example

```
'print 2 to 100 in 2's
Dim x as integer
For x = 1 to 100 step 2
       print x
next

'print 0.5 to 1.2 in 0.1 increments
Dim x as float
For x = 0.5 to 1.2 step 0.1
       print x
next
```

Related Topics

While...Wend | Exit

6.1.9 Function

General Information	
Type	Statement
Description	The Function statement is used to declare and define the name, arguments and type of a user defined function.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The Function statement is used to declare and define the name, arguments and type of a user defined function. The code for the function immediately follows the function statement and must be terminated by an End Function statement.

On entry to the function all local variables are initialized to zero including all elements of local arrays. All local string variables are initialized to the null string ("").

If a function takes no arguments then the argument-list (including the parentheses) must be omitted, both when declaring the function and when using the function.

The return value for the function is specified by making an assigment to the function name. See the example (cube) below.

Arguments, including array arguments, are passed by value. Arrays cannot be returned from functions.

Example

This example declares a function that calculates the cube of a floating point number.

```
Main
       dim LocalFloat as float
       LocalFloat = 1.234
       LocalFloat = cube(LocalFloat)
       print LocalFloat
End Main


Function cube( x as float) as float
       cube = x ^3
End Function
```

Related Topics

Dim | Static | Exit | Sub...End Sub

6.1.10 GoTo

General Information	
Type	Statement
Description	GOTO causes the software to jump to the specified label and continue executing from there.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

GoTo causes the software to jump to the specified label and continue executing from there. GoTo is not recommended as a looping technique. Excessive use of the GoTo statement can lead to disorganized and confusing programs. Preferred looping techniques are:

- For...Next
- If...Then...Else
- While...Wend

Related Topics

On Error GoTo

6.1.11 If...Then...Else

General Information	
Type	Statement
Description	If...Then...Else statements controls program execution based on the evaluation of numeric or string expressions.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

If...Then...Else statements controls program execution based on the evaluation of numeric or string expressions. The syntax of If...Then...Else statements are as follows:

```
IF condition1 THEN
    ...statement block1...
[ ElseIf condition2 Then
    ...statement block2...]
[Else
    ...statement block3...]
End If
```

If condition1 is True then statement block1 is executed. Otherwise, if condition2 is True then statement block2 is executed. If the original IF condition is False and all ELSEIF conditions are False then the ELSE statement block (statement block3) is executed.

Related Topics

Select Case | While...Wend | Exit

6.1.12 Input

General Information	
Type	Statement
Description	The Input statement reads a character string received from the console tab screen below the program editor window, terminated by a carriage-return.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The Input statement reads a character string received from the console tab screen below the program editor window, terminated by a carriage-return. The input variable can be integer, floating-point or a string. As an option, the prompt-string is transmitted when the Input statement is encountered. This prompt-string can be either a string constant or a string variable. If the prompt-string is followed by a semi-colon, then a question mark will be printed at the end of the prompt-string. If the prompt-string is followed by a comma then no question mark will be printed.

Example

```
Main
       dim YourName$ as string
       input     "What's your name"; YourName$
       print     "Hello ";YourName$;", I'm leav-
ing..."
End Main
```

Related Topics

Statement Table(1)

6.1.13 On Error GoTo

General Information	
Type	Statement
Description	On Error Goto allows you to define a run-time error handler to prevent run-time errors from halting program execution.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

On Error Goto allows you to define a run-time error handler to prevent run-time errors from halting program execution. Different error handlers can be defined for different parts of the program. An error handler is active from when the On Error Goto statement is executed until another one is executed.

An error handler has the same structure as a subroutine, but must end with a Restart statement. If the error handler does not end with a Restart statement then program execution will terminate at the End Sub statement.

Using the form On Error Goto 0 disables any user defined run-time error handler and reinstalls the default handler. Any subsequent run-time error will print an error message and halt the program.

Errors occurring within the error handler are handled by the default error handler. This means that they will halt program execution.

Example

```
dim Count as integer
Main
        dim y as integer
        if Count < 10 then
                on error goto MyHandler
        else
                on error goto 0
        end if
        y = 0
        pause(0.5)
        y = 1/y
        print         "I'll never get here"
end main

Sub MyHandler
        Count = Count+1
        print Count
        restart
End Sub
```

Related Topics

Restart

6.1.14 Pause()

General Information	
Type	Statement
Description	Causes the program execution to pause for a specified amount of time. The motion of the motor is not affected.
Units	seconds
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Causes the program execution to pause for a specified amount of time. The motion of the motor is not affected. Interrupts are active during a Pause() statement.

Example

```
dim x as float

for x = 0.1 to 2.0 step 0.1
        DOUT1.STATEU = 1
        Pause(x)
        DOUT1.STATEU = 0
        Pause(x)
next
```

Related Topics

Statement Table(1)

6.1.15 Print

General Information	
Type	Statement
Description	Displays formatted output through the console while the program is running.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

AKD BASIC defines zones of 13 characters which can be used to produce output in columns.

If a list of expressions is separated by commas (,) then each subsequent expression is printed in the next zone.

If a list of expressions is separated by semi-colons (;) then the zones are ignored and consecutive expressions are printed in the next available character space.

If a PRINT statement ends in a comma or semi-colon then carriage-return/line-feed at the end of serial output is suppressed.

Example

```
Print     "Hello" , "Goodbye"
Print     "Hello" ; "Goodbye"
Print     "Hello" , "Goodbye";
Print     "...The End."
```

Related Topics

Statement Table(1)

6.1.16 Restart

General Information	
Type	Statement
Description	Causes program execution to begin again from the beginning of the program.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Causes program execution to begin again from the beginning of the program. Restart is the only way to exit from an Error Handler routine. Any interrupts, WHEN statements or loops in progress will be aborted. If the RESTART statement is used to exit from a user error handler then an infinite loop will occur if the error condition is not cleared.

Note: RESTART does not clear the user program variables or by itself change any program variables, any predefined variables or have any effect on motor motion.

Related Topics

MOVE.ABORT | On Error GoTo

6.1.17 Select Case

General Information	
Type	Statement
Description	Select Case executes one of several statement blocks depending upon the value of an expression.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Select Case executes one of several statement blocks depending upon the value of an expression. The test-expression must evaluate to a numeric or floating-point value. There may be as many Cases in the Select Case statement as you want. There can only be one Case Else and it must be the last case in the sequence. The Case Else statement block is executed if all other tests fail.

Select Case statements where the expression-lists are integer constants are executed more quickly at run-time.

Example

This example prints out information about the numbers between 1 and 20.

```
Main
        dim x as integer
        for x = 1 to 20
                print x;" is ";
                select case x
                        case 1, 3, 5, 7, 9
                                print "Odd"
                        case 4, 8
                                print       "4 or 8"
                        case 12 to 18
                                print "between 12 and 18"
                        case else
                                print "other"
                end select
        next
End Main
```

Related Topics

If...Then...Else

6.1.18 Static

General Information	
Type	Statement
Description	The Static statement is used in a Function, Sub or Interrupt to specify that the specified variable's value be remembered even when the Function or Sub is finished.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The Static statement is used in a Function, Sub or Interrupt to specify that the specified variable's value be remembered even when the Function or Sub is finished. The next time that the Function, Sub or Interrupt is executed, the value will be available.

Example

This example illustrates the difference between using Dim and Static in a Sub procedure. 'x' always gets reset to zero, while 'y' continually gets incremented.

```
Main
        while 1
                call MySub
                pause(1)
        wend
End Main

Sub MySub
        dim x as integer          'value is forgotten
        static y as integer          'value is remem-
bered
        x = x + 1
        y = y + 1
        print x,y
End Sub
```

Related Topics

Dim | Sub...End Sub | Function | Interrupt...End Interrupt

6.1.19 Stop

General Information	
Type	Statement
Description	Stops execution of the user program.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Stops execution of the user program. When the user program stops AKD BASIC goes back to message mode, waiting for a command over the communications link.

Related Topics

MOVE.ABORT

6.1.20 Sub...End Sub

General Information	
Type	Statement
Description	The Sub statement declares a sub procedure and defines the sub procedures format.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The Sub statement declares a sub procedure and defines the sub procedures format. A sub procedure is invoked with the Call statement. A sub-procedure can accept arguments like a function, but does not return any value. If the sub-procedure does not take any arguments then it is illegal to provide an empty argument-list "( )" either when defining the sub-procedure or when calling it.

Example

This example defines a sub-procedure that takes one integer argument.

```
Main
        dim x as integer
        for x = 1 to 10
                call MySub(x)
                pause(1)
        next
End Main

Sub MySub(a as integer)
        print a;"---> ";
        if a <= 5 then
                print a * 0.5
        else
                print a * 2.0
        end if
End Sub
```

Related Topics

Call | Function | Exit

6.1.21 Swap

General Information	
Type	Statement
Description	Exchanges the value of two variables.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Exchanges the value of two variables. The two variables must be both numeric (floating point of integer) or both strings.

Example

```
Main
        dim A$, B$ as string
        A$ = "Hello"
        B$ = "Good-bye"
        print A$, B$
        Swap A$, B$
        print A$, B$
End Main
```

Related Topics

Statement Table(1)

6.1.22 While...Wend

General Information	
Type	Statement
Description	Executes a series of lines for as long as the condition after the WHILE is True.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

Executes a series of lines for as long as the condition after the WHILE is True. While...wend statements may be nested. Each Wend is matched to the most recent While. Unmatched While or Wend statements cause compile time errors.

Example

```
DRV.TIME = 0
While          DRV.TIME < 5
       MOVE.DIR = DIN1.STATE : MOVE.GOVEL
Wend
MOVE.ABORT
```

Related Topics

Exit | For...Next

6.2 Operators

This section describes the details of some operators available in AKD BASIC.

6.2.1 MOD

General Information	
Type	Operator
Description	This is the modulus or "remainder" operator. It divides one number by another and returns the remainder.

Syntax

x = y MOD z

Instructions

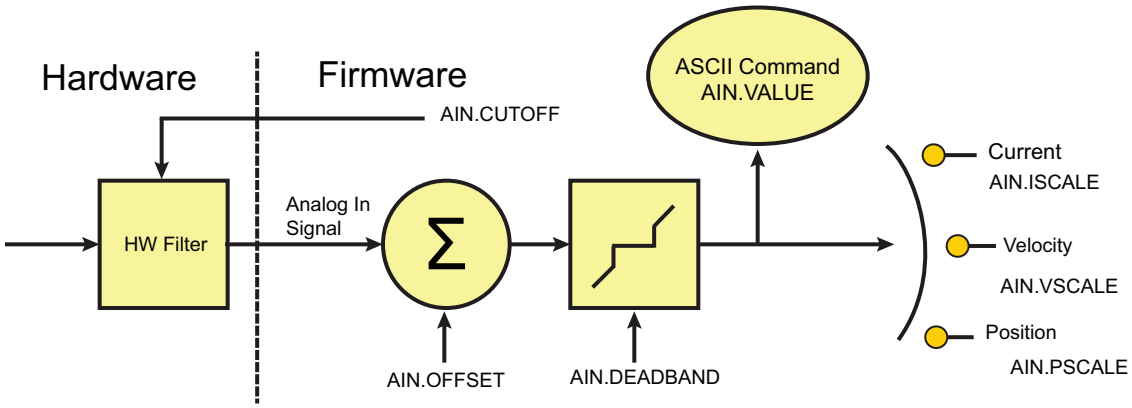
This MOD operator is only used in numeric expressions.

Example

```
Print 19 MOD 5  'prints: 4
```

6.3 AIN Parameters

This section describes the analog input (AIN) parameters. AIN parameters function as shown in the block diagram below:



6.3.1 AIN.CUTOFF

General Information	
Type	NV Parameter
Description	Sets the analog input low-pass filter cutoff frequency.
Units	Hz
Range	0 to 10,000 Hz
Default Value	5,000 Hz
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	0	No	32 bit	No	M_01-03-00-000

Description

AIN.CUTOFF sets the break frequency in Hz for two cascaded single-pole low-pass filters on the hardware command input. Since the two poles are cascaded at the same frequency, the -3 dB frequency is 0.64*AIN.CUTOFF in hertz and the 10% to 90% step response rise time is 0.53/AIN.CUTOFF in seconds.

Suggested operating values are as follows:

- Analog torque opmode: 5 kHz
- Analog velocity opmode: 2.5 kHz
- General purpose analog input high resolution: 500 Hz

6.3.2 AIN.DEADBAND

General Information	
Type	NV Parameter
Description	Sets the analog input signal dead-band.
Units	V
Range	0 to 12.5 V
Default Value	0 V
Data Type	Float
Start Version	M_01-00-00-000

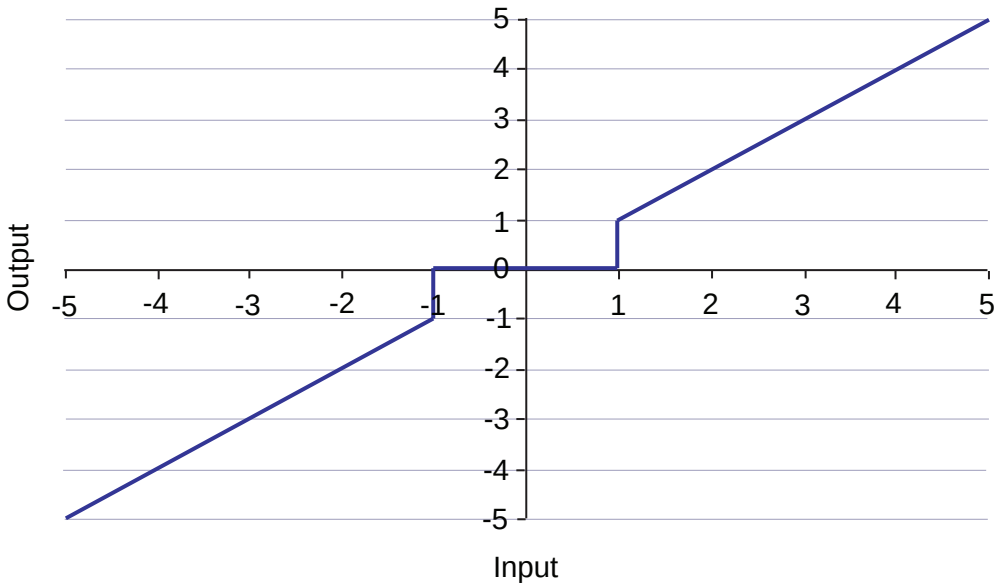
Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	2	No	16 bit	No	M_01-03-00-000

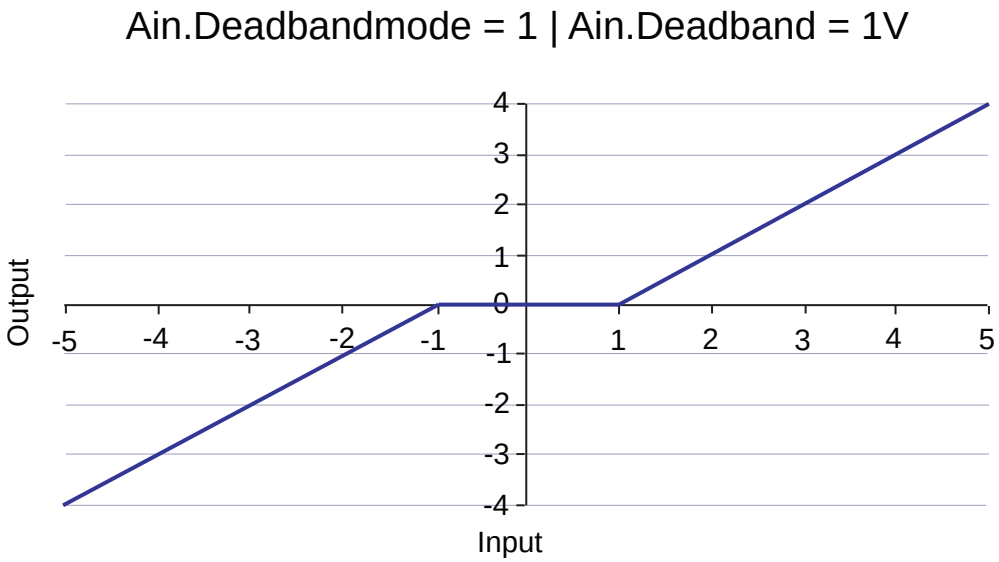
Description

AIN.DEADBAND sets the deadband of the analog input signal. When AIN.DEADBANDMODE is set to 0, and the value of the analog input is less than the value of AIN.DEADBAND, the analog command will be 0. When the analog input is greater or equal to the AIN.DEADBAND, then the analog command will be generated using the scaling specified.

When AIN.DEADBANDMODE is set to 1, the analog command is 0 if the input is less than the deadband value. When the input is greater than the deadband, the output is equal to (Input - Deadband) * Scaling. Below are illustrations of this behavior.

Ain.Deadbandmode = 0 | Ain.Deadband = 1V




Ain.Deadbandmode = 1 | Ain.Deadband = 1V
Output
Input
4
3
2
1
0
-1
-2
-3
-4
-5
-4
-3
-2
-1
1
2
3
4
5

6.3.3 AIN.DEADBANDMODE

General Information	
Type	NV Parameter
Description	Sets the analog input deadband mode.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	AIN.DEADBAND
Start Version	M_01-03-06-000

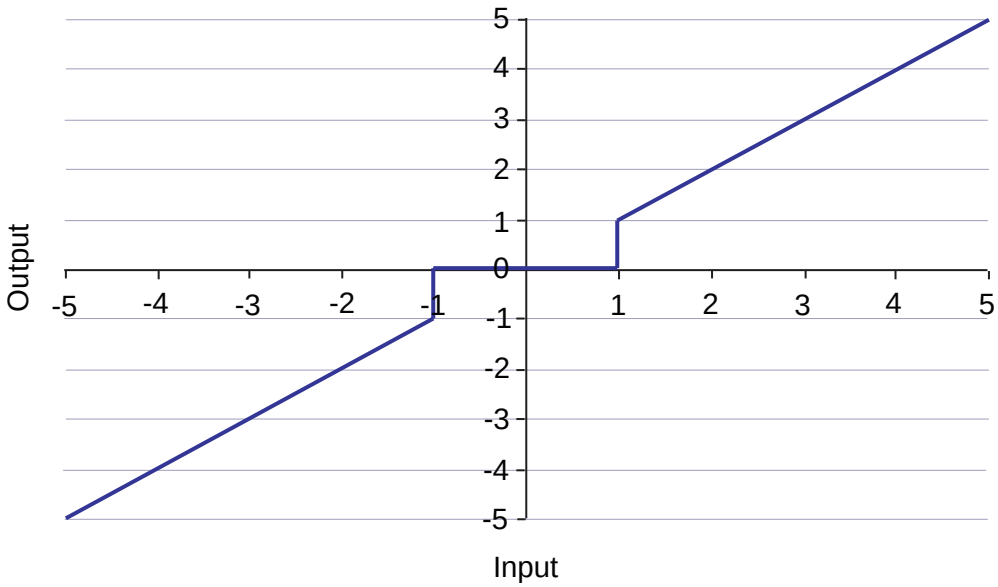
Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1186	No	16 bit	No

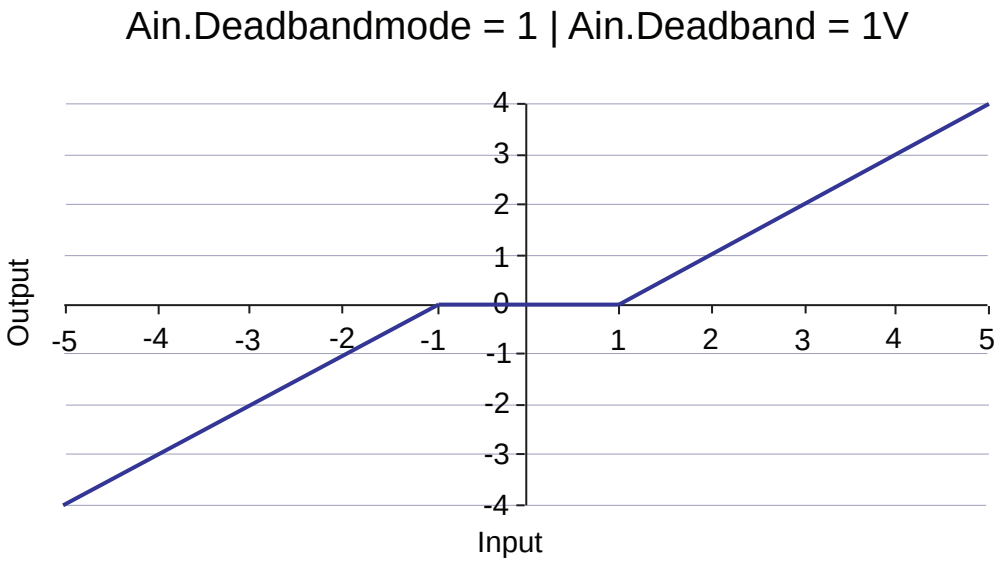
Description

When AIN.DEADBANDMODE is set to 0, and the value of the analog input is less than the value of AIN.DEADBAND, the analog command will be 0. When the analog input is greater or equal to the AIN.DEADBAND, then the analog command will be generated using the scaling specified.

When AIN.DEADBANDMODE is set to 1, the analog command is 0 if the input is less than the deadband value. When the input is greater than the deadband, the output is equal to (Input - Deadband) * Scaling. Below are illustrations of this behavior.

Ain.Deadbandmode = 0 | Ain.Deadband = 1V




Ain.Deadbandmode = 1 | Ain.Deadband = 1V
Output
Input
4
3
2
1
0
-1
-2
-3
-4
-5
-4
-3
-2
-1
1
2
3
4
5

6.3.4 AIN.ISCALE

General Information	
Type	NV Parameter
Description	Sets the analog current scale factor.
Units	A/V
Range	0.001 to 22.4 A/V
Default Value	0.001 A/V
Data Type	Float
Start Version	M_01-01-01-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	4	No	32 bit	No	M_01-03-00-000

Description

AIN.ISCALE sets the analog current scale factor that scales the analog input (AIN.VALUE) for DRV.OPMODE = 1 (analog torque mode).

The value entered is the motor current per 10 V of analog input. This value may be either higher or lower than 100%, but the actual analog input will be limited by the application current limit (IL.LIMITN and IL.LIMITP).

Related Topics

6.3.5 AIN.MODE

General Information	
Type	NV Parameter
Description	Analog input mode
Units	N/A
Range	0 to 2
Default Value	1
Data Type	Integer
See Also	AIN Parameters
Start Version	M_01-04-09-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1188	No	8 bit	No

AKD SynqNet	
Range	0

AKD BASIC	
Range	0 to 1

Description

The parameter AIN.MODE is used to assign a functionality to the voltage measured on the analog input pin.

0 – The analog input value is not used by any function.

1 – This mode only works when DRV.CMDSOURCE is set to 3 (analog). The measured voltage will be scaled with:

- AIN.ISCALE if DRV.OPMODE has been set to 0 (torque mode)
- AIN.VSCALE if DRV.OPMODE has been set to 1 (velocity mode)
- AIN.PSCALE if DRV.OPMODE has been set to 2 (position mode).

Afterwards, the value will be forwarded as a command value to the control-loops.

2 – This mode is used for generating a target velocity of a motion task. This mode works when DRV.OPMODE is set to 2 (position) and DRV.CMDSOURCE is set to 0 (service). The measured voltage will be scaled with AIN.VSCALE.

6.3.6 AIN.OFFSET

General Information	
Type	NV Parameter
Description	Sets the analog input offset.
Units	V
Range	–10 to +10 V
Default Value	0 V
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	6	No	16 bit	Yes	M_01-03-00-000

Description

AIN.OFFSET sets the analog offset, which is added to the analog input command to the drive. This value compensates for the analog input signal (AIN.VALUE) offset or drift.

6.3.7 AIN.PSCALE

General Information	
Type	NV Parameter
Description	Sets the analog position scale factor.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts/V,rad/V, deg/V, (custom units)/V, 16-bit counts/V Linear: counts/V, mm/V, µm/V, (custom units)/V, 16-bit counts/V
Range	Rotary: 1 to 9,223,372,036,854,775 counts/V 0 to 13,493,026.816 rad/V 0.06 to 179.0 deg/V 0 to 10,737,418.240 (PIN/POUT)/V 0 to 140,737,488,355.327 16-bit counts/V Linear: 1 to 9,223,372,036,854,775 counts/V 0 to 2,147,483.648 mm/V 0 to 2,147,483,648.000 µm/V 0 to 10,737,418.240 (PIN/POUT)/V 0 to 140,737,488,355.327 16-bit counts/V
Default Value	Rotary: 1 counts/V 0 rad/V 0 deg/V 0 (PIN/POUT)/V 0 16-bit counts/V Linear: 1 count/V 0 rad/V 0 deg/V 0 (PIN/POUT)/V 0 16-bit counts/V
Data Type	Float
Start Version	M_01-01-01-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3472h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	8	Yes	64 bit	Yes	M_01-03-00-000

Description

AIN.PSCALE is an analog position scale factor that scales the analog input (AIN.VALUE) for DRV.OPMODE = 2 , DRV.CMDSOURCE = 3 (analog position mode).

6.3.8 AIN.VALUE

General Information	
Type	R/O Parameter
Description	Reads the value of the analog input signal.
Units	V
Range	-12.5 to +12.5 V
Default Value	N/A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3470h/4 3509h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	12	No	16 bit	No	M_01-03-00-000

Description

AIN.VALUE reads the analog input value after the value is filtered (as shown in the Analog Input Block Diagram).

6.3.9 AIN.VSCALE

General Information	
Type	NV Parameter
Description	Sets analog velocity scale factor.
Units	Depends on UNIT.VROTARY or UNIT.ACCLINEAR Rotary: rpm/V, rps/V, (deg/s)/V, [(custom units)/s]/V, (rad/s)/V Linear: counts/s/V, (mm/s)/V, (µm/s)/V, [(custom units)/s]/V
Range	Rotary: 0.060 to 60,000 rpm/V 0.001 to 1,000 rps/V 0.359 to 360,000 (deg/s)/V 0.005 to 5,000 [(custom units)/s]/V 0.006 to 6,283.186 (rad/s)/V Linear: 0.001 to 1.000 counts/s/V 0.001*MOTOR.PITCH to 1,000.000*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH to 1,000,000.000*MOTOR.PITCH (µm/s)/V 0.005 to 5,000 [(custom units)/s]/V
Default Value	Rotary: 0.060 rpm/V 0.001 rps/V 0.359 (deg/s)/V 0.005 [(custom units)/s]/V 0.006 (rad/s)/V Linear: 0.001 counts/s/V 0.001*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH (µm/s)/V 0.005 to 5,000 [(custom units)/s]/V
Data Type	Float
Start Version	M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3629h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	14	No	32 bit	No	M_01-03-00-000

Description

AIN.VSCALE is an analog velocity scale factor that scales the analog input AIN.VALUE) for DRV.OPMODE = 1 (analog velocity mode).

The value entered is the motor velocity per 1 V of analog input. This value may be either higher or lower than the application velocity limit (VL.LIMITP orVL.LIMITN), but the actual analog I/O will be limited by VL.LIMITP or VL.LIMITN .

6.3.10 AIN.ZERO

General Information	
Type	Command
Description	Zeroes the analog input signal.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	AIN.VALUE , AIN.OFFSET
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	N/A	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	16	No	Command	No	M_01-03-00-000

Description

AIN.ZERO causes the drive to zero the analog input signal (AIN.VALUE). You may need to execute this command more than once to achieve zero offset, and AIN.OFFSET is modified in this process.

6.4 AOUT Parameters

This section describes the AOUT parameters.

6.4.1 AOUT.CUTOFF

General Information	
Type	NV Parameter
Description	Sets the analog output low-pass filter cutoff frequency.
Units	Hz
Range	0 to 10,000 Hz
Default Value	0 Hz
Data Type	Float
Start Version	M_01-04-01

Description

AOUT.CUTOFF sets the cutoff frequency in Hz for a single-pole low-pass filter on the Analog Output.

A value of 0 Hz will turn off the filter and will allow all frequencies to pass through.

The filter can be used with all modes of Analog Output.

6.4.2 AOUT.DEBUGADDR

General Information	
Type	NV Parameter
Description	Sets the memory address to debug.
Units	N/A
Range	4 to 4,292,870,142
Default Value	4
Data Type	Integer
Start Version	M_01-01-01-000

Description

AOUT.DEBUGADDR sets the memory address to debug when AOUT.MODE = 9 (debug mode).

6.4.3 AOUT.DEBUGDATATYPE

General Information	
Type	NV Parameter
Description	Sets the data type of the value to be debugged.
Units	N/A
Range	0 to 10
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-01-01-000

Description

AOUT.DEBUGDATATYPE is used in AOUT.MODE = 9 (debug mode).

This parameter sets the data type of the value to be debugged according to the table below:

Value	Data Type
0	Illegal type
1	Signed (1 byte)
2	Unsigned (1 byte)
3	Signed (2 bytes)
4	Unsigned (2 bytes)
5	Signed (4 bytes)
6	Unsigned (4 bytes)
7	Signed (8 bytes)
8	Unsigned (8 bytes)
9	Pointer to one byte
10	Fix shift

6.4.4 AOUT.DEBUGSCALE

General Information	
Type	NV Parameter
Description	Sets the scale to be used for debug.
Units	N/A
Range	0.001 to 9,223,372,036,854,775.000
Default Value	1
Data Type	Float
See Also	AOUT.MODE
Start Version	M_01-01-01-000

Description

AOUT.DEBUGSCALE sets the scale to be used for debug when AOUT.MODE = 9 (debug mode).

6.4.5 AOUT.ISCALE

General Information	
Type	NV Parameter
Description	Sets the analog current scale factor.
Units	A/V
Range	0.001 to 22.4 A/V
Default Value	0.001 to 22.4 A/V
Data Type	Float
See Also	AOUT.VALUE
Start Version	M_01-01-01-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	18	No	32 bit	No	M_01-03-00-000

Description

AOUT.ISCALE sets the analog current scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 4 or 5. The value entered is the motor current per 10 V of analog input or output. This value may be either higher or lower than 100%, but the actual analog I/O will be limited by the application current limit (IL.LIMITN and IL.LIMITP).

6.4.6 AOUT.MODE

General Information	
Type	NV Parameter
Description	Sets the analog output mode.
Units	N/A
Range	0 to 11
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3470h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	20	No	16 bit	No	M_01-03-00-000

SynqNet Information	
Range	12

Description

AOUT.MODE sets the analog output functionality.

AOUT.MODE	Description
0	User variable. The analog output signal is determined by the user (using AOUT.VALUEU).
1	Actual velocity. The analog signal describes the current velocity value (VL.FB).
2	Velocity error. The analog signal describes the velocity error value.
3	Velocity command. The analog signal describes the velocity command value.
4	Actual current. The analog signal describes the actual current value.
5	Current command. The analog signal describes the current command value.
6	Actual position. The analog signal describes the current position value.
7	Position error. The analog signal describes the position error value.
8	Triangle wave. The analog signal is a triangle wave (sawtooth pattern).
9	Debug mode. In this mode the user can define a drive variable to monitor via the analog output (AOUT.VALUEU).
10	Unfiltered Velocity (VL.FBUNFILTERED)
11	Filtered Velocity - 10Hz Lowpass (VL.FBFILTER)

Example

You can use AOUT.MODE and AOUT.VALUEU to configure an output signal as follows:

```
-->AOUT.MODE 0
-->AOUT.VALUEU 5
-->AOUT.VALUEU 4.33
```

6.4.7 AOUT.OFFSET

General Information	
Type	NV Parameter
Description	Sets the analog input offset.
Units	V
Range	-10 to +10 V
Default Value	0 V
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	22	No	16 bit	Yes	M_01-03-00-000

Description

This parameter sets the analog input offset.

6.4.8 AOUT.PSCALE

General Information	
Type	NV Parameter
Description	Sets the analog position scale factor.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts/V,rad/V, deg/V, (custom units)/V, 16-bit counts/V Linear: counts/V, mm/V, µm/V, (custom units)/V, 16-bit counts/V
Range	Rotary: 1 to 9,223,372,036,854,775 counts/V 0 to 13,493,026.816 rad/V 0 to 773,094,113.280 deg/V 0 to 10,737,418.240 (custom units)/V 0 to 140,737,488,355.327 16-bit counts/V Linear: 1 to 9,223,372,036,854,775 counts/V 0 to 2,147,483.648 mm/V 0 to 2,147,483,648.000 µm/V 0 to 10,737,418.240 (custom units)/V 0 to 140,737,488,355.327 16-bit counts/V
Default Value	Rotary: 1 counts/V 0 rad/V 0 deg/V 0 (custom units)/V 0 16-bit counts/V Linear: 1 counts/V 0 rad/V 0 deg/V 0 (custom units)/V 0 counts16 bit/V
Data Type	Float
See Also	AOUT.VALUE
Start Version	M_01-01-01-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3471h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	24	Yes	64 bit	No	M_01-03-00-000

Description

AOUT.PSCALE is an analog position scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 6, or 7 (actual position or position error) per 10 V of analog input or output.

6.4.9 AOUT.VALUE

General Information	
Type	R/O Parameter
Description	Reads the analog output value.
Units	V
Range	–10 to +10 V
Default Value	0
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3470h/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	28	Yes	64 bit	Yes	M_01-03-00-000

Description

AOUT.VALUE reads the analog output value.

6.4.10 AOUT.VALUEU

General Information	
Type	R/W Parameter
Description	Sets the analog output value.
Units	V
Range	–10 to +10 V
Default Value	0
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3470h/3	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	32	Yes	64 bit	Yes	M_01-03-00-000

Description

AOUT.VALUEU reads/writes the analog output value when AOUT.MODE = 0 (analog output signal is determined by the user).

6.4.11 AOUT.VSCALE

General Information	
Type	NV Parameter
Description	Sets the velocity scale factor for analog output.
Units	Depends on UNIT.VROTARY or UNIT.ACCLINEAR Rotary: rpm/V, rps/V, (deg/s)/V, [(custom units)/s]/V, (rad/s)/V Linear: counts/s/V, (mm/s)/V, (µm/s)/V, [(custom units)/s]/V
Range	Rotary: 0.060 to 60,000 rpm/V 0.001 to 1,000 rps/V 0.359 to 360,000 (deg/s)/V 0.005 to 5,000 [(custom units)/s]/V 0.006 to 6,283.186 (rad/s)/V Linear: 0.001 to 1.000 counts/s/V 0.001*MOTOR.PITCH to 1,000.000*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH to 1,000,000.000*MOTOR.PITCH(µm/s)/V 0.005 to 5,000 [(custom units)/s]/V
Default Value	Rotary: 0.060 rpm/V 0.001 rps/V 0.359 (deg/s)/V 0.005 [(custom units)/s]/V 0.006 (rad/s)/V Linear: 0.001 counts/s/V 0.001*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH (µm/s)/V 0.005 [(custom units)/s]/V
Data Type	Float
See Also	AOUT.VALUE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3470h/5	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	36	No	32 bit	No	M_01-03-00-000

Description

AOUT.VSCALE is an analog velocity scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 1, 2, or 3. The value entered is the motor velocity per 10 V of analog output. This value may be either higher or lower than the application velocity limit (VL.LIMITP or VL.LIMITN), but the actual analog I/O will be limited by VL.LIMITP or VL.LIMITN.

6.5 CAM Parameters

This section describes the CAM parameters.

6.5.1 CAM.ACTIVATE

General Information	
Type	Parameter
Description	Activates the specified cam table.
Units	None
Range	0 to 8
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAM.ACTIVE activates the specified cam table. The Position Command is calculated according to the Master Position and the points in the specified cam table.

When you activate a new cam, the drive accelerates (at EGEAR.ACCLIMIT) or decelerates (at EGEAR.DECLIMIT) as necessary to the speed required by the present motion of the Cam Master and the slave position profile defined in the cam table.

When speed synchronization is achieved, EGEAR.LOCK is set to one and a correction move is performed to bring the slave into position lock with the cam table. The direction of this move is controlled by CAM.CORRECTDIR. The parameters of this correction move are the same as for any other move (i.e., MOVE.ACC, MOVE.DEC, MOVE.RUNSPEED).

If the master is not moving or if the slave position profile in the cam table does not require cam motion when the cam is activated, the speed synchronization occurs instantly and the correction move is executed as soon as the cam is activated.

CAM.ACTIVATE is automatically set to zero (i.e., any cam is disengaged) when the drive is disabled.

To disable the correction move, set CAM.CORRECTDIR = 3.

You must declare and create a cam table before you make it active. If MOVE.RUNSPEED is equal to zero when you set CAM.ACTIVATE, a run-time error is generated because the correction move cannot be performed.

Examples

The following example declares, creates, and activates a cam.

```
Main
       CAM.CREATE(1, 5) 'allocate space for cam #1, 5
points
                'start the cam create block
                CAM.ADDPOINT(0, 0)
                CAM.ADDPOINT(200, 65536 / 10)
                CAM.ADDPOINT(400, 65536 / 8)
                'add the points
                CAM.ADDPOINT(600, 65536 * 3 / 4)
                CAM.ADDPOINT(800, 65536)
       End           'end the cam create block
       DRV.SWENABLE = 0 'Disable motor while setting
position
       CAM.MASTER = 2 'Cam Master = External Encoder
```

```
       EXTENCODER.POSMODULO = 800 'set master counts
per cycle
       PL.MODPEN = 1 'enable slave modulo
       PL.MODP2 = 65536 'set slave (AKD BASIC) counts
per cycle
       EXTENCODER.POSITION = 0 'set Master position
to 0
       MOVE.POSCOMMAND = 0 'set slave (AKD BASIC)
position to 0
       DRV.SWENABLE = 1 'enable the motor
       CAM.ACTIVATE = 1 'activate cam#1
       While 1=1:wend
End Main
```

Related Topics

CAM.CORRECTDIR

6.5.2 CAM.ADDPOINT

General Information	
Type	Statement
Description	CAM.ADDPOINT(Master Position, Slave Position) Adds the specified "point" to the cam table being created.
Units	Master Encoder Counts, Position User Units
Range	-2^63 to 2^63-1
Default Value	N/A
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAM.ADDPOINT adds the specified “point” (master position and corresponding slave position) to the cam table being created. This statement is only used inside a CAM.CREATE block.

You must be inside a CAM.CREATE block to use the CAM.ADDPOINT statement. The master position for the first CAM.ADDPOINT statement in a CAM.CREATE block must always be zero. The master position must always increase as you add points to the cam table. There must be at least two points in your cam table.

Examples

The following example declares, creates, and activates a cam.

```
Main
        CAM.CREATE(1, 5) 'allocate space for cam #1, 5
points
                'start the cam create block
                CAM.ADDPOINT(0, 0)
                CAM.ADDPOINT(200, 65536 / 10)
                CAM.ADDPOINT(400, 65536 / 8)
                'add the points
                CAM.ADDPOINT(600, 65536 * 3 / 4)
                CAM.ADDPOINT(800, 65536)
        End         'end the cam create block
        DRV.SWENABLE = 0 'Disable motor while setting
position
        CAM.MASTER = 2 'Cam Master = External Encoder

        EXTENCODER.POSMODULO = 800 'set master counts
per cycle
        PL.MODPEN = 1 'enable slave modulo
        PL.MODP2 = 65536 'set slave (AKD BASIC) counts
per cycle
        EXTENCODER.POSITION = 0 'set Master position
to 0
        MOVE.POSCOMMAND = 0 'set slave (AKD BASIC)
position to 0
        DRV.SWENABLE = 1 'enable the motor
```

```
        CAM.ACTIVATE = 1 'activate cam#1
        While 1=1:wend
End Main
```

6.5.3 CAM.CORRECTDIR

General Information	
Type	R/W
Description	Specifies the direction of the correction move when a new cam table is activated (set CAM.ACTIVATE = n) or when speed synchronization is achieved.
Units	None
Range	0 to 3
Default Value	2 (shortest distance)
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAM.CORRECTDIR takes one of the following values:

- 0 move is done clockwise
- 1 move is done counter-clockwise
- 2 move is done in the direction yielding the shortest move (see
- below)
- 3 no correction move is performed.

Use MOVE.ACC, MOVE.DEC and MOVE.RUNSPEED for the correction move. Even if CAM.CORRECTDIR specifies a clockwise correction move, it only specifies the direction of the superimposed move. If the cam generated speed is the opposite direction and larger than MOVE.RUNSPEED, the slave slows down.

For CAM.CORRECTDIR = 2, the direction of the correction is calculated (based upon PosModulo) to yield the shortest distance move. For example, if PosModulo = 10000 and the clockwise correction move is 8000, a counter-clockwise move of 2000 is performed instead.

Example

In the following example, the correction move is in the direction yielding the shortest move distance.

```
....
'The cam table for Cam #1 needs to have been
'already declared and created
'_________________________
CAM.CORRECTDIR = 2
CAM.ACTIVATE = 1
....
```

Related Topics

CAM.ACTIVATE

6.5.4 CAM.CREATE

General Information	
Type	Statement
Description	CAM.CREATE (x, y) Initiates the creation of a cam table.
Units	N/A
Range	x = 1-8, y = 3-1000
Default Value	N/A
Data Type	Integers
Start Version	M_01-06-00-000

Description

CAM.CREATE initiates the creation of a cam table. The actual points in the cam table are inserted with a series of CAM.ADDPOINT statements. The CAM.CREATE block must terminated by an End statement.

You can create a cam table as many times as you want. You must create a cam table before you make it active. You cannot create a cam table if it is active. The master position for the first entry must be 0. The master positions must keep increasing as you add points. EXTENCODER.POSMODULO must equal the total master distance in you CAM. For a repeating CAM, PL.MODP2 should be set equal to the distance that the slave travels in one CAM cycle.

Examples

```
Main
       CAM.CREATE(1, 5) 'allocate space for cam #1, 5
points
               'start the cam create block
               CAM.ADDPOINT(0, 0)
               CAM.ADDPOINT(200, 65536 / 10)
               CAM.ADDPOINT(400, 65536 / 8)
               'add the points
               CAM.ADDPOINT(600, 65536 * 3 / 4)
               CAM.ADDPOINT(800, 65536)
       End       'end the cam create block
       DRV.SWENABLE = 0 'Disable motor while setting
position
       CAM.MASTER = 2 'Cam Master = External Encoder

       EXTENCODER.POSMODULO = 800 'set master counts
per cycle
       PL.MODPEN = 1 'enable slave modulo
       PL.MODP2 = 65536 'set slave (AKD BASIC) counts
per cycle
       EXTENCODER.POSITION = 0 'set Master position
to 0
       MOVE.POSCOMMAND = 0 'set slave (AKD BASIC)
position to 0
       DRV.SWENABLE = 1 'enable the motor
       CAM.ACTIVATE = 1 'activate cam#1
```

```
       While 1=1:wend
End Main
```

Related Topics

CAM.ADDPOINT | CAM.ACTIVATE

6.5.5 CAM.MASTER

General Information	
Type	R/W
Description	Specifies the source of the input to the cam table for cam profiling.
Units	None
Range	0 to 2
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAM.MASTER takes one of the following values:

Value	Description
0	EXTENCODER.POSITION + CAMVM.POSITION
1	CAMVM.POSITION only (EXTENCODER.POSITION is ignored)
2	EXTENCODER.POSITION only (CAMVM.POSITION is ignored)

Related Topics

CAM.MASTERPOS

6.5.6 CAM.MASTERPOS

General Information	
Type	R/O
Description	Gives the value of the master position presently being used as the input to the cam table.
Units	Encoder counts
Range	0 to EXTENCODER.POSMODULO
Default Value	N/A
Data Type	Integer
Start Version	M_01-06-00-000

Description

The value of CAM.MASTERPOS depends upon EXTENCODER.POSITION, CAMVM.POSITION and CAM.MASTER as follows:

Value of CAM.MASTER	Value of CAM.MASTERPOS
0	CAMVM.POSITION + EXTENCODER.POSITION
1	CAMVM.POSITION
2	EXTENCODER.POSITION

Related Topics

CAM.MASTER | DRV.HANDWHEEL | CAMVM.POSITION

6.5.7 CAM.SLAVEOFFSET

General Information	
Type	R/O
Description	CAM.SLAVEOFFSET indicates the offset (or difference) between MOVE.POSCOMMAND and the position command that is calculated from the active cam table based upon the present value of EXTENCODER.POSITION and/or CAMVM.POSITION.
Units	Feedback counts
Range	N/A
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAM.SLAVEOFFSET indicates the offset (or difference) between PL.CMD and the position command that is calculated from the active cam table based upon the present value of DRV.HANDWHEEL and/or CAMVM.POSITION. This offset is the result of incremental (MOVE.GOREL) or velocity (MOVE.GOVEL) moves superimposed (by the user) on the cam table. If there is no active cam (CAM.ACTIVE = 0), the value of this variable is undefined.

6.5.8 CAMVM.DIR

General Information	
Type	R/W
Description	Specifies the direction the virtual encoder goes when CAMVM.GOVEL is executed.
Units	None
Range	0, 1
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAMVM.DIR specifies the direction the virtual encoder goes when CAMVM.GOVEL is executed. It also sets the direction of the virtual encoder when CAMVM.GOUPDATE is executed if the virtual encoder is performing a CAMVM.GOVEL move.

0 is positive

1 is negative

Example

```
'This runs the virtual encoder forward at
20,000 counts/sec
CAMVM.FREQ = 20000
CAMVM.DIR = 0
CAMVM.GOVEL
pause(5)
'This runs the virtual encoder backwards at
40,000 counts/sec
CAMVM.FREQ = 40000
CAMVM.DIR = 1
CAMVM.GOVEL
```

Related Topics

CAMVM.GOVEL

6.5.9 CAMVM.FREQ

General Information	
Type	R/W
Description	CAMVM.FREQ sets the maximum frequency allowed during a relative (CAMVM.GOREL) move, and sets the commanded speed during a velocity move (CAMVM.GOVEL)
Units	Encoder counts/second
Range	0 to 1,000,000
Default Value	10,000
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAMVM.FREQ sets the maximum frequency allowed during a relative (CAMVM.GOREL) move, and sets the commanded speed during a velocity move (CAMVM.GOVEL).

Example

```
'This will run the virtual encoder forward at
20,000 counts/sec
'-----------------------------------------------
----------------
CAMVM.FREQ = 20000
CAMVM.DIR = 0
CAMVM.GOVEL
```

6.5.10 CAMVM.GOREL

General Information	
Type	Statement
Description	Makes the virtual master move the distance specified by CAMVM.RELATIVEDIST.
Units	None
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	M_01-06-00-000

Description

CAMVM.GOREL (Go Relative) causes the virtual master to move a distance specified by CAMVM.RELATIVEDIST. The virtual master runs at the frequency specified by CAMVM.FREQ. Use CAMVM.GOUPDATE to modify this frequency during the move.

Program execution continues with the line immediately following the CAMVM.GOREL statement as soon as the move is initiated. Program execution does not wait until the move is complete. The drive does not need to be enabled in order for to use the virtual master.

Related Topics

CAMVM.GOVEL | CAMVM.STOP | CAMVM.GOUPDATE

6.5.11 CAMVM.GOUPDATE

General Information	
Type	Statement
Description	Updates a move in progress with new move parameters.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	M_01-06-00-000

Description

Updates a move in progress with new move parameters. This allows you to change motion on-the-fly without having to stop motion and initiate a new move. CAMVM.GOUPDATE updates CAMVM.DIR (for a CAMVM.GOVEL) and CAMVM.FREQ (for a CAMVM.GOVEL or CAMVM.GOREL).

Program execution continues with the line immediately following CAMVM.GOUPDATE as soon as the move is initiated. Program execution does not wait until the move is complete. CAMVM.GOUPDATE does not initiate motion if there is no move in progress.

Related Topics

CAMVM.GOREL | CAMVM.GOREL

6.5.12 CAMVM.GOVEL

General Information	
Type	Statement
Description	CAMVM.GOVEL (Go at Velocity) causes the virtual master to move continuously at the frequency specified by CAMVM.FREQ in the direction (positive or negative) specified by CAMVM.DIR. The frequency or direction is modified during the move using CAMVM.GOUPDATE.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	M_01-06-00-000

Description

CAMVM.GOVEL (Go at Velocity) causes the virtual master to move continuously at the frequency specified by CAMVM.FREQ in the direction (positive or negative) specified by CAMVM.DIR. The frequency or direction is modified during the move using CAMVM.GOUPDATE.

When the move is initiated, program execution continues with the line immediately following CAMVM.GOVEL. Program execution does not wait until the move is complete. CAMVM.STOP stops a velocity move on the virtual encoder. Executing CAMVM.GOREL after CAMVM.GOVEL and before CAMVM.STOP causes the virtual encoder to switch to an incremental move that terminates when CAMVM.RELATIVEDIST encoder counts have been put out. The drive does not need to be enabled to use the virtual master.

Example

This runs the virtual encoder forward at 20,000 counts/sec

```
CAMVM.FREQ = 20000
CAMVM.DIR = 0
CAMVM.GOVEL
```

Related Topics

CAMVM.GOREL | CAMVM.STOP | CAMVM.GOUPDATE

6.5.13 CAMVM.MOVING

General Information	
Type	R/O
Description	Indicates if the virtual encoder is moving.
Units	None
Range	0, 1
Default Value	N/A
Data Type	Integer
Start Version	M_01-06-00-000

Description

CAMVM.MOVING indicates if the virtual encoder is moving.

- 0 - virtual encoder is not moving
- 1 - virtual encoder is moving

Example

```
'Start an incremental move on the virtual
encoder
CAMVM.FREQ = 10000
CAMVM.RELATIVEDIST = 123456
CAMVM.GOREL
DRV.TIME = 0
while CAMVM.MOVING : wend
print DRV.TIME
```

Related Topics

CAMVM.GOVEL | CAMVM.GOREL

6.5.14 CAMVM.POSITION

General Information	
Type	R/W
Description	Contains the current value of the virtual encoder counter.
Units	counts
Range	0 to (EXTENCODER.POSMODULO - 1)
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

Control the virtual encoder using CAMVM.GOVEL and CAMVM.GOREL.

EXTENCODER.POSMODULO is used as the modulo value for CAMVM.POSITION.

Example

This example shows how CAMVM.POSITION is updated during a CAMVM.GOREL move.

```
CAMVM.FREQ = 10000
CAMVM.RELATIVEDIST = 100000
DRV.TIME = 0
EXTENCODER.POSMODULO = 200000
CAMVM.POSITION = 0
CAMVM.GOREL
While DRV.RUNTIME < 12
Print "DRV.TIME=" ; DRV.RUNTIME , "CAMVM.P-
OSITION=" ; CAMVM.POSITION , "CAMVM.MOVING=" ;
CAMVM.MOVING
Pause(1)
Wend
```

Related Topics

CAMVM.GOREL | CAMVM.GOVEL | CAMVM.MOVING

6.5.15 CAMVM.RELATIVEDIST

General Information	
Type	R/W
Description	Specifies the number or counts that the virtual encoder (virtual master) will put out during an incremental move (CAMVM.GOREL).
Units	Encoder counts
Range	-2^63 to 2^63-1
Default Value	0
Data Type	Integer
Start Version	M_01-06-00-000

Description

The "move" is performed based upon the value of CAMVM.FREQ. The value of the virtual encoder counter is in the variable CAMVM.POSITION. The modulo value EXTENCODER.POSMODULO is applied to CAMVM.POSITION as well. You can check whether or not the virtual encoder is moving using the variable CAMVM.MOVING. You can move the virtual encoder (using CAMVM.GOREL or CAMVM.GOVEL) whether or not the drive is enabled or disabled.

Example

This example moves the virtual encoder 100,000 counts at a frequency of 20,000 counts/second. This move will take about 5 seconds.

```
'set up CAMVM.POSTION and virtual move param-
eters
'-----------------------------------------------
---------------------------------
CAMVM.POSTION = 0
CAMVM.RUNFREQ = 20000
CAMVM.RELATIVEDIST = 100000
'initiate the move
'------------------------
DRV.TIME = 0 'set time to zero just for meas-
urement
CAMVM.GOREL
'wait for the move to be complete
'-----------------------------------------------
while CAMVM.MOVING = 1 : wend
'print the results
'----------------------
print "CAMVM.POSITION = ";CAMVM.POSITION
print "time = "; DRV.TIME
```

6.5.16 CAMVM.STOP

General Information	
Type	Statement
Description	CAMVM.STOP stops the virtual encoder
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	M_01-06-00-000

Description

CAMVM.STOP stops the virtual encoder. CAMVM.POSITION stays at its present value.

Program execution continues with the line immediately following CAMVM.STOP as soon as the move is initiated. Program execution does not wait until the move is complete.

Example

Run the virtual encoder forward at 20,000 counts/sec for 5 seconds and then stop.

```
CAMVM.FREQ = 20000
CAMVM.DIR = 0
CAMVM.GOREL
pause(5)
CAMVM.STOP
```

Related Topics

CAMVM.GOREL | CAMVM.GOVEL

6.6 CAP Parameters

This section describes the CAP parameters.

6.6.1 CAP0.EDGE, CAP1.EDGE

General Information	
Type	NV Parameter
Description	Selects the capture edge.
Units	N/A
Range	1 to 3
Default Value	1
Data Type	U8
See Also	CAP0.PREEDGE, CAP1.PREEDGE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	54	CAP0.EDGE	No	8 bit	No	M_01-03-00-000
	80	CAP1.EDGE				

Description

The filtered trigger source is monitored for rising edge, falling edge, or both edges. The event mode logic may ignore the precondition edge detection; however, the trigger always uses edge detection.

The precondition logic has an identical feature controlled by CAP0.PREEDGE, CAP1.PREEDGE .

Value	Description
0	Reserved
1	Rising edge
2	Falling edge
3	Both edges

6.6.2 CAP0.EN, CAP1.EN

General Information	
Type	NV Parameter
Description	Enables or disables the related capture engine.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Boolean
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	56	CAP0.EN	No	8 bit	No	M_01-03-00-000
	82	CAP1.EN				

Description

This parameter enables or disables the related capture engine. After each successful capture event, this parameter is reset to 0 and must be activated again for the next capture. Also note that CAP0.PLFB, CAP1.PLFB is set to 0 when this parameter is set to 1.

0 = Disable

1 = Enable

6.6.3 CAP0.EVENT, CAP1.EVENT

General Information	
Type	NV Parameter
Description	Controls the precondition logic.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CAN-open	3460h /5	CAP0.EVENT	M_01-00-00-000
	3460h /6	CAP1.EVENT	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	58	CAP0.EVENT	No	8 bit	No	M_01-03-00-000
	84	CAP1.EVENT				

Description

The event mode controls use of the precondition logic. If this field is not 0, then the precondition input is selected by CAPx.TRIGGER. If this field is 1, then the precondition edge is selected by the CAPx.PREEDGE. The four event modes are listed below.

Event	Description
0	Precondition settings ignored.
1	Trigger on first trigger event after selected edge on precondition input.
2	Trigger on first trigger event to occur while precondition input is 1
3	Trigger on first trigger event to occur while precondition input is 0.

Example

Event 0

The following diagram shows an example of Event = 0 (trigger on edge, trigger edge = rising). In this mode, the precondition logic is ignored.

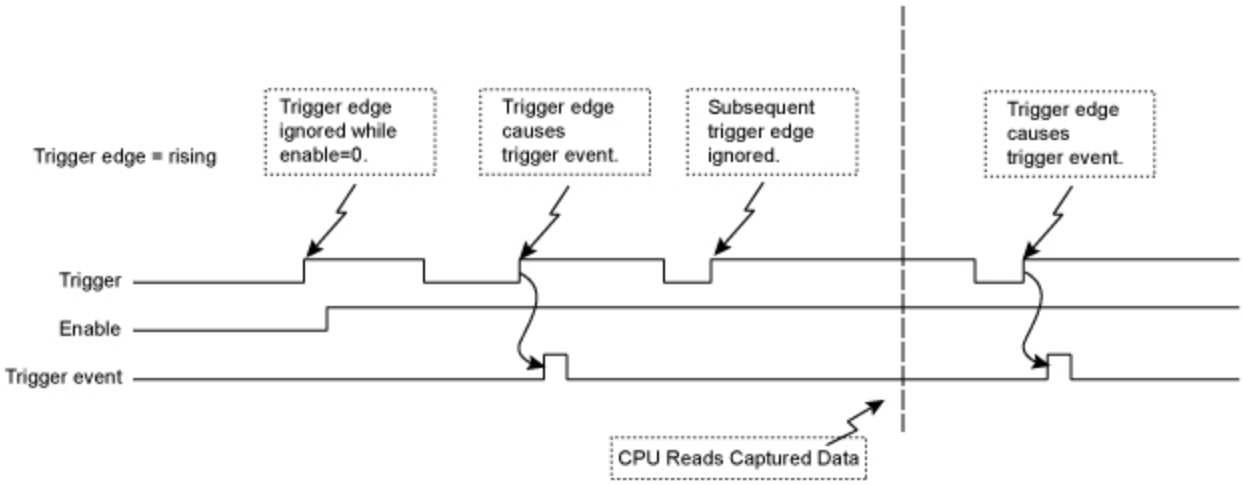


Figure 1: Trigger Edge Mode

Events 2 and 3 (Trigger edge while precondition = 0 or 1)

In these events, the precondition logic samples the current (post-filter) state of the selected pre-condition source input. The capture engine looks for a trigger edge while the precondition input is at a “1” or “0” state.

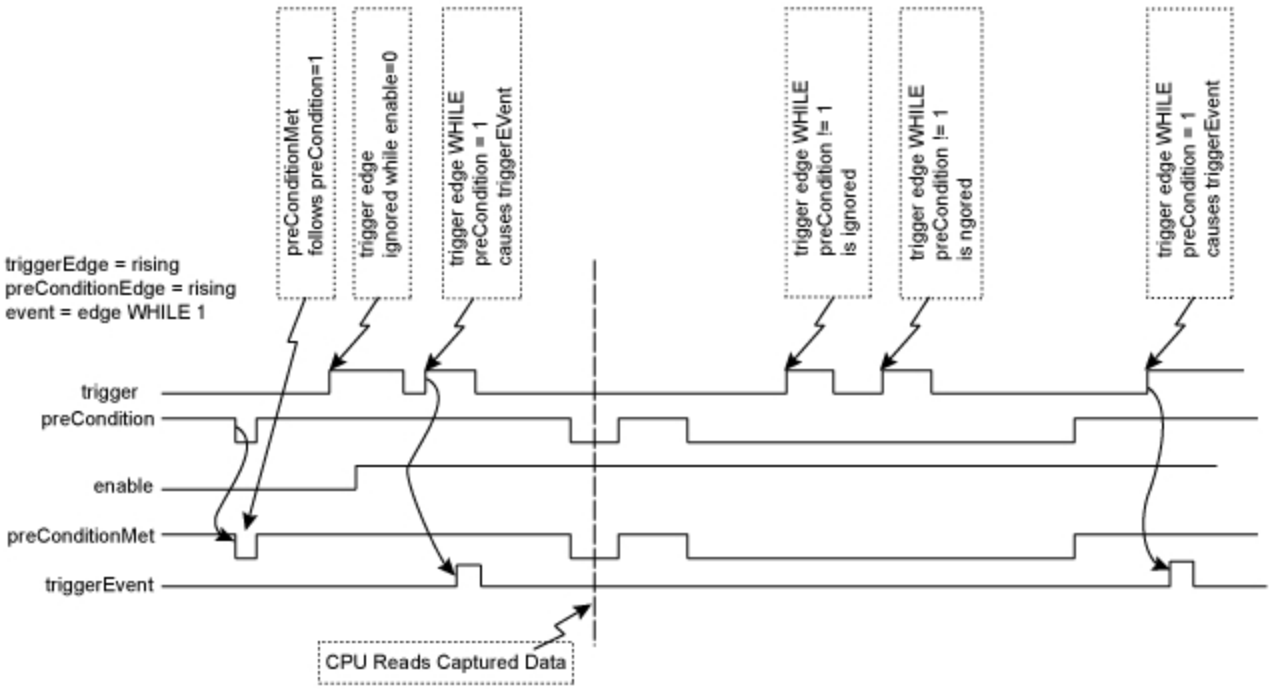


Figure 2: Trigger edge WHILE precondition edge

Event 1 (Trigger edge after precondition)

In this event, each trigger event requires Enable=1, a new precondition edge, followed by a new trigger edge. The sequence requirements are shown in the figure below.

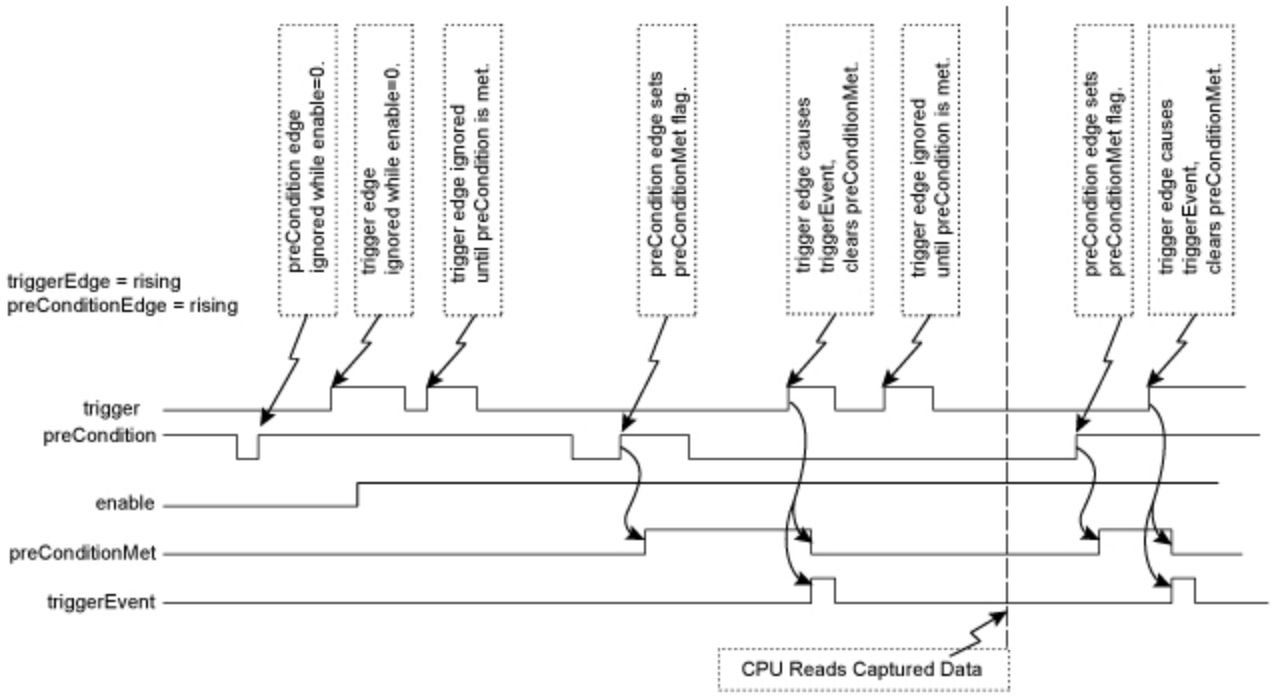


Figure 3: Trigger edge after precondition edge

Note: If the precondition and trigger edges occur at the same time, it is not a valid trigger event. A subsequent trigger edge must occur after the precondition edge. The same time resolves to a single 40 ns clock tick in the trigger event logic (after the optional filter function as well as any sensor, cable, or noise delays).

Related Topics

1 Using Position Capture

6.6.4 CAP0.FILTER, CAP1.FILTER

General Information	
Type	R/W Parameter
Description	Sets the filter for the capture source input.
Units	N/A
Range	0 to 2
Default Value	0
Data Type	U8
See Also	CAP0.PREFILTER, CAP1.PREFILTER
Start Version	M_01-00-00-000
End Version	M_01-03-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	60	CAP0.FILTER	No	8 bit	No	M_01-03-00-000
	86	CAP1.FILTER				

Description

These parameters are not functional in M_01-03-00-000. In future releases, you can use DINx.FILTER to select a filter on the input channel.

Related Topics

DIN1.FILTER TO DIN7.FILTER

6.6.5 CAP0.MODE, CAP1.MODE

General Information	
Type	NV Parameter
Description	Selects the captured value.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CAN-open	3460h /3	CAP0.MODE	M_01-00-00-000
	3460h /4	CAP1.MODE	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	62	CAP0.MODE	No	8 bit	No	M_01-03-00-000
	88	CAP1.MODE				

Description

Mode 0 is the standard position capture, which stores PL.FB . Data can be retrieved with CAP0.PLFB, CAP1.PLFB .

Mode 1 is the drive internal time capture. Data can be retrieved with CAP0.T, CAP1.T .

Mode 2 is the KAS EtherCAT distributed clock time (DCT) capture. Instead of using a position value, the DCT is calculated. There is no user parameter to retrieve the captured DCT. Attempting to set Mode = 2 with anything other than an EtherCAT system will result in an invalid parameter error.

Mode 3 is the capture of the primary encoder signal. This mode is used to home onto a feedback index. This mode sets the other parameters needed for this mode. These parameters can be changed later, but this is not recommended unless the input source of the index signal varies. Parameters set in this mode are:

- CAPx.TRIGGER 10: index mark of primary encoder
- CAPx.EDGE 1: rising edge
- CAPx.EVENT 0: ignore precondition

Also the capture engine is immediately enabled and is continuously triggered again.

Mode 4 is similar to Mode 0 (standard position capture), except that the re-enabling of the capture is done automatically. This mode can be used for the registration move.

6.6.6 CAP0.PLFB, CAP1.PLFB

General Information	
Type	R/O Parameter
Description	Reads captured position value.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, µm, custom units, 16-bit counts
Range	Full range of a signed 64 bit variable
Default Value	0
Data Type	S64
See Also	UNIT.PROTARY , UNIT.PLINEAR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CAN-open	20A0h /0	CAP0.PLFB	M_01-00-00-000
	20A1h /0	CAP0.PLFB	
	20A2h /0	CAP1.PLFB	
	20A3h /0	CAP1.PLFB	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	64	CAP0.PLFB	Yes	64 bit	Yes	M_01-03-00-000
	90	CAP1.PLFB				

Description

This parameter reads the captured position value scaled to actual set units. See UNIT.PROTARY or UNIT.PIN for these units.

6.6.7 CAP0.PREEDGE, CAP1.PREEDGE

General Information	
Type	NV Parameter
Description	Selects the capture precondition edge.
Units	N/A
Range	1 to 3
Default Value	1
Data Type	U8
See Also	CAP0.EDGE, CAP1.EDGE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CAN-open	3460h /7	CAP0.PREEDGE	M_01-00-00-000
	3460h /8	CAP1.PREEDGE	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	68	CAP0.PREEDGE	No	8 bit	No	M_01-03-00-000
	94	CAP1.PREEDGE				

Description

The precondition edge is monitored for rising edge, falling edge, or both. The event mode logic may ignore the precondition edge detection (trigger always uses edge detection).

The filtered trigger source has an identical feature controlled by CAP0.EDGE, CAP1.EDGE .

Value	Description
0	Reserved
1	Rising edge
2	Falling edge
3	Both edges

6.6.8 CAP0.PREFILTER, CAP1.PREFILTER

General Information	
Type	NV Parameter
Description	Sets the filter for the precondition input source.
Units	N/A
Range	0 to 2
Default Value	0
Data Type	U8
See Also	CAP0.FILTER, CAP1.FILTER
Start Version End Version	M_01-00-00-000 M_01-03-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	70	CAP0.PREFILTER	No	8 bit	No	M_01-03-00-000
	96	CAP1.PREFILTER				

Description

These parameters are not functional in M_01-03-00-000. In future releases, you can use DINx.FILTER to select a filter on the input channel.

Related Topics

DIN1.FILTER TO DIN7.FILTER

6.6.9 CAP0.PRESELECT, CAP1.PRESELECT

General Information	
Type	NVParameter
Description	Sets the precondition trigger.
Units	N/A
Range	0 to 11
Default Value	0
Data Type	U8
See Also	CAP0.TRIGGER, CAP1.TRIGGER
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3460h/9	CAP0.PRESELECT	M_01-00-00-000
	3460h /10	CAP1.PRESELECT	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	72	CAP0.PRESELECT	No	8 bit	No	M_01-03-00-000
	98	CAP1.PRESELECT				

Description

This parameter specifies the input signal for the precondition trigger.

Trigger Source	Input Name
0	General Input 1 (X7)
1	General Input 2 (X7)
2	General Input 3 (X7)
3	General Input 4 (X7)
4	General Input 5 (X8)
5	General Input 6 (X7)
6	General Input 7 (X7)
7	RS485 Input 1 (X9)
8	RS485 Input 2 (X9)
9	RS485 Input 3 (X9)
10	Primary Index

6.6.10 CAP0.STATE, CAP1.STATE

General Information	
Type	R/O Parameter
Description	Indicates whether or not trigger source was captured.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	74	CAP0.STATE	No	8 bit	No	M_01-03-00-000
	100	CAP1.STATE				

Description

When enabling the capture (CAP0.EN, CAP1.EN), this parameter is set to 0 until the next event is captured.

0 = Not captured or Capture Disabled

1 = Captured

6.6.11 CAP0.T, CAP1.T

General Information	
Type	R/O Parameter
Description	Reads time capture (if time capture was configured).
Units	ns
Range	N/A
Default Value	N/A
Data Type	U32
See Also	CAP0.MODE, CAP1.MODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	20A0h/0	CAP0.T	M_01-00-00-000
	20A1h/0	CAP0.T	
	20A2h/0	CAP1.T	
	20A3h/0	CAP1.T	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	76	CAP0.T	No	32 bit	No	M_01-03-00-000
	102	CAP1.T				

Description

If time capture was configured, the captured time is stored in this parameter. The reference time is the occurrence of the last MTS signal (recurring every 62.5 µs), so this is a purely drive internal time.

6.6.12 CAP0.TRIGGER, CAP1.TRIGGER

General Information	
Type	NV Parameter
Description	Specifies the trigger source for the position capture.
Units	N/A
Range	0 to 11
Default Value	0
Data Type	U8
See Also	CAP0.PRESELECT, CAP1.PRESELECT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3460h /1	CAP0.TRIGGER	M_01-00-00-000
	3460h /2	CAP1.TRIGGER	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	78	CAP0.TRIGGER	No	8 bit	No	M_01-03-00-000
	104	CAP1.TRIGGER				

Description

This parameter specifies the trigger source (capture input signal).

Trigger Source	Input Name
0	General Input 1
1	General Input 2
2	General Input 3
3	General Input 4
4	General Input 5
5	General Input 6
6	General Input 7
7	RS485 Input 1
8	RS485 Input 2
9	RS485 Input 3

Trigger Source	Input Name
10	Primary Index
11	Tertiary Index

6.7 CS Parameters

Controlled stop (CS) parameters set the values for the controlled stop process.

6.7.1 CS.DEC

General Information	
Type	NV Parameter
Description	Sets the deceleration value for the controlled stop process.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s^2, (custom units)/s^2, rad/s^2 Linear: counts/s^2, mm/s^2, µm/s^2, (custom units)/s^2
Range	Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s² 0.155 to 4,166,666.752 (custom units)/s² 0.012 to 5,235,987.968 rad/s² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s^2 0.031*MOTOR.PITCH to 833333.333*MOTOR.PITCH mm/s^2 30.994*MOTOR.PITCH to 833333333.333*MOTOR.PITCH µm/s^2 0.155 to 4,166,666.667 (custom units)/s^2
Default Value	Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s² 833.333 (custom units)/s² 1,047.2 rad/s² Linear: 715,840,000.000 counts/s² 166.714*MOTOR.PITCH MOTOR.PITCH mm/s² 166,714.191*MOTOR.PITCH MOTOR.PITCH µm/s² 833.571 (custom units)/s²
Data Type	Float
See Also	CS.VTHRESH , CS.TO , DRV.DIS, DIN1.MODE to DIN19.MODE, DRV.DISMODE, DRV.DISSOURCES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3440h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	106	Yes	64 bit	No	M_01-03-00-000

Description

This parameter sets the deceleration value for the controlled stop process.

6.7.2 CS.STATE

General Information	
Type	R/O Parameter
Description	Returns the internal status of the controlled stop process.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	CS.DEC , CS.VTHRESH , CS.TO DRV.DISMODE, DRV.DI-SSOURCES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3441h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	110	No	8 bit	No	M_01-03-00-000

Description

CS.STATE returns the internal state machine value of the controlled stop.

0 = controlled stop is not occurring.

1 = controlled stop is occurring

6.7.3 CS.TO

General Information	
Type	NV Parameter
Description	Sets the time value for the drive velocity to be within CS.VTHRESH .
Units	ms
Range	1 to 30,000 ms
Default Value	6 ms
Data Type	Integer
See Also	CS.DEC , CS.VTHRESH , CS.STATE, DRV.DIS, DIN1.MODE to DIN19.MODE, DRV.DISMODE, DRV.DISSOURCES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3440h/3	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	112	No	32 bit	No	M_01-03-00-000

Description

CS.TO is the time value for the drive velocity to be within CS.VTHRESH before the drive disables.

Example

Set time value to 100 ms:

```
-->CS.TO 100
```

6.7.4 CS.VTHRESH

General Information	
Type	NV Parameter
Description	Sets the velocity threshold for the controlled stop.
Units	rpm, rps, deg/s, custom units/s
Range	Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 µm/s 0.000 to 1,250.000 custom units/s
Default Value	5 rpm
Data Type	Float
See Also	CS.DEC , CS.TO , CS.STATE , DRV.DIS , DIN1.MODE to DIN19.MODE, DRV.DISMODE, DRV.DISSOURCES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT and CAN-open	3440h/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	114	Yes	low 32 bit word	No	M_01-03-00-000

Description

CS.VTHRESH is the velocity threshold for the controlled stop algorithm.

Example

Set velocity threshold for controlled stop at 100 rpm:

```
-->CS.VTHRESH 100
```

6.8 DIN Parameters

This section describes the DIN parameters.

6.8.1 DIN.ROTARY

General Information	
Type	R/O Parameter
Description	Reads the rotary knob value.
Units	N/A
Range	0 to 99
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	116	No	8 bit	No	M_01-03-00-000

Description

DIN.ROTARY reads the rotary knob value.

6.8.2 DIN.STATES

General Information	
Type	R/O Parameter
Description	Reads the digital input states.
Units	N/A
Range	0000000 to 1111111
Default Value	N/A
Data Type	String
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	118	No	8 bit	No	M_01-03-00-000

Description

DIN.STATES reads the states of the seven digital inputs. The leftmost bit represents digital input 1 (DIN1) and the rightmost bit represents digital input 7 (DIN7).

6.8.3 DIN1.FILTER TO DIN7.FILTER

General Information	
Type	R/W Parameter
Description	Filter mode for digital inputs 1 to 7.
Units	N/A
Range	0 to 3
Default Value	1 for DIN1 and DIN2 2 for DIN3 to DIN7
Data Type	Integer
See Also	N/A
Start Version	M_01-03-07-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?
Modbus	918	DIN1.FILTER	No	16 bit	No
	920	DIN2.FILTER			
	922	DIN3.FILTER			
	924	DIN4.FILTER			
	926	DIN5.FILTER			
	928	DIN6.FILTER			
	930	DIN7.FILTER			

Description

This parameter sets the digital input filter configuration for channel x when followed with the values defined below. DINx.FILTER retrieves this information when not followed by data.

Value	Description
DINX.FILTER 0	The drive digital input channel detects all input signals with an input pulse width of ≥ 40 ns (no filtering applied).
DINX.FILTER 1	The drive digital input channel detects all input signals with an input pulse width of ≥ 10.24 µs, ± 0.64 µs (fast filter applied).
DINX.FILTER 2	The drive digital input channel detects all input signals with an input pulse width of ≥ 163 µs, ± 10.24 µs (standard filter applied).
DINX.FILTER 3	The drive digital input channel detects all input signals with an input pulse width of ≥ 2.62 ms, ± 0.16384 ms (slow filter applied).

6.8.4 DIN1.INV to DIN7.INV

General Information	
Type	RW Parameter
Description	Sets the indicated polarity of a digital input mode.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Boolean
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Sign
Modbus	120	DIN1.INV	No	8 bit	No
	130	DIN2.INV			
	140	DIN3.INV			
	150	DIN4.INV			
	160	DIN5.INV			
	170	DIN6.INV			
	180	DIN7.INV			

Description

Sets the indicated polarity of a digital input mode.

Example

```
DIN1.INV = 0 : Input is active high.
DIN1.INV = 1 : Input is active low.
```

6.8.5 DIN1.MODE to DIN19.MODE

General Information	
Type	R/W Parameter
Description	Sets the digital input modes.
Units	N/A
Range	0 to 23
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CAN-open	3562h /0	DIN1.MODE	M_01-00-00-000
	3565h /0	DIN2.MODE	
	3568h /0	DIN3.MODE	
	356Bh /0	DIN4.MODE	
	36F6h /0	DIN5.MODE	
	36F9h /0	DIN6.MODE	
	36FCh /0	DIN7.MODE	
	60FDh /0	DIN1.MODE TO DIN7.MODE	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Sig
Modbus	122	DIN1.MODE	No	16 bit	No
	132	DIN2.MODE			
	142	DIN3.MODE			
	152	DIN4.MODE			
	162	DIN5.MODE			
	172	DIN6.MODE			
	182	DIN7.MODE			

AKD SynqNet Information	
Range	0

AKD BASIC Information	
Range	1 to 19

Description

This parameter sets the functionality of the digital inputs 1 through 7. Digital inputs and corresponding X7 and X8 pin connectors are described in the *AKD Installation Manual*, section 8.16.4, Digital Inputs. The table below summarizes the digital input modes; for detailed descriptions of each mode, see 1 Digital Inputs and Outputs.

DINx.MODE	Description	Task
0	No function; off	0 - None
1	Fault reset	1 - Background
7	Reserved	7 - Background
12	Reserved	12 - None
14	Reserved	14 - None
18	Positive limit switch	18 - 4 kHz
19	Negative limit switch	19 - 4kHz

6.8.6 DIN1.STATE TO DIN7.STATE

General Information	
Type	R/O Parameter
Description	Reads a specific digital input state.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Sign
Modbus	128	DIN1.STATE	No	8 bit	No
	138	DIN2.STATE			
	148	DIN3.STATE			
	158	DIN4.STATE			
	168	DIN5.STATE			
	178	DIN6.STATE			
	188	DIN7.STATE			

Description

DIN1.STATE to DIN7.STATE reads the state of one digital input according to the number identified in the command.

6.8.7 DIN9.STATE to DIN11.STATE

General Information	
Type	NV Parameter
Description	Shows on selected pin if signal is high or low.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-00-000

Description

This parameter allows the user to see the actual level of the input signal, when the IO is set to input mode. Parameter value is 0 if signal is low and 1 if signal is high. DIOx.INV can affect the value in this register.

This parameter can be read at any time. The value is only guaranteed to correspond to the output on the X9 connector when DRV.EMUEMODE is set to 10 and the DIOX.DIR is 0.

6.9 DIO Parameters

This section describes the DIO parameters.

6.9.1 DIO9.INV to DIO11.INV

General Information	
Type	NV Parameter
Description	Inverting the output voltage of the IO, when in the output direction.
Units	NA
Range	0 to 1
Default Value	0
Data Type	U8
Start Version	M_01-05-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?
Modbus	1192	DIO10.INV	No	8 bit	No
	1196	DIO11.INV			
	1200	DIO9.INV			

Description

This parameter changes the logic sense of the differential input/output signals. When false, a logic 1 occurs when the + signal is higher than the – signal. When true, a logic 1 occurs when the – signal is higher than the + signal.

The drive output parameters DOUTx.STATE and DOUTx.STATEU are not affected by changes in this parameter. The drive input parameters DINx.STATE will be affected.

This parameter can be set at any time. It will be ignored unless DRV.EMUEMODE is set to 10.

6.9.2 DIO9.DIR to DIO11.DIR

General Information	
Type	NV Parameter
Description	Changing direction of the IOs from the X9 connector.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	U8
Start Version	M_01-05-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?
Modbus	1190	DIO10.DIR	No	8 bit	No
	1194	DIO11.DIR			
	1198	DIO9.DIR			

Description

This parameter changes the direction of the general purpose IO from the X9 connector. If DIOx.DIR is set 0 then the IO configured as an input, while if DIOx.DIR is 1 the IO is configured as an output.

DIO9.DIR controls pins 1 and 2

DIO10.DIR controls pins pin 4 and 5

DIO11.DIR controls pins pin 7 and 8.

This parameter can be set at any time. It will be ignored unless DRV.EMUEMODE is set to 10.

6.10 DOUT Parameters

This section describes the DOUT parameters.

6.10.1 DOUT.RELAYMODE

General Information	
Type	R/W Parameter
Description	Indicates faults relay mode.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	192	No	8 bit	No	M_01-03-00-000

Description

DOUT.RELAYMODE indicates the faults relay mode as follows:

If DOUT.RELAYMODE= 0 and faults exist, then the relay is open.

If DOUT.RELAYMODE= 0 and faults do not exist, then the relay is closed.

If DOUT.RELAYMODE = 1 and the drive is disabled, then the relay is open.

If DOUT.RELAYMODE = 1 and the drive is enabled, then the relay is closed.

6.10.2 DOUT.STATES

General Information	
Type	R/O Parameter
Description	Reads the state of the two digital outputs.
Units	N/A
Range	0 to 11
Default Value	N/A
Data Type	String
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	194	No	8 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

DOUT.STATES reads the states of the two digital outputs. The rightmost bit represents DOUT2 and the leftmost bit represents DOUT1.

6.10.3 DOUT8.MODE to DOUT11.MODE

General Information	
Type	NV Parameter
Description	Sets the digital output mode.
Units	N/A
Range	0 to 17
Default Value	0
Data Type	Integer
See Also	DOUT1.PARAM AND DOUT2.PARAM
Start Version	M_01-04-02-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	196	DOUT1.MODE	No	8 bit	No	M_01-03-00-000
	206	DOUT2.MODE				

AKD BASIC Information	
Range	8 to 11

Description

DOUTx.MODE sets the functionality of the digital outputs. The table below summarizes the digital output modes; for detailed descriptions of each mode, see 1 Digital Inputs and Outputs.

DOUTx.MODE	Description
0	User (default = 0)
8	Drive enabled
9	Reserved
10	Motor brake
11	Drive produced fault

6.10.4 DOUT1.PARAM AND DOUT2.PARAM

General Information	
Type	NV Parameter
Description	Sets extra parameters for the digital outputs.
Units	N/A
Range	DOUT1.PARAM: –357,913.941 to 357,913.941 DOUT2.PARAM: –79,164,837,199.872 to 79,164,837,199.856
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	198	DOUT1.PARAM	Yes	64 bit	Yes	M_01-03-00-000
	208	DOUT2.PARAM				

Description

DOUT1.PARAM and DOUT2.PARAM set the extra parameter needed for the digital outputs calculations, respectively.

6.10.5 DOUT1.STATE AND DOUT2.STATE

General Information	
Type	R/O Parameter
Description	Reads the digital output state.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	202	DOUT1.MODE	No	8 bit	No	M_01-03-00-000
	212	DOUT2.MODE				

Description

DOUT1.STATE and DOUT2.STATE read the state of one digital output according to the value stated in the command.

6.10.6 DOUT1.STATEU AND DOUT2.STATEU

General Information	
Type	R/W Parameter
Description	Sets the state of the digital output node.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-01-01-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	204	DOUT1.STATEU	No	8 bit	No	M_01-03-00-000
	214	DOUT2.STATEU				

Description

DOUT1.STATEU and DOUT2.STATEU set the state of the digital output node as follows:

0 = deactivated

1 = activated

DOUT1.STATEU and DOUT2.STATEU are used when DOUT8.MODE to DOUT11.MODE = 0 (user mode).

6.10.7 DOUT9.STATE to DOUT11.STATE

General Information	
Type	NV parameter
Description	Shows on selected pin if signal is high or low.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-00-000

Description

This parameter allows the user to see the actual level of the output signal, when the IO is set to output mode. Parameter value is 0 if signal is low and 1 if signal is high. DIOx.INV can affect the signals driven onto the X9 connector.

This parameter can be read at any time. The value is only guaranteed to correspond to the output on the X9 connector when DRV.EMUEMODE is set to 10 and the DIOX.DIR is 0.

6.10.8 DOUT9.STATEU to DOUT11.STATEU

General Information	
Type	NV Parameter
Description	Allows user to set level of selected pin to high or low.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-00-000

Description

This parameter allows the user to set the level of the output signal, when the IO is set to output mode. Parameter value is 0 if signal is low and 1 if signal is high. DIOx.INV can affect the signals driven onto the X9 connector.

This parameter can be written at any time. The value is only guaranteed to correspond to the output on the X9 connector when DRV.EMUEMODE is set to 10 and the DIOX.DIR is 0.

Example

The following settings set the direction for the differential signals on pin 4 and 5, so that the output will have a high level signal.

First set the following settings:

```
DRV.EMUEMODE 10
DIO10.DIR 1
DOUT10.STATEU 1
```

Then change the level of the signal:

```
DOUT.STATEU 0
```

or

```
DIO10.INV
```

Note: Inverting the signal will also alter the signal in input mode.

6.11 DRV Parameters

This section describes the DRV parameters.

6.11.1 DRV.ACC

General Information	
Type	NV Parameter
Description	Describes the acceleration ramp for the velocity loop.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s², (custom units)/s² , rad/s² Linear: counts/s² , mm/s² , µm/s² , (custom units)/s²
Range	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s² 0.155 to 4,166,666.752 (custom units)/s² 0.012 to 5,235,987.968 rad/s² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s² 30.995*MOTOR.PITCH to 2,147,483.647*MOTOR.PITCH µm/s² 0.155 to 2,147,483.647 (custom units)/s²
Default Value	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s² 833.333 (custom units)/s² 1,047.2 rad/s² Linear: 715,840,000.000 counts/s² 166.714*MOTOR.PITCH mm/s² 166,714.191*MOTOR.PITCH µm/s² 833.571 (custom units)/s²
Data Type	Float
See Also	DRV.DEC , UNIT.ACCLINEAR , UNIT.ACCROTARY
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3501h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	216	Yes	64 bit	No	M_01-03-00-000

Description

Describes the acceleration ramp for the velocity central loop.

6.11.2 DRV.ACTIVE

General Information	
Type	R/O Parameter
Description	Reads the enable status of an axis.
Units	N/A
Range	0, 1, 3
Default Value	N/A
Data Type	Integer
See Also	DRV.EN ,DRV.DISSOURCES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	220	No	8 bit	No	M_01-03-00-000

Description

DRV.ACTIVE reads the enable status of an axis as follows:

- DRV.ACTIVE = 0 drive disabled
- DRV.ACTIVE = 1 drive enabled
- DRV.ACTIVE = 3 drive enabled and in dynamic brake mode

There is no state 2.

When the drive is in state 3, the drive display shows a blinking decimal point. Additionally, if the drive is in state 3 the Parameter Load/Save view does not allow you to download a parameter file.

If an axis is not enabled (DRV.ACTIVE is 0), but DRV.EN is 1 and the hardware enable is high, read the value of DRV.DISSOURCES to query the reason that the drive is not enabled.

6.11.3 DRV.BLINKDISPLAY

General Information	
Type	Command
Description	Causes the display to blink for 10 seconds.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	N/A
Start Version	M_01-00-00-000

Description

DRV.BLINKDISPLAY causes the drive display located on the front of the drive to blink for 10 seconds.

This command allows the user to identify the drive that is currently communicating with WorkBench.

6.11.4 DRV.CLRFAULTHIST

General Information	
Type	Command
Description	Clears the fault history log in the NV.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	DRV.FAULTHIST
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	222	No	Command	No	M_01-03-00-000

Description

DRV.CLRFAULTHIST clears the fault history from the nonvolatile memory of the drive.

This command erases all faults returned by DRV.FAULTHIST.

6.11.5 DRV.CLRFAULTS

General Information	
Type	Command
Description	Tries to clear all active faults in the drive.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	DRV.FAULTS, DRV.EN , DRV.DIS
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	224	No	Command	No	M_01-03-00-000

Description

When DRV.CLRFAULTS is sent, the drive will try to clear all active faults. When a fault occurs, the fault is registered in the drive fault handler. DRV.CLRFAULTS clears the fault from the drive fault handler. However, if the fault still exists in the system, DRV.CLRFAULTS fails and the fault is re-registered in the fault handler.

If the DRV.CLRFAULTS succeeds, then the reply to DRV.FAULTS states that no faults exist. If the condition that triggered the fault is still present, the fault condition will remain. See1 Fault and Warning Messages for details regarding the behavior of individual faults.

Note that executing a drive disable (DRV.DIS) followed by a drive enable (DRV.EN) has the same effect as executing DRV.CLRFAULTS.

6.11.6 DRV.CMDSOURCE

General Information	
Type	NV Parameter
Description	Sets the command source (service, fieldbus, analog input, gearing, digital, or Bode).
Units	N/A
Range	0 to 5
Default Value	0
Data Type	Integer
See Also	DRV.OPMODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	226	No	8 bit	No	M_01-03-00-000

AKD SynqNet Information	
Range	0

AKD BASIC Information	
Range	0, 3, 5

Description

DRV.CMDSOURCE specifies the source of the command to the drive. DRV.OPMODE sets the operation mode to the relevant control loop.

DRV.CMDSOURCE values can be set as follows:

Value	Description
0	Service, TCP/IP command
3	Analog command
5	Program command

If DRV.CMDSOURCE is set to 5 then DRV.OPMODE must be set to 3.

DRV.CMDSOURCE can be changed while the drive is enabled or disabled. If you use the terminal to change the operation mode, then it is recommended that you disable the drive before changing the command source.

⚠ WARNING	If you change DRV.CMDSOURCE from the terminal while the drive is enabled, the system may experience a step change in command.

Example

To set the command source to the TCP/IP channel and the operation mode to velocity:

```
-->DRV.CMDSOURCE 0
-->DRV.OPMODE 1
```

6.11.7 DRV.DBILIMIT

General Information	
Type	NV Parameter
Description	Sets the maximum amplitude of the current for dynamic braking.
Units	Arms
Range	0 to minimum of drive peak current (DRV.IPEAK) and motor peak current (MOTOR.IPEAK).
Default Value	Minimum of drive continuous current (DRV.ICONT) and motor continuous current (MOTOR.ICONT).
Data Type	Float
See Also	DRV.DISMODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3444h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	228	No	32 bit	No	M_01-03-00-000

Description

This parameter sets the maximum amplitude of the current for dynamic braking.

Example

Setting DRV.DBILIMIT to 2 limits the dynamic brake current to 2 Arms.

6.11.8 DRV.DEC

General Information	
Type	NV Parameter
Description	Sets the deceleration value for the velocity loop.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s², (custom units)/s², rad/s² Linear: counts/s², mm/s², µm/s², (custom units)/s²
Range	Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s² 0.155 to 4,166,666.752 (custom units)/s² 0.012 to 5,235,987.968 rad/s² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s² 30.994*MOTOR.PITCH to 833,333,333.333*MOTOR.PITCH µm/s² 0.155 to 4,166,666.667 (custom units)/s²
Default Value	Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s² 833.333 (custom units)/s² 1,047.2 rad/s² Linear: 715,840,000.000 counts/s² 166.71*MOTOR.PITCH4MOTOR.PITCH mm/s² 166,714.191*MOTOR.PITCHMOTOR.PITCH µm/s² 833.571 (custom units)/s²
Data Type	Float
See Also	DRV.ACC , UNIT.ACCROTARY , UNIT.ACCLINEAR , DRV.OPMODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3522h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	230	Yes	64 bit	No	M_01-03-00-000

Description

DRV.DEC sets the deceleration value for the velocity loop command (VL.CMDU) and for the analog

velocity command (AIN.VALUE). The operation mode (DRV.OPMODE) must be set to velocity mode for this command to function.

6.11.9 DRV.DIR

General Information	
Type	R/W Parameter
Description	Changes drive direction.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	352Ah/0	M_01-00-00-000

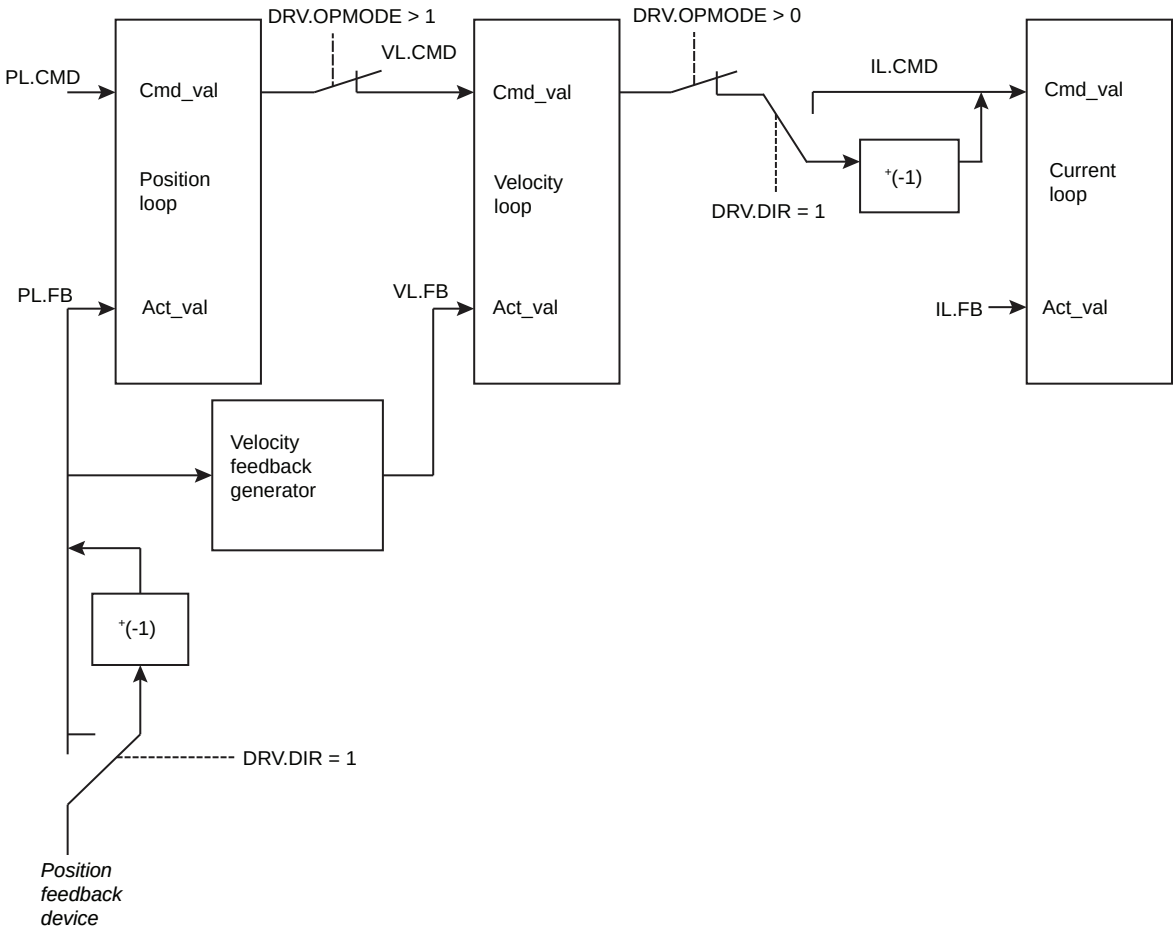
Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	234	No	8 bit	No	M_01-03-00-000

Description

DRV.DIR changes the direction of the motor by changing the algebraic sign of the current command and position feedback value according to the figure below.

Note the following when using DRV.DIR:

- You can only change the DRV.DIR command when the drive is disabled.
- The drive status changes to "Axis not homed" as soon as the DRV.DIR parameter changes value (see DRV.MOTIONSTAT).
- You must verify the settings of the hardware limit switches. If necessary, switch the positive and negative hardware limit switches by swapping the wires at the digital inputs.


DRV.OPMODE > 1
VL.CMD
DRV.OPMODE > 0
IL.CMD
PL.CMD
Cmd_val
Position loop
Act_val
Cmd_val
Velocity loop
Act_val
Cmd_val
Current loop
Act_val
DRV.DIR = 1
+(-1)
PL.FB
VL.FB
IL.FB
Velocity feedback generator
+(-1)
DRV.DIR = 1
Position feedback device

6.11.10 DRV.DIS

General Information	
Type	Command
Description	Disables the axis (software).
Units	N/A
Range	N/A
Default Value	Analog drive software enabled. All other types of drive software disabled.
Data Type	N/A
See Also	DRV.EN , DRV.DISSOURCES , DRV.ACTIVE , DRV.DISMODE, DRV.DISTO
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3443h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	236	No	Command	No	M_01-03-00-000

Description

DRV.DIS issues a software disable to the drive. The method by which the drive will be disabled (either immediately or with a ramp down first) is controlled by DRV.DISMODE.

By querying the value of DRV.ACTIVE , you can check whether the drive is currently enabled or disabled.

By querying the value of DRV.DISSOURCES , you can check whether the software enable bit is high (software enabled was issued by executing DRV.EN) or the software enable bit is low (software disable was issued by executing DRV.DIS).

If DRV.DIS is commanded the emergency timeout is started. If the drive does not disable or activate dynamic brake within DRV.DISTO , fault 703 is reported.

6.11.11 DRV.DISSOURCES

General Information	
Type	R/O Parameter
Description	Returns the possible reason for a drive disable.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	DRV.ACTIVE, DRV.FAULTS, DRV.EN, DRV.DIS
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	240	No	16 bit	No	M_01-03-00-000

Description

DRV.DISSOURCES is a bitwise parameter that returns the status of possible causes of a drive disable. If this parameter is 0, then the drive is enabled.

The return value specific bits are as follows:

Bit	Status and Response
0	Software disable (execute DRV.EN to issue software enable)
1	Fault exists (read DRV.FAULTS to get the active faults)
2	Hardware disable (remote enable input is low)
3	In-rush disable (the in-rush relay is opened)
4	Initialization disable (the drive did not finish the initialization process)
5	Controlled stop disable from a digital input.

Related Topics

1 Controlled Stop

6.11.12 DRV.DISTO

General Information	
Type	R/W Parameter
Description	Sets the emergency time-out
Units	ms
Range	0 to 120,000 ms
Default Value	1,000 ms
Data Type	U32
See Also	DRV.DIS , DRV.DI-SMODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3445h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	242	No	32 bit	No	M_01-03-00-000

Description

This timer starts when DRV.DIS is issued (regardless of the DRV.DIS origin). After this time-out elapses, the actual state of the drive is compared to the DRV.DISMODE setting. If the actual state does not match the DRV.DISMODE setting, a fault is reported and the hardware immediately executes the DRV.DISMODE setting (for instance, disable or activate dynamic brake). Setting DRV.DISTO to 0 will disable the timeout.

6.11.13 DRV.EMUEDIR

General Information	
Type	R/W Parameter
Description	Sets the direction of the emulated encoder output (EEO) signal.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	DRV.EMUEMODE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3493h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	244	No	8 bit	No	M_01-03-00-000

Description

This parameter allows the user to change the direction of the emulated encoder output. DRV.DIR also affects the output direction (through an XOR, "exclusive or", operation). The drive uses DRV.DIR and DRV.EMUEDIR to decide the direction of the emulated encoder output. If DRV.DIR and DRV.EMUEDIR have the same value, then DRV.EMUEDIR is set to 0 (meaning an increase in the motor feedback will result an increase of the encoder emulation output and vice-versa). If these parameters have different values, then DRV.EMUEDIR is set to 1 (meaning an increase in the motor feedback will result in a decrease of the encoder emulation output and vice-versa).

6.11.14 DRV.EMUEMODE

General Information	
Type	R/W Parameter
Description	Sets the mode of the emulated encoder output (EEO) connector.
Units	N/A
Range	0 to 11
Default Value	0
Data Type	Integer
See Also	DRV.EMUERES , DRV.EMUEZOFFSET , DRV.EMUE-MTURN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3534h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	246	No	16 bit	No	M_01-03-00-000

Description

When the emulated encoder output (EEO) is configured to generate an absolute index pulse (DRV.EMUEMODE is 2, 7 or 9) this parameter and DRV.EMUEZOFFSET define the location of the Z pulse. DRV.EMUEMTURN is used to define which turn of the position range the Z pulse is located. DRV.EMUEZOFFSET is used to define the position of the Z pulse within one revolution.

This parameter sets the EEO connector to act as either an input or output as follows.

Setting	Function
0 (recommended)	Input (see FB2.MODE to select the type of inputs the secondary feedback will accept)
1	EEO Output, A/B with once per rev index
2	EEO Output, A/B with absolute index pulse.
3	Input, A/B signals (Deprecated)
4	Input, step and direction signals (Deprecated)
5	Input, CW/CCW (Up/Down) Signals (Deprecated)
6	Step/Dir with one Z-pulse/rev
7	Step/Dir with one absolute Z-pulse (depends on DRV.EMUEOFFSET and DRV.EMUETURN)
8	CW/CCW output with one Z-pulse/rev
9	CW/CCW output with one absoulte Z-pulse (depends on DRV.EMUEOF-FSET and DRV.EMUETURN)
10	Allows the X9 connector to be used as a General Purpose I/O or SynqNet fieldbus controlled I/O (See DIO9.DIR to DIO11.DIR)
11	FB3 Input (Tertiary feedback is reported with FB3.P). Use FB3.MODE to select the feedback type.

Modes 3 to 5 are backwards compatible but deprecated. Refer to FB2.MODE and FB2.SOURCE instead.

Note: If you are using multi-turn or single tune absolute feedback devices the Z pulse from generated by the EEO will always be aligned with the same mechanical position of the of the primary feedback position. If you are using an incremental feedback device then the origin of the primary feedback is not at the same mechanical position each time the drive powers up.

6.11.15 DRV.EMUEMTURN

General Information	
Type	R/W Parameter
Description	Defines the location of the index pulse on the EEO (emulated encoder output) when DRV.EMUEMODE=2.
Units	revolutions
Range	0 to 4,294,967,295
Default Value	0
Data Type	Integer
See Also	DRV.EMUEMODE , DRV.EMUERES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3491h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	248	No	32 bit	No	M_01-03-00-000

Description

When the emulated encoder output (EEO) is configured to generate an absolute index pulse (DRV.EMUEMODE is 2, 7 or 9) this parameter and DRV.EMUEZOFFSET define the location of the Z pulse. DRV.EMUEMTURN is used to define which turn of the position range the Z pulse is located. DRV.EMUEZOFFSET is used to define the position of the Z pulse within one revolution.

Note: If you are using multi-turn or single tune absolute feedback devices the Z pulse from generated by the EEO will always be aligned with the same mechanical position of the of the primary feedback position. If you are using an incremental feedback device then the origin of the primary feedback is not at the same mechanical position each time the drive powers up.

6.11.16 DRV.EMUERES

General Information	
Type	R/W Parameter
Description	Sets the resolution of the EEO (emulated encoder output).
Units	lines/rev (when DRV.EMUEMODE = 1, 2, or 3) counts/rev (when DRV.EMUEMODE = 4 or 5)
Range	0 to 16,777,215 lines per revolution
Default Value	0 lines per revolution
Data Type	Integer
See Also	DRV.EMUEMODE
Start Version	M_01-00-00-000 (resolution increased from 65,535 to 16,777,215 in M_01-04-00-000)

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3535h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	250	No	32 bit	No	M_01-03-00-000

Description

This parameter sets the emulated encoder (EEO) resolution. DRV.EMUERES also defines how many lines are output for one revolution of the primary feedback (when this port is configured as an output), or how many lines will be considered a full revolution of the handwheel (when this port is configured as an input).

6.11.17 DRV.EMUEZOFFSET

General Information	
Type	R/W Parameter
Description	Sets the location of the EEO (emulated encoder output) index pulse (when DRV.EMUEMODE=1).
Units	1/65536 rev
Range	0 to 65535 rev
Default Value	0 rev
Data Type	Integer
See Also	DRV.EMUEMODE , DRV.EMUEMTURN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3537h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	252	No	16 bit	No	M_01-03-00-000

Description

When emulated encoder output (EEO) multiturn is selected (DRV.EMUEMODE =1), this parameter is used by itself to define the position if the Z pulse within one revolution.When the primary feedback position (within a revolution) equals this value, an index pulse will output. Also, if DRV.EMUEMODE=1 then this parameter is used in conjunction with DRV.EMUEMTURN.

When the EEO is configured to generate an absolute index pulse (DRV.EMUEMODE is 2, 7 or 9) this parameter and DRV.EMUEZOFFSET define the location of the Z pulse. DRV.EMUEMTURN is used to define which turn of the position range the Z pulse is located and DRV.EMUEZOFFSET is used to define the position of the Z pulse within one revolution.

Note: If you are using multi-turn or single tune absolute feedback devices the Z pulse from generated by the EEO will always be aligned with the same mechanical position of the of the primary feedback position. If you are using an incremental feedback device then the origin of the primary feedback is not at the same mechanical position each time the drive powers up.

6.11.18 DRV.EN

General Information	
Type	Command
Description	Enables the axis (software).
Units	N/A
Range	N/A
Default Value	Analog drive software is enabled. All other types of drive software are disabled.
Data Type	N/A
See Also	DRV.DIS , DRV.DISSOURCES DRV.ACTIVE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	254	No	Command	No	M_01-03-00-000

Description

DRV.EN issues a software enable to the drive. You can query the value of DRV.ACTIVE to check whether the drive is currently enabled or disabled.

You can also query the value of DRV.DISSOURCES to check whether the software enable bit is high (software enabled was issued by executing DRV.EN) or the software enable bit is low (software disable was issued by executing DRV.DIS). If the drive software enable bit is low and DRV.EN is executed, then drive faults are automatically cleared during the software enable process.

6.11.19 DRV.FAULT1 to DRV.FAULT10

General Information	
Type	R/O
Description	Location of fault codes for any active fault conditions.
Units	N/A
Range	Any supported fault code or 0.
Default Value	N/A
Data Type	Integer
Start Version	tbd

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?
Modbus	954	DRV.FAULT1	No	16 bit	No
	956	DRV.FAULT2			
	958	DRV.FAULT3			
	960	DRV.FAULT4			
	962	DRV.FAULT5			
	964	DRV.FAULT6			
	966	DRV.FAULT7			
	968	DRV.FAULT8			
	970	DRV.FAULT9			
	972	DRV.FAULT10			

Description

These parameters are holding registers where any active faults will be kept. A value of zero represents that no fault is present. Non-zero values correspond to specific fault codes in the drive (see fault and warning messages). The registers are populated in the order of when the fault occurs (DRV.FAULT1, DRV.FAULT2, DRV.FAULT3, and so on).

Notes:

- If DRV.FAULT1 value is 0, then the drive does not have any faults.
- Only active faults are shown. This is not a fault history.
- These registers are an alternative to the string type parameter DRV.FAULTLIST so that fieldbusses and AKD BASIC users have easier access to the details of the faults in the drive.
- Warnings are not shown in the registers, only faults.

Related Topics

Modbus | DRV.ACTIVE | DRV.WARNING1 to DRV.WARNING10

6.11.20 DRV.HANDWHEEL

General Information	
Type	R/O Parameter
Description	Reads the EEO input value.
Units	1/4,294,967,296 rev
Range	0 to 4,294,967,295 rev
Default Value	0 rev
Data Type	Integer
See Also	DRV.EMUERES , DRV.EMU-EMODE
Start Version End Version	M_01-00-00-000 M_01-03-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	2050h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	258	No	32 bit	No	M_01-03-00-000

Description

When the EEO is selected as an input (DRV.EMUEMODE =3,4,5), this parameter reads the EEO value (where 4,294,967,296 is a full revolution, then the value rolls over). DRV.EMUERES defines the how many counts constitute a revolution on the EEO. This parameter represents the feedback 2 positions when feedback 2 is configured to be active.

When secondary feedback is selected (DRV.EMUEMODE is 0 and FB2.SOURCE = 1 (X9), or FB2.SOURCE = 2 (X7)), this parameter represents the secondary feedback position (where 4,294,967,296 is a full revolution, then the value rolls over). FB2.ENCRES defines how many counts define a revolution for the secondary feedback.

6.11.21 DRV.HANDWHEELSRC

General Information	
Type	NV Parameter
Description	Selects the feedback for handwheel operation.
Units	None
Range	2-3
Default Value	2
Data Type	U8
See Also	N/A
Start Version	M_01-05-08-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1224	No	8 bit	No

Description

This command sets the feedback which will be used as the handwheel source. If the selected Feedback is incompatible with the selected emulated encoder mode, a warning will be displayed.

Feedback 3 is only supported on drives with model numbers similar to AKD-x-xxxxx-NBxx-xxxx and will only work with Endat 2.2 multiturn encoder.

6.11.22 DRV.HWENABLE

General Information	
Type	R/O
Description	Status of the hardware enable .
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
Start Version	tbd

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1054	No	8 bit	No

Description

Status of the Hardware Enable.

0 - not enabled

1 - enabled

Notes: This parameter reflects the status of the hardware enable only, not if the status of the power stage. The status of the power stage enable is determined by DRV.ACITVE.

Related Topics

DRV.DISSOURCES | DRV.ACTIVE

6.11.23 DRV.ICONT

General Information	
Type	R/O Parameter
Description	Reads the continuous rated current value.
Units	Arms
Range	N/A
Default Value	N/A
Data Type	Float
See Also	DRV.IPEAK
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	262	No	32 bit	Yes	M_01-03-00-000

Description

DRV.ICONT returns the drive continuous rated current in Arms.

The value of the continuous current is read automatically on drive boot from the power EEP-ROM of the drive. This value cannot be modified.

6.11.24 DRV.IPEAK

General Information	
Type	R/O Parameter
Description	Reads the peak rated current value.
Units	Arms
Range	N/A
Default Value	N/A
Data Type	Float
See Also	DRV.ICONT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	264	No	32 bit	Yes	M_01-03-00-000

Description

DRV.IPEAK returns the drive peak rated current in Arms.

The value of the peak current is read automatically on drive boot from the power EEPROM of the drive. This value cannot be modified.

6.11.25 DRV.NAME

General Information	
Type	NV Parameter
Description	Sets and reads the name of the drive.
Units	N/A
Range	N/A
Default Value	No-Name
Data Type	String
See Also	N/A
Start Version	M_01-00-00-000

Description

You can assign a unique name to any drive under the following conditions:

- Only use ASCII characters
- Max length of 20 characters
- No spaces in name

This name is one way to identify the drive in a multiple drive network (for instance, in a TCP/IP network on which multiple drives reside).

From the terminal screen, DRV.NAME returns the name of the drive as ASCII characters.

6.11.26 DRV.NVLOAD

General Information	
Type	R/O Parameter
Description	Loads all data from the NV memory of the drive into the RAM parameters.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	DRV.NVLOAD DRV.NVLIST
Start Version	M_01-00-00-000

Description

DRV.NVLOAD loads all data from the NV memory of the drive into the RAM parameters.

6.11.27 DRV.NVSAVE

General Information	
Type	Command
Description	Saves the drive parameters from the RAM to the NV memory.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	DRV.RSTVAR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	1010h/1 35EBh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	938	No	Command	No

Description

DRV.NVSAVE saves the current drive parameter values from the RAM to the NV memory.

The drive parameters that were saved to the NV are read from the NV on the next drive boot, causing the values to be automatically set to the saved values on every drive boot.

Executing DRV.RSTVAR does not modify the values of the NV, but instead sets the drive values in RAM to their defaults.

6.11.28 DRV.OPMODE

General Information	
Type	NV Parameter
Description	Sets the drive operation mode (current, velocity, or position).
Units	N/A
Range	0 to 2
Default Value	0
Data Type	Integer
See Also	DRV.CMDSOURCE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	35B4h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	270	No	8 bit	No	M_01-03-00-000

Description

DRV.OPMODE specifies the operation mode of the drive. You must also use DRV.CMDSOURCE to set the source of the command to the drive.

The operation mode values can be set as follows:

Mode	Description
0	Current (torque) operation mode
1	Velocity operation mode
2	Position operation mode
3	Position mode

DRV.OPMODE can be changed while the drive is enabled or disabled. If you are using the terminal to change the operation mode, then it is recommended that you disable the drive before changing the operation mode. If you change the operation mode from the terminal while the drive is enabled, the system may experience a step change in demand.

Example

Set the source of the command to a TCP/IP channel and the desired operation mode to velocity:

```
-->DRV.CMDSOURCE 0
-->DRV.OPMODE 1
```

6.11.29 DRV.RSTVAR

General Information	
Type	Command
Description	Sets default values in the drive without re-booting the drive and without resetting the NV memory.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	272	No	Command	No	M_01-03-00-000

Description

DRV.RSTVAR causes the drive to return to the default values without the need to re-boot the drive first and without resetting the NV memory. Use DRV.RSTVAR to return to the default settings and recover a working drive.

6.11.30 DRV.SETUPREQBITS

General Information	
Type	R/O Parameter
Description	Reads the bitwise set status of parameters that must be set before the drive can be enabled
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	DRV.SETUPREQLIST, MOTOR.AUTOSET
Start Version	M_01-00-00-000

Description

This parameter returns the bitwise set status of parameters that needs to be set up before the drive can be enabled. Only when this parameter returns 0 can the drive be enabled.

Parameter	Bits
IL.KP	0x00000001
MOTOR.IPEAK	0x00000002
MOTOR.ICONT	0x00000004
MOTOR.VMAX	0x00000008
MOTOR.POLES	0x00000010
MOTOR.PHASE	0x00000020

Please note that if MOTOR.AUTOSET is set to 1 (parameters automatically calculated from motor ID data), then all values in the list will be initialized from the feedback device. Otherwise, the parameters must be set manually.

6.11.31 DRV.STOP

General Information	
Type	Command
Description	This command stops all drive motion.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	35FEh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	274	No	Command	No	M_01-03-00-000

Description

This command stops all drive motion.

6.11.32 DRV.SWENABLE

General Information	
Type	R/W Parameter
Description	Controls whether power can flow to the motor.
Units	none
Range	0 to 1
Default Value	0
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1056	No	8 bit	No	M_01-05-11-000

Description

The following commands disable or enable the drive:

- 0 (disables the drive)
- 1 (enables the drive)

Before power can flow to the motor, the following must all be true:

1. Drive is not faulted.
2. DRV.SWENABLE* input (J4-6) is connected to I/O RTN.
3. DRV.SWENABLE Parameter is set to 1.

Related Topics

DRV.ACTIVE

6.11.33 DRV.TIME

General Information	
Type	R/W
Description	A continuous time counter in the drive.
Units	Milliseconds
Range	0 to 4294967295 (~ 49 days)
Default Value	N/A
Data Type	Integer
Start Version	tbd

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1058	No	32 bit	No

Description

A continuous time counter in the drive. The timer starts at zero and counts up until it rolls over. If a new value is written to the timer, it continues to count up starting at the written value. The DRV.TIME value is set to zero when the AKD BASIC is powered on.

Related Topics

DRV.RUNTIME | WHEN.DRV.TIME

6.11.34 DRV.WARNING1 to DRV.WARNING10

General Information	
Type	R/O
Description	Location of fault codes for any active warning conditions.
Units	N/A
Range	Any supported fault code or 0
Default Value	N/A
Data Type	Integer
Start Version	tbd

Description

These parameters are holding registers where any active warnings will be displayed. A value of zero represents that no warning is present. Non-zero values correspond to specific warning codes in the drive (see fault and warning messages). The registers are populated in the order of when the warning occurs (DRV.WARNING1, DRV.WARNING2, DRV.WARNING3, and so on).

Notes:

- If DRV.WARNING1 value is 0, then the drive does not have any faults.
- Only active warnings are shown. This is not a warning history.
- These registers are an alternative to the string type parameter DRV.WARNINGS so that fieldbuses and AKD BASIC user programs have integer-type parameters to access to the details of the warnings in the drive.
- Faults are not shown in the registers, only warnings.

Related Topics

DRV.FAULT1 to DRV.FAULT10 | Modbus

6.12 EGEAR Parameters

This section describes the EGEAR parameters.

6.12.1 EGEAR.ACCLIMIT

General Information	
Type	R/W
Description	EGEAR.ACCLIMIT sets the maximum acceleration.
Units	rpm/sec
Range	1 to 16,000,000 rpm/sec
Default Value	16,000,000 rpm/sec
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1060	Yes	low 32 bit word	No

Description

EGEAR.ACCLIMIT sets the maximum acceleration that will be commanded on the follower when EGEAR.ON is turned ON or the electronic gearing ratio (EGEAR.RATIO or EGEAR.PUSLESOUT / EGEAR.PUSLESIN) is increased. This maximum acceleration limit remains in effect until EGEAR.LOCK is achieved. Once EGEAR.LOCK is achieved the follower will follow the master with whatever acceleration or deceleration is required.

EGEAR.ACCLIMIT is independent of EGEAR.DECLIMIT. Each variable must be set, independently, to the appropriate value for the desired motion.

Example

```
' This example shows how to use EGEAR.ACCLIMIT
to limit
' acceleration and then make up the lost dis-
tance.
EGEAR.ACCLIMIT = 10000
EGEAR.RATIO = 1
DRV.SWENABLE = 1
EGEAR.ERROR = 0
EGEAR.TYPE = 0
EGEAR.ON = 1
While             EGEAR.LOCK = 0 : wend
  'wait for LOCK
MOVE.RELATIVEDIST = EGEAR.ERROR
MOVE.GOREL
```

6.12.2 EGEAR.DECLIMIT

General Information	
Type	R/W
Description	EGEAR.DECLIMIT sets the maximum deceleration that will be commanded on the follower when EGEAR.ON is turned OFF or the electronic gearing ratio (EGEAR.RATIO or EGEAR.PULSESOUT / EGEAR.PUSLESIN) is decreased.
Units	rpm / sec
Range	1 to 16,000,000 rpm/sec
Default Value	16,000,000 rpm/sec
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1062	Yes	low 32 bit word	No

Description

EGEAR.DECLIMIT sets the maximum deceleration that will be commanded on the follower when EGEAR.ON is turned OFF or the electronic gearing ratio (EGEAR.RATIO or EGEAR.PULSESOUT / EGEAR.PUSLESIN) is decreased. This maximum deceleration limit remains in effect until EGEAR.LOCK is achieved. Once EGEAR.LOCK is achieved the follower will follow the master with whatever acceleration or deceleration is required.

Set EGEAR.DECLIMIT prior to initiating EGEAR.ON.

Related Topics

EGEAR.ACCLIMIT |EGEAR.ERROR | EGEAR.LOCK

6.12.3 EGEAR.ERROR

General Information	
Type	R/W
Description	Indicates the amount of position deviation that has accumulated on the slave axis (in an electronic gearing application) as a result of the slave axis limiting its acceleration or deceleration while achieving velocity synchronization.
Units	Position counts
Range	TBD
Default Value	TBD
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1064	Yes	low 32 bit word	No	M_01-05-11-000

Description

EGEAR.ERROR is never automatically set to zero. It accumulates position deviation each time acceleration limiting is activated. Typically, set EGEAR.ERROR to zero before doing something that activates acceleration limiting.

The slave axis' acceleration or deceleration is limited to EGEAR.ACCLIMIT or EGEAR.DECLIMIT whenever:

1. Gearing is turned on or turned off.
2. Ratio is changed.
3. EGEAR.PULSESIN or EGEAR.PULSESOUT is changed.

Example

```
EGEAR.ACCLIMIT = 10000
EGEAR.PULSESIN = 1
EGEAR.PULSESOUT = 1
EGEAR.ERROR = 0
EGEAR.TYPE = 0
EGEAR.ON = 1
While              EGEAR.LOCK = 0 : wend
MOVE.RELATIVEDIST = EGEAR.ERROR
MOVE.GOREL              'catch up the position
lost while acceleration was being limited
```

Related Topics

EGEAR.LOCK | EGEAR.ACCLIMIT |EGEAR.DECLIMIT

6.12.4 EGEAR.LOCK

General Information	
Type	R/O
Description	EGEAR.LOCK indicates when the slave axis (follower axis) in an electronic gearing application has achieved velocity synchronization with the electronic gearing master.
Units	
Range	0 to 1
Default Value	
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1066	No	8 bit	No	M_01-05-11-000

Description

EGEAR.LOCK indicates when the slave axis (follower axis) in an electronic gearing application has achieved velocity synchronization with the electronic gearing master. EGEAR.ERROR contains the amount of position deviation accumulated while the slave axis was limiting its acceleration or deceleration.

The slave axis' acceleration or deceleration is limited to EGEAR.ACCLIMIT or EGEAR.DECLIMIT whenever:

1. Gearing is turned on or turned off.
2. Ratio is changed.
3. EGEAR.PULSESINor EGEAR.PULSESOUT is changed.

Example

```
EGEAR.ACCLIMIT = 10000
EGEAR.PULSESIN = 1
EGEAR.PULSESOUT = 0
EGEAR.TYPE = 0
EGEAR.ON = 1
While              EGEAR.LOCK = 0 : wend
MOVE.RELATIVEDIST = EGEAR.ERROR
MOVE.GOREL              'Catch up the position
lost while acceleration was being limited
```

Related Topics

EGEAR.ACCLIMIT | EGEAR.DECLIMIT | EGEAR.ERROR

6.12.5 EGEAR.ON

General Information	
Type	Command
Description	Starts the electronic gearing; active in opmode 2 (position) only.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
Start Version	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1068	No	8 bit	No	M_01-05-11-000

Description

The command EGEAR.ON starts the electronic gearing procedure according to the selected electronic gearing mode:

Mode	Description
0	Gearing Off
1	Gearing On

Related Topics

1 Electronic Gearing

6.12.6 EGEAR.PULSESIN

General Information	
Type	R/W
Description	Specifies the number of encoder counts used when specifying an exact electronic gearing ratio.
Units	Motor revolutions / Encoder Revolution
Range	-2,000 to 2,000
Default Value	1
Data Type	Integer
Start Version	TBD

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1070	No	16 bit	No	M_01-05-11-000

Description

EGEAR.PULSESIN is the number of encoder counts the motor moves for each EGEAR.PULSESOUT number of position counts. EGEAR.PULSESIN must be set more recently than EGEAR.RATIO in order to use exact electronic gearing.

Related Topics

IN.PLFBLIMIT | MOVE.MOVING | EGEAR.ON | FB2.SOURCE | FB2.ENCRES | FB2.MODE

6.12.7 EGEAR.PULSEOUT

General Information	
Type	R/W
Description	Specifies the number of position counts used in an exact electronic gearing ratio.
Units	Position counts
Range	-CountsPerRev/2 to CountsPerRev/2
Default Value	1
Data Type	Integer
Start Version	TBD

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1072	No	8 bit	Yes	M_01-05-11-000

Description

EGEAR.PULSESOUT is the number of position counts the motor moves for each EGEAR.PULSESIN number of encoder counts. EGEAR.PULSESOUT must be set more recently than EGEAR.RATIO in order to use exact electronic gearing.

Related Topics

EGEAR.PULSESIN | EGEAR.RATIO |EGEAR.ON | FB2.SOURCE | FB2.ENCRES | FB2.MODE

6.12.8 EGEAR.RATIO

General Information	
Type	TBD
Description	Sets the electronic gearing ratio (rev to rev) between the encoder shaft (master) and the motor shaft (slave).
Units	Motor revolutions / Encoder Revolution
Range	-2,000 to 2,000
Default Value	1
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1074	No	32 bit	No	M_01-05-11-000

Description

Ratio must be set more recently than EGEAR.PULSESIN or EGEAR.PULSESOUT in order to use EGEAR.RATIO to control electronic gearing.

Related Topics

EGEAR.ON | EGEAR.PULSESIN | EGEAR.PULSEOUT | FB2.SOURCE | FB2.ENCRES | FB2.MODE

6.12.9 EGEAR.TYPE

General Information	
Type	R/W
Description	Sets the allowed direction of motion for electronic gearing.
Units	N/A
Range	0 to 2
Default Value	0
Data Type	Integer
Start Version	TBD

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1076	No	8 bit	No	M_01-05-11-000

Description

EGEAR.TYPE sets allowed direction of motion for electronic gearing:

0 - Both directions

1 - Positive direction

2 - Negative direction

NOTE EGEAR.TYPE cannot be changed on the fly. EGEAR.TYPE is considered only when Gearing is turned on (EGEAR = 1).

When unidirectional gearing is used (EGEAR.TYPE = 1 or 2) then motion in the allowed direction occurs only when the master encoder returns to the point at which it originally reversed direction. Other motion commands, such as MOVE.GOVEL or MOVE.GOREL, may be executed while gearing is active. These moves will be superimposed (added to) on the motion caused by electronic gearing.

Related Topics

EGEAR.ON

6.13 EXTENCODER Parameters

This section describes the EXTENCODER parameters.

6.13.1 EXTENCODER.FREQ

General Information	
Type	R/O
Description	Gets the external encoder (EEO) velocity.
Units	Hz
Range	-2^63 to +2^63 - 1
Default Value	0 Counts
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1078	No	32 bit	No	M_01-05-11-000

Description

Gets the external encoder (EEO) velocity.

6.13.2 EXTENCODER.POSITION

General Information	
Type	R/W
Description	Gets the external encoder (EEO) position.
Units	1/(2^64)
Range	-2^63 to +2^63 -1 OR 0 to EXTENCODER.POSMODULO
Default Value	0 Counts
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1080	Yes	64 bit	Yes	M_01-05-11-000

Description

Gets the external encoder (EEO) position.

6.13.3 EXTENCODER.POSMODULO

General Information	
Type	R/W
Description	Sets/gets the external encoder (EEO) modulo position.
Units	1/(2^64)
Range	0 to +2^64 - 1
Default Value	0 Counts (off)
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1084	Yes	64 bit	No	M_01-05-11-000

Description

Sets/gets the external encoder (EEO) modulo position.

6.14 FAULT Parameters

This section describes the FAULT parameters.

6.14.1 FAULTx.ACTION

General Information	
Type	R/W
Description	Gets/Sets the Fault Action for Fault 130, 131, 132, 134, 139, 451, and 702.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
Start Version	M_01-04-16-000

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?
Modbus	1202	FAULT130.ACTION	No	8 bit	No
	1204	FAULT131.ACTION			
	1206	FAULT132.ACTION			
	1208	FAULT134.ACTION			
	1210	FAULT702.ACTION			
	1230	FAULT451.ACTION			

Description

This Parameter determines the action the drive should take when Fault 130, 131, 132, 134, 139, 451, or 702 occurs.

Parameter Value	Drive Action
0	Disable Ampllifier
1	Ignore (fault will not be reported)

6.15 FB1 Parameters

This section describes the FB1 parameters.

6.15.1 FB1.BISSBITS

General Information	
Type	NV Parameter
Description	Specifies the number of Biss Sensor (Position) Bits for the BiSS Mode C encoder in use.
Units	bits
Range	0 to 64 bits
Default Value	32 bits
Data Type	Integer
See Also	FB1.SELECT , FB1.IDENTIFIED
Start Version	M_01-01-00-100 and M_01-01-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	280	No	8 bit	No	M_01-03-00-000

Description

FB1.BISSBITS specifies the number of BiSS sensor (position) bits for the BiSS Mode C encoder in use. Typically the value is either 26 or 32 for a BiSS Mode C Renishaw encoder. The required value for this parameter is provided by the feedback device manufacturer for the particular device being used.

Related Topics

1 Feedback 1

6.15.2 FB1.ENCRES

General Information	
Type	NV Parameter
Description	Sets the resolution of the motor encoder.
Units	Encoder counts
Range	0 to 2^{32}-1
Default Value	1,024
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3533h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	282	No	32 bit	No	M_01-03-00-000

Description

This parameter sets or gets the resolution of the motor encoder (encoder feedback systems only) in number of counts per revolution for a rotary motor and the number of encoder pitches per motor pole pitch for a linear motor. The number of encoder counts per revolution is obtained by multiplying the motor catalog resolution in units of PPR by four. For example, for a 1024 PPR resolution motor, the number of encoder counts per revolution is 1024*4 = 4096. For this motor FB1.ENCRES must be set to 4096.

For linear motors, the value of FB1.ENCRES is set to the number of encoder pitches per motor pole pitch. For a motor with 32 mm pole pitch, and a 40 µm encoder pitch, the value for FB1.-ENCRES should be set to 32 mm/40 µm = 800.

6.15.3 FB1.HALLSTATE

General Information	
Type	R/O Parameter
Description	Reads the Hall switch values (encoder feedback only).
Units	Binary
Range	0 0 0 to 1 1 1
Default Value	N/A
Data Type	String
See Also	N/A
Start Version	M_01-00-00-000

AKD BASIC Information	
Data Type	Integer

Description

FB1.HALLSTATE reads the Hall switch values (encoder feedback only).

6.15.4 FB1.HALLSTATEU

General Information	
Type	R/O Parameter
Description	Reads the state of Hall switch U.
Units	N/A
Range	0 and 1
Default Value	1
Data Type	Integer
See Also	FB1.HALLSTATE
Start Version	M_01-03-07-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	932	No	8 bit	No

Description

FB1.HALLSTATEU reads the state of Hall switch U.

6.15.5 FB1.HALLSTATEV

General Information	
Type	R/O Parameter
Description	Reads the state of Hall switch V.
Units	N/A
Range	0 and 1
Default Value	1
Data Type	Integer
See Also	FB1.HALLSTATE
Start Version	M_01-03-07-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	934	No	8 bit	No

Description

FB1.HALLSTATEV reads the state of Hall switch V.

6.15.6 FB1.HALLSTATEW

General Information	
Type	R/O Parameter
Description	Reads the state of Hall switch W.
Units	N/A
Range	0 and 1
Default Value	1
Data Type	Integer
See Also	FB1.HALLSTATE
Start Version	M_01-03-07-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	936	No	8 bit	No

Description

FB1.HALLSTATEW reads the state of Hall switch W.

6.15.7 FB1.IDENTIFIED

General Information	
Type	R/O Parameter
Description	Reads the type of feedback device used by the drive/motor.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	FB1.SELECT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	284	No	8 bit	No	M_01-03-00-000

Description

This parameter is set according to FB1.SELECT on drive power up if FB1.SELECT is not –1; otherwise the parameter value is read from the drive memory.

Type	Description
0	Unknown
10	Incremental encoder with A/B Quad, marker pulse and Hall
11	Incremental encoder with A/B Quad, marker pulse and no Hall
20	Sine Encoder , with marker pulse and Hall
21	Sine encoder , with marker pulse & No Halls
30	EnDat 2.1 with Sine Cosine
31	EnDat 2.2
32	BiSS with Sine Cosine
33	HIPERFACE
34	BiSS Mode C Renishaw
40	Resolver
41	SFD

6.15.8 FB1.INITSIGNED

General Information	
Type	NV Parameter
Description	Sets initial feedback value as signed or unsigned.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	FB1.ORIGIN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	286	No	8 bit	Yes	M_01-03-00-000

Description

This parameter sets whether the initial value of the feedback read from the feedback device will be set as a signed or as an unsigned value.

0 = Unsigned

1 = Signed

The drive internal process for the feedback initialization is as follows:

1. Reads the position feedback.
2. Adds the origin to the feedback.
3. Determines modulo from Step 2 by the actual feedback bits.
4. Sets the position feedback sign according to FB1.INITSIGNED.

6.15.9 FB1.MECHPOS

General Information	
Type	R/O Parameter
Description	Reads the mechanical position.
Units	counts
Range	0 to 4,294,967,295 counts
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	288	No	32 bit	No	M_01-03-00-000

Description

FB1.MECHPOS reads the mechanical angle which is equal to the lower 32 bits in the 64-bit position feedback word.

6.15.10 FB1.MEMDUMP

General Information	
Type	R/O Parameter
Description	Dumps the memory ID data values of a feedback with memory
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	N/A
Start Version	M_01-00-00-000

Description

FB1.MEMDUMP dumps the meory ID data values of a feedback with memory.

Related Topics

1 Feedback 1

6.15.11 FB1.MEMVER

General Information	
Type	R/O Parameter
Description	Returns the memory feedback version.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Description

FB1.MEMVER returns the memory feedback version (only applicable for feedbacks with memory).

6.15.12 FB1.ORIGIN

General Information	
Type	NV Parameter
Description	Adds to the initial feedback position.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, µm, custom units, 16-bit counts
Range	Rotary: 0.000 to 5,123,372,000,000.000 counts 0.000 to 7,495.067 rad 0.000 to 429,436.096 deg 0.000 to 5,964.390 custom units 0.000 to 78,176,452.636 16-bit counts Linear: 0.000 to 5,123,372,000,000.000 counts 0.000 to 1,192.878 mm 0.000 to 1,192,877.952 µm 0.000 to 5,964.390 custom units 0.000 to 78,176,452.636 counts16 Bit
Default Value	0 counts
Data Type	Float
See Also	FB1.INITSIGNED
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3656h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	294	Yes	64 bit	No	M_01-03-00-000

Description

FB1.ORIGIN is a value that is added to the feedback device position. Initial value and modulo are determined from the number of bits of the feedback:

Initial position value = (<feedback from device> + FB1.ORIGIN) modulo <number of feedback bits>

The number of feedback bits is set according to the feedback type. For memory feedbacks it is the number of feedback bits; for none memory it is always single turn.

The drive internal process for the feedback initialization is as follows:

1. Reads the position feedback.
2. Adds the origin to the feedback.
3. Determines modulo from Step 2 by the actual feedback bits.
4. Sets the position feedback sign according to FB1.INITSIGNED.

Example

This example uses UNIT.PROTARY set to 2 (degrees)

It also assumes that the drive is connected to a single turn feedback device with memory.

FB1.ORIGIN is set to 22 and saved into NV memory.

Drive boots and reads from feedback device position 340 degrees. According to the description section above, calculation will be:

(340 + 22) modulo 360 = 2 degrees.

Therefore the initial feedback value will be set to 2 degrees.

6.15.13 FB1.P

General Information	
Type	R/O Parameter
Description	Reads position from the primary feedback.
Units	Depends on FB1.UNIT counts or custom units.
Range	N/A
Default Value	N/A
Data Type	S64
See Also	FB1.HALLSTATE
Start Version	M_01-05-08-000

Description

This parameter reads the position of the primary feedback device connected to X10. The position can be read as counts or in customer units. This is the raw position read back from the device. The output format is 32:32, the upper 32 bits represent the multi-turns and the lower 32 bits represent the position of the feedback.

6.15.14 FB1.PDIR

General Information	
Type	NV-Parameter
Description	Sets the counting direction for feedback channel 1.
Units	None
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-11-000

Description

FB1.PDIR will change the sign and with it the direction of feedback channel 1.

6.15.15 FB1.POFFSET

General Information	
Type	NV-Parameter
Description	Sets the offset for primary feedback.
Units	counts, custom units
Range	-5,123,372,000,000,005.000 to 5,123,372,000,000,005.000 counts or -10,485,760.000 to 10,485,760.000 custom units
Default Value	0
Data Type	S64
See Also	N/A
Start Version	M_01-05-11-000

Description

FB1.POFFSET is the value added to the primary feedback position (FB1.P).

Example

If FB1.P is 10000 counts and FB1.POFFSET is set to –10000 counts, then the next read of FB1.P will return ~0 counts.

6.15.16 FB1.POLES

General Information	
Type	R/W Parameter
Description	Reads the number of feedback poles.
Units	N/A
Range	2 to 128
Default Value	2
Data Type	Integer
See Also	MOTOR.POLES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	302	No	16 bit	No	M_01-03-00-000

Description

FB1.POLES sets the number of individual poles in the feedback device. This variable is used for the commutation function, as well as for velocity feedback scaling, and represents the number of individual poles (not pole pairs). The division value of motor poles (MOTOR.POLES) and feedback poles (FB1.POLES) must be an integer when moving drive to enable, otherwise a fault is issued.

6.15.17 FB1.PSCALE

General Information	
Type	R/W Parameter
Description	Sets position scaling value for fieldbus transferred position objects.
Units	N/A
Range	0 to 32
Default Value	20
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	304	No	8 bit	No	M_01-03-00-000

Description

Position values transfered over fieldbus are converted from native 64-bit values to a maximum 32-bit position value. This parameter sets the resolution/revolution of position values back to the controller.

FB1.PSCALE determines the counts per revolution of position values delivered by fieldbus. The default value is 20, which yields 2^20 counts/revolution. This scaling is used for CAN PDOs 6064 (Position Actual Value) and 60F4 (Following Error Actual Value).

Example

The drive always works internally with 64-bit position values. The drive internal 64-bit actual position should contain the following value:

0x0000.0023.1234.ABCD

The lower 32 bits represent the mechanical angle of the feedback. The upper 32 bits represent the number of turns.

FB1.PSCALE = 20

The 32-bit position is: 0x0231234A

FB1.PSCALE = 16

The 32-bit position is: 0x00231234

Related Topics

1 Feedback 1

6.15.18 FB1.PUNIT

General Information	
Type	NV Parameter
Description	Sets the unit for FB1.P.
Units	N/A
Range	0, 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-11-000

Description

FB1.UNIT sets the position unit for FB1.P.

Value	Description
0	Counts (32.32 format)
3	(FB1.PIN/FB1.POUT) per revolution.

Related Topics

FB1.P

6.15.19 FB1.SELECT

General Information	
Type	NV Parameter
Description	Sets user entered type or identified type (–1).
Units	N/A
Range	–1, 10, 20, 30, 31, 32, 40, 41
Default Value	–1
Data Type	Integer
See Also	FB1.IDENTIFIED
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	353Bh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	310	No	8 bit	Yes	M_01-03-00-000

Description

FB1.SELECT sets the feedback type manually (see FB1.IDENTIFIED) or allows the drive to automatically identify the feedback type on power up.

FB1.SELECT Input Values

Input Value	Description
–1	The drive automatically identifies the type of feedback as part of the power up process. Setting this value does not modify FB1.IDENTIFIED, unless it is saved in the NV memory for the next power up. If a feedback with memory is connected to the drive, the value of FB1.IDENTIFIED is set automatically to the feedback identified and all parameters read from the feedback are set according to the values read from the feedback. If no feedback is connected or a feedback with no memory is connected, the value of FB1.IDENTIFIED is set to 0 (no feedback identified) and all values normally read from the feedback are read from NV memory (if stored in NV) otherwise they are set to the default values.
10	Manually sets the type to incremental encoder. This input sets the value of FB1.IDENTIFIED to 10. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).
20	Manually sets the type to sine encoder. This input sets the value of FB1.IDENTIFIED to 20. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).
30	Manually sets the type to Endat 2.1. This input sets the value of FB1.IDENTIFIED to 30. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).
31	Manually sets the type to Endat 2.2. This input sets the value of FB1.IDENTIFIED to 31. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).

Input Value	Description
32	Manually sets the type to BiSS. This input sets the value of FB1.IDENTIFIED to 32. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).
33	Manually sets the type to Hiperface. This input sets the value of FB1.IDENTIFIED to 33. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). Note that all Hiperface feedback types are supported by the AKD. This includes SEL/SEK 37, SEL/SEK 52, SKM/SKS 36, SRS/SRM 50, SRS/SRM 60, SEK 90, SEK160, and SEK 260. The AKD drive will support any new Hiperface device, since any new device will be released with a label type of 0xFF. Devices with this label type have all of the pertinent information to configure these devices (number of single turn bits, number of multi-turn bits, and number of sine/cosine periods) stored in their memory. The AKD is able to read this information, and automatically configure the drive for proper operation. Note that the devices SEK 90, SEK 160, and SEK 260 are label type 0xFF.
40	Manually sets the type to resolver. This input sets the value of FB1.IDENTIFIED to 40. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).
41	Manually sets the type to SFD. This input sets the value of FB1.IDENTIFIED to 41. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified).

FB1.SELECT Feedback Types

Type	Description
0	Unknown
10	Incremental encoder with A/B Quad, marker pulse and Hall
11	Incremental encoder with A/B Quad, marker pulse and no Hall
20	Sine Encoder , with marker pulse and Hall
21	Sine encoder , with marker pulse & No Halls
30	EnDat 2.1 with Sine Cosine
31	EnDat 2.2
32	BiSS with Sine Cosine
33	HIPERFACE
34	BiSS Mode C Renishaw
40	Resolver
41	SFD

6.16 FB2 Parameters

This section describes the FB2 parameters.

6.16.1 FB2.ENCRES

General Information	
Type	NV Parameter
Description	Sets the secondary feedback (FB2) resolution.
Units	counts/rev
Range	0 to 262,140 counts/rev
Default Value	0
Data Type	Integer
See Also	FB2.MODE, FB2.SOURCE
Start Version	M_01-03-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	984	No	32 bit	No

Description

This parameter sets the feedback 2 (FB2) resolution and defines how many counts input into the secondary feedback will be considered a full revolution.

6.16.2 FB2.MODE

General Information	
Type	R/W Parameter
Description	Sets the mode for the second feedback inputs, EEO connector (X9) and high speed opto inputs (pins 9 and 10 on X7).
Units	N/A
Range	0 to 2
Default Value	0
Data Type	Integer
See Also	FB2.ENCRES , PL.FBSOURCE
Start Version	M_01-03-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	986	No	16 bit	No

Description

This parameter sets the feedback 2 input type as follows:

0 = Input A/B Signals

1 = Input Step and Direction Signals

2 = Input, up-down signals

6.16.3 FB2.P

General Information	
Type	R/O Parameter
Description	Reads position from the secondary feedback.
Units	Depends on FB2.UNIT counts or custom units.
Range	N/A
Default Value	N/A
Data Type	U64
See Also	FB1.HALLSTATE
Start Version	M_01-05-08-000

Description

This parameter reads the position back from the secondary feedback device that is connected to X7 or X9, depending on the value of DRV.EMUEMODE. The position can be read as 32-bit counts or in customer units.

6.16.4 FB2P.DIR

General Information	
Type	NV-Parameter
Description	Sets the counting direction for feedback channel 2.
Units	None
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-11-000

Description

FB2.PDIR will change the sign and with it the direction of feedback channel 2.

6.16.5 FB2.POFFSET

General Information	
Type	NV-Parameter
Description	Sets the offset for secondary feedback.
Units	counts, custom units
Range	-5,123,372,000,000,005.000 to 5,123,372,000,000,005.000 counts or -10,485,760.000 to 10,485,760.000 custom units
Default Value	0
Data Type	S64
See Also	N/A
Start Version	M_01-05-11-000

Description

FB2.POFFSET is the value added to the primary feedback position (FB2.P).

Example

If FB2.P is 10000 counts and FB2.POFFSET is set to –10000 counts, then the next read of FB2.P will return ~0 counts.

6.16.6 FB2.PUNIT

General Information	
Type	NV Parameter
Description	Sets the unit for FB2.P.
Units	N/A
Range	0, 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-11-000

Description

FB2.UNIT sets the position unit for FB2.P.

Value	Description
0	Counts (32 bit format)
3	(FB2.PIN/FB2.POUT) per revolution.

Related Topics

FB2.P

6.16.7 FB2.SOURCE

General Information	
Type	R/W Parameter
Description	Sets the source for the second feedback input. Choices are the EEO connectors (X9) which are RS485 inputs, or the X7 connector's high speed opto inputs (pins 9 and 10).
Units	N/A
Range	0 to 2
Default Value	0
Data Type	Integer
See Also	FB2.ENCRES, FB2.MODE, PL.FBSOURCE
Start Version	M_01-03-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	988	No	16 bit	No

Description

This parameter sets the secondary feedback source to be either the EEO connector (X9) or the high speed opto inputs on the I/O Connector (X7) as follows:

0 = None

1 = Feedback Source X9 (EEO connector)

2 = Feedback Source X7 (High Speed Opto Inputs on the I/O Connector)

Related Topics

1 Feedback 2

6.17 FB3 Parameters

This section describes the FB3 parameters.

6.17.1 FB3.MODE

General Information	
Type	NV Parameter
Description	Selects the type of feedback connected to X9.
Units	N/A
Range	0
Default Value	0
Data Type	Integer
See Also	NA
Start Version	M_01-04-15-000

Fieldbus	Index/Subindex
Modbus	1044

Description

This parameter selects the type of feedback connected to X9. The position is reported as the tertiary feedback position, by FB3.P.

Value	Feedback
0	Endat 2.2 Feedback Device

This parameter is only supported on drives with model numbers similar to AKD-x-xxxxx-NBxx-xxxx.

6.17.2 FB3.POFFSET

General Information	
Type	NV-Parameter
Description	Sets the offset for tertiary feedback.
Units	counts, custom units
Range	-5,123,372,000,000,005.000 to 5,123,372,000,000,005.000 counts or -10,485,760.000 to 10,485,760.000 custom units
Default Value	0
Data Type	S64
See Also	N/A
Start Version	M_01-05-11-000

Description

FB3.POFFSET is the value added to the primary feedback position (FB3.P).

Example

If FB3.P is 10000 counts and FB3.POFFSET is set to –10000 counts, then the next read of FB3.P will return ~0 counts.

6.17.3 FB3.PUNIT

General Information	
Type	NV Parameter
Description	Sets the unit for FB3.P.
Units	N/A
Range	0, 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-05-11-000

Description

FB3.UNIT sets the position unit for FB3.P.

Value	Description
0	Counts (32.32 format)
3	(FB3.PIN/FB3.POUT) per revolution.

Related Topics

FB3.P

6.18 HWLS Parameters

This section describes the HWLS parameters.

6.18.1 HWLS.NEGSTATE

General Information	
Type	R/O Parameter
Description	Reads the status of the negative hardware limit switch.
Units	0 to 1
Range	N/A
Default Value	Integer
Data Type	HWLS.POSSTATE
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	422	No	8 bit	No	M_01-03-00-000

Description

HWLS.NEGSTATE reads the status of the negative HW limit switch as follows:

0 = Low

1 = High

6.18.2 HWLS.POSSTATE

General Information	
Type	R/O Parameter
Description	Reads the status of the positive hardware limit switch.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	HWLS.NEGSTATE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	424	No	8 bit	No	M_01-03-00-000

Description

HWLS.POSSTATE reads the status of the positive hardware limit switch as follows:

0 = Low

1 = High

6.19 IL Parameters

This section describes the IL parameters.

6.19.1 IL.BUSFF

General Information	
Type	R/O Parameter
Description	Displays the current feedforward value injected by the field-bus.
Units	Arms
Range	N/A
Default Value	N/A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	426	No	32 bit	Yes	M_01-03-00-000

Description

This parameter displays the current feedforward value injected by the fieldbus.

6.19.2 IL.CMD

General Information	
Type	R/O Parameter
Description	Reads the value of the q-component current command.
Units	Arms
Range	± Drive peak current (DRV.IPEAK)
Default Value	N/A
Data Type	Float
See Also	DRV.IPEAK
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	428	No	32 bit	Yes	M_01-03-00-000

Description

IL.CMD displays the q-component current command value of the current loop after any limitation (such as a parameter setting or I^2t calculation).

IL.CMD is limited also by motor peak current, IL.LIMITN and IL.LIMITP .

6.19.3 IL.CMDU

General Information	
Type	R/W Parameter
Description	Sets the user current command.
Units	Arms
Range	Minimum range value = maximum of IL.LIMITN and -MOTOR.IPEAK Maximum range value = minimum of IL.LIMITP and MOTOR.IPEAK
Default Value	0 Arms
Data Type	Float
See Also	DRV.IPEAK , DRV.OPMODE ,DRV.CMDSOURCE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	430	No	32 bit	Yes	M_01-03-00-000

Description

This parameter sets the user current command value.

The current command value, which is provided to the current loop (IL.CMD), can be limited further using a parameter setting or I^2t calculation. IL.CMDU is limited also by motor peak current, IL.LIMITN and IL.LIMITP .

6.19.4 IL.DFOLDT

General Information	
Type	R/W Parameter
Description	Reads the drive foldback time constant of the exponential current drop (fold-back).
Units	s
Range	1 to 65.535 s
Default Value	Usually 2.500 s, but may change according to the drive type.
Data Type	Float
Start Ver-sion	M_01-00-00-000

Description

IL.DFOLDT is the time constant of the exponential drop (foldback) of the current towards the drive continuous current.

6.19.5 IL.DIFOLD

General Information	
Type	R/O Parameter
Description	Reads the drive foldback current limit.
Units	Arms
Range	0 to 2,147,483.647 Arms
Default Value	N/A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3559h/0	M_01-00-00-000

Description

IL.DIFOLD is the output of the drive foldback algorithm. It is an artificial current, which can be higher or lower than the drive peak current (DRV.IPEAK). When IL.DIFOLD is lower than the existing current limit (such as IL.LIMITP), it becomes the active current limit.

IL.DIFOLD decreases when the actual current is higher than drive continuous current and increases (up to a certain level) when the actual current is lower than drive continuous current.

6.19.6 IL.FB

General Information	
Type	R/O Parameter
Description	Reads the actual value of the d-component current.
Units	Arms
Range	± Drive peak current (DRV.IPEAK)
Default Value	N/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3558h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	432	No	32 bit	Yes	M_01-03-00-000

Description

This parameter reads the measured, de-rotated actual current value of the motor.

Note: Internally the resolution of the current scale is 20130 increments. For an AKD with a peak current of 9 amps, the current resolution applied is 9/20130 = .447 mA. For a 48 amp peak current drive, the resolution is 48/20130 = 2.38 mA. The current scaling is hard coded and cannot be changed by decreasing the peak current settings in the drive.

6.19.7 IL.FF

General Information	
Type	R/O Parameter
Description	Displays the current loop overall feedforward value
Units	Arms
Range	N/A
Default Value	N/A
Data Type	Float
See Also	IL.KBUSFF, IL.KVFF, IL.OFFSET, IL.FRICTION, IL.KACCFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	434	No	32 bit	No	M_01-03-00-000

Description

This parameter displays the current loop overall feedforward value.

Related Topics

1 Current Loop

6.19.8 IL.FOLDFTHRESH

General Information	
Type	R/O Parameter
Description	Reads the foldback fault level.
Units	Arms
Range	0 to 500 Arms
Default Value	Drive peak current (DRV.IPEAK)
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3420h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	436	No	32 bit	No	M_01-03-00-000

Description

IL.FOLDFTHRESH is the fault level of the current foldback algorithm. If IL.IFOLD drops below the value for IL.FOLDFTHRESH, then a fault is generated and the drive is disabled.

To avoid reaching the current foldback fault level, set IL.FOLDFTHRESHU well below the continuous current value for both the drive and the motor or set the IL.FOLDFTHRESHU value to zero.

6.19.9 IL.FOLDWTHRESH

General Information	
Type	NV Parameter
Description	Sets the foldback warning level.
Units	Arms
Range	0 to 500 Arms
Default Value	0 A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	355Ah/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	440	No	32 bit	Yes	M_01-03-00-000

Description

IL.FOLDWTHRESH is the warning level of the current foldback algorithm. When IL.IFOLD drops **below** IL.FOLDWTHRESH a warning is generated.

To ensure that the current foldback warning level is never reached, IL.FOLDWTHRESH should be set well below the continuous current value for both the drive and the motor. You can also set the IL.FOLDFTHRESH value to zero.

6.19.10 IL.IFOLD

General Information	
Type	R/O Parameter
Description	Reads the overall foldback current limit.
Units	A
Range	0 to 2,147,483.647 A
Default Value	N/A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3425h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	444	No	32 bit	No	M_01-03-00-000

Description

Two current foldback algorithms run in parallel in the drive: the drive foldback algorithm and the motor foldback algorithm. Each algorithm uses different sets of parameters.

Each algorithm has its own foldback current limit, IL.DIFOLD and IL.MIFOLD. The overall foldback current limit is the minimum of the two at any given moment.

```
IL.IFOLD = min (IL.DIFOLD, IL.MIFOLD).
```

IL.DIFOLD is an artificial current, which can be higher or lower than the drive or motor peak current. When IL.IFOLD becomes lower than the existing current limit (such as IL.LIMITP), it becomes the active current limit.

6.19.11 IL.IUFB

General Information	
Type	R/O Parameter
Description	Reads the sigma-delta measured current in the u-winding of the motor.
Units	A
Range	± Drive peak current (DRV.IPEAK)
Default Value	N/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	446	No	32 bit	Yes	M_01-03-00-000

Description

This parameter displays the measured current in the u-winding of the motor.

6.19.12 IL.IVFB

General Information	
Type	R/O Parameter
Description	Sets the sigma-delta measured current in the u-winding of the motor.
Units	A
Range	± Drive peak current (DRV.IPEAK)
Default Value	0 A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	448	No	32 bit	Yes	M_01-03-00-000

Description

IL.IVFB is an offset value that is added to the measured current in the u-winding of the motor. This value is used for compensating for an error in the current measurement. The drive measures 256 times the current in the u-winding when powering-up the drive. Afterwards, the drive calculates the average value of the measured current and uses this value for the offset value.

6.19.13 IL.KP

General Information	
Type	NV Parameter
Description	Sets the proportional gain of the q-component of the PI regulator.
Units	V/A
Range	0 to 2,000 V/A
Default Value	Read from the motor or, if no memory, 50.009 V/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3598h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	454	No	16 bit	No	M_01-03-00-000

Description

IL.KP is used to modify the proportional gain of the PI-loop that controls the q-component of the current.

6.19.14 IL.KPDRATIO

General Information	
Type	NV Parameter
Description	Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP
Units	N/A
Range	0 to 100
Default Value	1
Data Type	Float
See Also	IL.KP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3596h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	456	No	32 bit	No	M_01-03-00-000

Description

This parameter modifies the proportional gain of the PI-loop, which controls the d-component of the current.

6.19.15 IL.LIMITN

General Information	
Type	NV Parameter
Description	Sets the negative user (application-specific) current limit.
Units	A
Range	Negative drive peak current (DRV.IPEAK) to 0 A
Default Value	Negative drive peak current (DRV.IPEAK)
Data Type	Float
See Also	IL.LIMITP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	356Fh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	460	No	32 bit	Yes	M_01-03-00-000

Description

This parameter sets the negative user limit clamp value of the torqueproducing q-component current command (IL.CMD). The current command is additionally limited by the motor peak current setting (MOTOR.IPEAK) and by the present value of the foldback I^2t peak motor current protection.

6.19.16 IL.LIMITP

General Information	
Type	NV Parameter
Description	Sets the positive user (application-specific) current limit.
Units	A
Range	0 A to drive peak current (DRV.IPEAK)
Default Value	Drive peak current (DRV.IPEAK)
Data Type	Float
See Also	IL.LIMITN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	356Eh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	462	No	32 bit	Yes	M_01-03-00-000

Description

This parameter sets the positive user limit clamp value of the torque-producing q-component current command (IL.CMD). The current command is additionally limited by the motor peak current setting (MOTOR.IPEAK) and by the present value of the foldback I²t peak motor current protection.

6.19.17 IL.MFOLDD

General Information	
Type	R/O Parameter
Description	Sets the motor foldback maximum time at motor peak current.
Units	s
Range	0.1 to 2400 s
Default Value	10 s
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	464	No	32 bit	No	M_01-03-00-000

Description

IL.MFOLDD sets the maximum time allowed for the motor to remain at peak current before starting to fold towards the motor continuous current. When at motor peak current, IL.MFOLDD is the amount of time before the foldback algorithm starts to reduce the current.

6.19.18 IL.MFOLDR

General Information	
Type	R/O Parameter
Description	Sets the motor foldback recovery time.
Units	s
Range	0.1 to 65,535 s
Default Value	Calculated from other foldback parameters.
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	466	No	32 bit	No	M_01-03-00-000

Description

IL.MFOLDR sets the recovery time for the motor foldback algorithm. If 0 current is applied for at least the recovery time duration, it is possible to apply motor peak current for the duration of IL.MFOLDD time.

The IL.MFOLDR value is automatically calculated from other foldback parameters.

6.19.19 IL.MFOLDT

General Information	
Type	R/O Parameter
Description	Sets the motor foldback time constant of the exponential current drop (fold-back).
Units	s
Range	0.1 to 2,400 s
Default Value	10 s
Data Type	Float
Start Ver-sion	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	468	No	32 bit	No	M_01-03-00-000

Description

IL.MFOLDT sets the time constant of the exponential drop (foldback) of the current towards motor continuous current.

6.19.20 IL.MIFOLD

General Information	
Type	R/O Parameter
Description	Sets the motor foldback current limit.
Units	A
Range	0 to 2147483.647 A
Default Value	N/A
Data Type	Float
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	35A4h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	470	No	32 bit	No	M_01-03-00-000

Description

IL.MIFOLD sets the output of the motor foldback algorithm. It is an artificial current, which can be higher or lower than the motor peak current. When IL.MIFOLD becomes lower than the existing current limit (IL.LIMITP) it becomes the active current limit.

IL.MIFOLD decreases when the actual current is higher than motor continuous current and increases (up to a certain level) when the actual current is lower than the motor continuous current.

Related Topics

1 Current Loop

6.19.21 IL.VCMD

General Information	
Type	R/O Parameter
Description	Sets the output of the q-component PI regulator.
Units	Vrms
Range	0 Vrms to bus voltage
Default Value	N/A
Data Type	Integer
See Also	IL.VDCMD
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	474	No	16 bit	Yes	M_01-03-00-000

Description

Sets the output of the current loop that controls the q-component of the current.

6.19.22 IL.VUFB

General Information	
Type	R/O Parameter
Description	Reads the measured voltage on the u-winding of the motor.
Units	V
Range	–1200*VBusScale to +1200*VBusScale
Default Value	N/A
Data Type	Integer
See Also	IL.VVFB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	476	No	16 bit	Yes	M_01-03-00-000

Description

Reads the measured voltage on the u-winding of the motor.

6.19.23 IL.VVFB

General Information	
Type	R/O Parameter
Description	Reads the measured voltage on the v-winding of the motor.
Units	V
Range	–1200*VBusScale to +1200*VBusScale
Default Value	N/A
Data Type	Integer
See Also	IL.VUFB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	478	No	16 bit	Yes	M_01-03-00-000

Description

The range for this parameter depends on whether the drive model is an MV/240 Vac or an HV/480 Vac.

The VBusScale parameter sets the drive model:

MV/240 Vac: VBusScale = 1

HV/480 Vac: VBusScale = 2

VBusScale is used for multiple parameter ranges that are model dependent, such as IL.KP.

6.20 INTR Parameters

This section describes the INTR parameters.

6.20.1 Interrupt {Source}

General Information	
Type	R/W
Description	Interrupt sources enable or disable Interrupt...End Interrupt statements.
Units	none
Range	0 (disabled) or 1 (enabled)
Default Value	0 (disabled)
Data Type	Integer

Description

Interrupt sources enable or disable Interrupt ... End Interrupt statements. If you enable a given interrupt then there must be an Interrupt ... End Interrupt definition for that interrupt source in your program.

The table below lists the source names and a brief description for all the interrupt sources available on the AKD BASIC.

Interrupt Source	Interrupt Occurs
INTR.DIN1HI	when DIN1.STATE goes from 0 to 1
INTR.DIN1LO	when DIN1.STATE goes from 1 to 0
INTR.DIN2HI	when DIN2.STATE goes from 0 to 1
INTR.DIN2LO	when DIN2.STATE goes from 1 to 0
INTR.DIN3HI	when DIN3.STATE goes from 0 to 1
INTR.DIN3LO	when DIN3.STATE goes from 1 to 0
INTR.DIN4HI	when DIN4.STATE goes from 0 to 1
INTR.DIN4LO	when DIN4.STATE goes from 1 to 0
INTR.DIN5HI	when DIN5.STATE goes from 0 to 1
INTR.DIN5LO	when DIN5.STATE goes from 1 to 0
INTR.DIN6HI	when DIN6.STATE goes from 0 to 1
INTR.DIN6LO	when DIN6.STATE goes from 1 to 0
INTR.DIN7HI	when DIN7.STATE goes from 0 to 1
INTR.DIN7LO	when DIN7.STATE goes from 1 to 0
INTR.DISABLE	when the drive gets disabled
INTR.DRV.FAULTS	when the drive faults
INTR.DRV.HWENABLE	when DRV.HWENABLE goes from 0 to 1
INTR.DRV.WARNINGS	when the drive produces a warning
INTR.HWLS.NEGSTATE	when HWLS.NEGSTATE goes from 0 to 1
INTR.HWLS.POSSTATE	when HWLS.POSSTATE goes from 0 to 1
INTR.MODBUS	when a Modbus User Parameter changes.
INTR.PL.ERR	when PL.ERR = PL.ERRFTHRESH
INTR.PLS.P1 to INTR.PLS.P8	when PLS.P1 to PLS.P8 is enabled and goes high, respectively.
INTR.SWLS.LIMIT0	when PL.FB > SWLS.LIMIT0 (if SWLS.LIMIT0 is the upper limit)

Interrupt Source	Interrupt Occurs
INTR.SWLS.LIMIT1	when PL.FB < SWLS.LIMIT1 (if SWLS.LIMIT1 is the lower limit)
INTR.TIMER	after a number of milliseconds specified by VM.INTRTIMER

Example

```
Main
        DRV.TIME = 0
        INTR.DIN1LO = 1
        while 1
                pause(0.5)
                DOUT1.STATE=0 : Pause(0.005) : DOUT1.STATE=1
        wend
end main
'-------------- Interrupt Routines ------------
-------
Interrupt                          DIN1LO
        print                         "I'm awake"
        If                         DRV.TIME > 10 then
                print                          "OK. That's it."
        else
                INTR.DIN1LO = 1
        end if
End Interrupt
```

Related Topics

Interrupt...End Interrupt

6.20.2 Interrupt...End Interrupt

General Information	
Type	Statement
Description	The interrupt feature permits execution of a user-defined subroutine upon receipt of a hardware interrupt signal or a pre-defined interrupt event.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Description

The Interrupt statment marks the beginning of an Interrupt Service Routine. The Interrupt Service Routine is defined by a program structure resembling a subroutine. The interrupt feature permits execution of a user-defined subroutine upon receipt of a hardware interrupt signal or a pre-defined interrupt event.

Interrupts are triggered by pre-defined events or external hardware sources. The interrupt-source-name and interrupt enable flag are unique for each interrupt source.

Receiving an interrupt will suspend program execution and the interrupt service routine will be executed. Then program execution will resume at the point that it was interrupted.

Interrupts are enabled (or disabled) by setting (or clearing) the associated interrupt enable flag. Interrupts are disabled until explicitly enabled. After an interrupt is triggered it is automatically disabled until it is enabled again in your program.

Example

```
Main
        DRV.TIME = 0
        INTR.DIN1LO = 1
        while 1
                pause(0.5)
                DOUT1.STATE=0 : Pause(0.005) : DOUT1.STATE=1
        wend
end main
'-------------- Interrupt Routines ------------
-------
Interrupt               DIN1LO
        print               "I'm awake"
        If              DRV.TIME > 10 then
                print               "OK. That's it."
        else
                INTR.DIN1LO = 1
        end if
End Interrupt
```

Related Topics

Interrupt {Source} | Sub...End Sub | Restart

6.21 LOAD Parameters

This section describes the LOAD parameters.

6.21.1 LOAD.INERTIA

General Information	
Type	NV Parameter
Description	Sets the load inertia.
Units	kgcm² for rotary motors kg for linear motors
Range	1 to 1,000,000 kgcm² or kg
Default Value	0 kgcm² or kg
Data Type	Float
See Also	N/A
Start Version	M_01-03-06-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1214	No	32 bit	No

Description

LOAD.INERTIA sets the load inertia.

6.22 MODBUS Paramters

This section describes the MODBUS parameters.

6.22.1 MODBUS.READFLOAT

General Information	
Type	R/W
Description	This function reads a floating-point value from the specified ModBus slave and returns the value read.
Units	TBD
Range	TBD
Default Value	TBD
Data Type	Float
Start Version	TBD

Description

This function reads a floating-point value from the specified ModBus slave and returns the value read. If any error occurs, this function returns zero and sets MODBUS.ERR to indicate the source of the error. The register address passed to this function is the first register address of the 32 bit floating point value.

This is a ModBus master function. Set RuntimeProtocol to 3 before using this function or a runtime error is received. ModBus master statements and functions cannot be nested. If you get an interrupt while waiting for a response to a ModBus master statement or function, you cannot initiate another ModBus master statement or function in the interrupt handler. If you do, you get runtime error 36.

There is not complete standardization on the format of floating-point numbers among all ModBus devices. You may need to set MODBUS.FLOATWORDORDER to 0 (its default value is 1) in order to properly receive floating point numbers from a ModBus slave. Refer to Using an AKD BASIC as a ModBus Master.

Related Topics

MODBUS.READ16 | MODBUS.READ32

6.22.2 MODBUS.WRITEFLOAT

General Information	
Type	R/W
Description	This statement writes a floating-point value to the specified ModBus slave.
Units	TBD
Range	TBD
Default Value	TBD
Data Type	Float
Start Version	TBD

Description

This statement writes a floating-point value to the specified ModBus slave. If any error occurs, this function sets MODBUS.ERR to indicate the source of the error. The register address passed to this function is the first register address of the 32 bit floating point value.

This is a ModBus master statement. Set RuntimeProtocol to 3 before using it or a runtime error is received.

ModBus master statements and functions cannot be nested. If you get an interrupt while waiting for a response to a ModBus master statement or function, you cannot initiate another ModBus master statement or function in the interrupt handler. If you do, you get runtime error 36.

There is not complete standardization on the format of floating-point numbers among all ModBus devices. Set Modbus.FloatWordOrder to 0 to properly write floating point numbers to a ModBus slave. Refer to Using an AKD BASIC as a ModBus Master.

Related Topics

MODBUS.WRITE16 | MODBUS.WRITE32

6.23 MOTOR Parameters

This section describes the MOTOR parameters.

6.23.1 MOTOR.BRAKE

General Information	
Type	NV Parameter
Description	Sets the presence or absence of a motor brake.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Boolean
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3587h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	482	No	8 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

The MOTOR.BRAKE parameter notifies the firmware whether a brake exists or not. It does not apply or release the brake. If a brake is found to be present, the firmware considers hardware indications regarding the brake circuits (such as open circuit or short circuit). If a brake does not exist, then the firmware ignores the hardware indications since they are irrelevant.

Value	Status
0	Motor brake does not exist.
1	Motor brake exists and brake hardware circuitry checks are enabled.

Enabling the MOTOR.BRAKE (value set to 1) when no motor brake exists creates a fault.

The motor brake is polled every 16 ms.

6.23.2 MOTOR.BRAKEIMM

General Information	
Type	NV Parameter
Description	Brake Immediately: in the case of a drive disable, apply the brake in all situations.
Units	N/A
Range	0 to 1
Default Value	0 (Inactive)
Data Type	Boolean
See Also	N/A
Start Version	M_01-05-11-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1232	No	8 bit	No

Description

With the standard configuration, when the drive disables, the brake will not apply until velocity falls below CS.VTHRESH for CS.TO milliseconds. However, in some machines (such as a vertical axis) the brake should be applied immediately whenever the drive disables.

To ensure that the brake is applied immediately after any disable (due to fault, disable command, etc), set MOTOR.BRAKEIMM = 1.

6.23.3 MOTOR.BRAKERLS

General Information	
Type	Command
Description	Allows a user to release the motor brake.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3450h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	484	No	8 bit	No	M_01-03-00-000

Description

This command allows a user to release the motor brake.

0 = Drive controls the brake.

1 = Brake is released.

Note: A digital input mode is also used for the same purpose. The two mechanisms are independent.

Related Topics

1 Motor

6.23.4 MOTOR.ICONT

General Information	
Type	NV Parameter
Description	Sets the motor continuous current.
Units	A
Range	0.1 to 500 A
Default Value	1.0 A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	358Eh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	488	No	32 bit	No	M_01-03-00-000

AKD BASIC Information	
Type	R/W

Description

This parameter is used to configure the motor continuous current.

6.23.5 MOTOR.INERTIA

General Information	
Type	NV Parameter
Description	Sets the motor inertia.
Units	kgcm² for rotary motors kg for linear motors
Range	1 to 200,000 kgcm² or kg
Default Value	100 kgcm² or kg
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	35ABh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	492	No	32 bit	No	M_01-03-00-000

Description

This parameter sets the motor inertia.

6.23.6 MOTOR.IPEAK

General Information	
Type	NV Parameter
Description	Sets the motor peak current.
Units	mA
Range	0.200 to 1,000 A
Default Value	2.000 A
Data Type	Float
See Also	IL.LIMITP , IL.LIMITN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	358Fh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	494	No	32 bit	No	M_01-03-00-000

Description

This parameter configures the drive for the motor's peak, instantaneous-rated current. MOTOR.IPEAK is used to limit clamp the magnitude of the torque producing q-component current command (IL.CMD).

6.23.7 MOTOR.KE

General Information	
Type	NV Parameter
Description	Sets the motor back EMF constant.
Units	Vpeak/krpm for Rotary Motors Vpeak/m/s for Linear Motors
Range	0.0 to 100,000
Default Value	0
Data Type	Float
See Also	N/A
Start Version	M_01-03-06-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?
Modbus	1216	No	32 bit	No

Description

MOTOR.KE defines the back EMF constant for the motor. The back EMF constant defines how much voltage is generated at the motors coils. The relationship between MOTOR.KE and speed is described by the following equation:

```
Coil Voltage = MOTOR.KE * VL.FB
```

Where:

VL.FB is in units of krpm for rotary motors and in units of m/s for linear motors

6.23.8 MOTOR.KT

General Information	
Type	NV Parameter
Description	Sets the torque constant of the motor.
Units	Nm/A
Range	0.001 Nm/A to 1,000,000.000 Nm/A for rotary motors. 0.001 Nm/A to 1,000,000.000 N/A for linear motors.
Default Value	0.1 Nm/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3593h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	496	No	32 bit	No	M_01-03-00-000

Description

This parameter is the torque constant of the motor in Nm/A. The value can be online checked according to the following equation:

$$Kt = 60 * \sqrt{3} * Ui / (2 * \pi * n)$$

Where:

Ui = induced voltage of the motor
n = actual rotor velocity

6.23.9 MOTOR.LQLL

General Information	
Type	NV Parameter
Description	Sets the line-to-line motor Lq.
Units	mH
Range	1 to 2^{32} H
Default Value	17.000 H
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3455h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	498	No	32 bit	No	M_01-03-00-000

Description

This parameter is used to configure the motor line-to-line inductance.

6.23.10 MOTOR.NAME

General Information	
Type	NV Parameter
Description	Sets the motor name.
Units	N/A
Range	11 chars
Default Value	N/A
Data Type	String
See Also	N/A
Start Version	M_01-00-00-000

Description

This parameter is used to set the motor name.

6.23.11 MOTOR.PHASE

General Information	
Type	NV Parameter
Description	Sets the motor phase.
Units	Electrical degrees
Range	0 to 360°
Default Value	0°
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	359Ch/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	500	No	16 bit	No	M_01-03-00-000

Description

This parameter sets the motor phase.

6.23.12 MOTOR.PITCH

General Information	
Type	NV Parameter
Description	Sets the motor pitch.
Units	µm
Range	1,000 to 1,000,000 µm
Default Value	1.000 µm
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	502	No	32 bit	No	M_01-03-00-000

Description

This parameter defines the pole-to-pair pitch for the linear motor in micrometers.

6.23.13 MOTOR.POLES

General Information	
Type	NV Parameter
Description	Sets the number of motor poles.
Units	N/A
Range	0 to 128
Default Value	6
Data Type	Integer
See Also	FB1.POLES
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	359Dh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	504	No	16 bit	No	M_01-03-00-000

Description

MOTOR.POLES sets the number of motor poles. This command is used for commutation control and represents the number of individual magnetic poles of the motor (not pole pairs). The division value of motor poles (MOTOR.POLES) and feedback poles (FB1.POLES) must be an integer when setting drive to enable, otherwise a fault is issued.

6.23.14 MOTOR.R

General Information	
Type	NV Parameter
Description	Sets the stator winding resistance phase-phase in ohms.
Units	Ω
Range	0.001 to 650 Ω
Default Value	10 Ω
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3456h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	506	No	32 bit	No	M_01-03-00-000

Description

MOTOR.R sets the stator winding resistance phase-to-phase in ohms.

6.23.15 MOTOR.TBRAKEAPP

General Information	
Type	NV Parameter
Description	The delay time used for applying the motor brake.
Units	ms
Range	0 to 1,000 ms
Default Value	75 ms
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	366Eh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	510	No	16 bit	No	M_01-03-00-000

Description

This parameter is used to configure the mechanical delay when applying the motor brake. MOTOR.TBRAKEAPP is a time delay that is applied when a brake exists and the drive is disabled at the end of a controlled stop. This delay lasts from the time that the brake is commanded to apply until the time that the drive is disabled.

This feature allows you to disable the drive and apply the brake on a vertical application without the load falling. Without this time delay, if you immediately disable the drive, then the load falls during the time needed for the brake to mechanically apply.

6.23.16 MOTOR.TBRAKERLS

General Information	
Type	NV Parameter
Description	The delay time used for releasing the motor brake.
Units	ms
Range	0 to 1,000 ms
Default Value	75 ms
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	366Fh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	512	No	16 bit	No	M_01-03-00-000

Description

This parameter is used to configure the mechanical delay when releasing the motor brake. MOTOR.TBRAKERLS is a time delay that is applied when a brake exists and the drive is enabled. When the drive is enabled, the brake is commanded to release and, during the MOTOR.TBRAKERLS period of time, the drive does not accept a motion command. This delay allows the brake to fully release before the drive begins a new motion.

6.23.17 MOTOR.TEMP

General Information	
Type	R/O Parameter
Description	Reads the motor temperature represented as the resistance of the motor PTC.
Units	Ω
Range	0 to 2^{32} Ω
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3612h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	514	No	32 bit	No	M_01-03-00-000

Description

This parameter is used to get the motor temperature which is represented as the resistance of the motor PTC.

6.23.18 MOTOR.TEMPFAULT

General Information	
Type	NV Parameter
Description	Sets the motor temperature fault level.
Units	Ω
Range	0 to 2,000,000,000 Ω
Default Value	0 Ω = switched off
Data Type	Integer
See Also	MOTOR.TEMP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3586h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	516	No	32 bit	No	M_01-03-00-000

Description

This parameter is used to configure the motor temperature fault level as a resistance threshold of the motor PTC.

A zero value prevents any warning from being issued.

6.23.19 MOTOR.TEMPWARN

General Information	
Type	NV Parameter
Description	Sets the motor temperature warning level.
Units	Ω
Range	0 to 2,000,000,000 Ω
Default Value	0 Ω = switched off
Data Type	Integer
See Also	MOTOR.TEMP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3453h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	518	No	32 bit	No	M_01-03-00-000

Description

This parameter is used to configure the motor temperature warning level as a resistance threshold of the motor PTC.

A zero value prevents any warning from being created.

6.23.20 MOTOR.TYPE

General Information	
Type	NV Parameter
Description	Sets the motor type.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	520	No	8 bit	No	M_01-03-00-000

Description

MOTOR.TYPE sets the drive control algorithms to different motor types as follows:

0 = Rotary motor

1 = Linear motor

6.23.21 MOTOR.VOLTMAX

General Information	
Type	NV Parameter
Description	Sets the motor maximum voltage.
Units	Vrms
Range	110 to 900 Vrms
Default Value	230 Vrms
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3452h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	524	No	16 bit	No	M_01-03-00-000

Description

This parameter sets the maximum permissible motor voltage. For instance, if a motor that is rated for a 400 V supply is connected to the drive, then the MOTOR.VOLTMAX setting is 400. This value also sets regen resistor and over voltage thresholds in the drive to acceptable values for the motor so that the motor windings are not damaged.

6.24 MOVE Parameters

This section describes the MOVE parameters.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

6.24.1 MOVE.ABORT

General Information	
Type	Command
Description	MOVE.ABORT stops motor motion and allows continued program execution.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1144	No	Command	No	M_01-05-11-000

Description

MOVE.ABORT stops motor motion and allows continued program execution. Deceleration is determined by CS.DEC.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

This program segment commands the motor at constant velocity until input 1 goes to a logic 0. Then, the motor is commanded to stop.

```
MOVE.ACC = 5000
'Set acceleration rate equal to 5,000 rpm/sec
CS.DEC = 10000
'Set controlled stop deceleration rate to
10,000 rpm/sec
MOVE.RUNSPEED = 120
'Set Run speed equal to 120 rpm
MOVE.GOVEL
When                DIN2.STATE = 0, MOVE.ABORT
'Motor decelerates to a stop at CS.DEC (10,000
rpm/sec)
Print                "Move Aborted!"
```

Related Topics

"Stop" (=> S. 117) | Statement Table | CS.DEC

6.24.2 MOVE.ACC

General Information	
Type	R/W
Description	Sets the maximum commanded acceleration rate when the speed is increased.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s², (custom units)/s² , rad/s² Linear: counts/s² , mm/s² , µm/s² , (custom units)/s²
Range	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s² 0.155 to 4,166,666.752 (custom units)/s² 0.012 to 5,235,987.968 rad/s² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s² 30.995*MOTOR.PITCH to 2,147,483.647*MOTOR.PITCH µm/s² 0.155 to 2,147,483.647 (custom units)/s²
Default Value	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s² 833.333 (custom units)/s² 1,047.2 rad/s² Linear: 715,840,000.000 counts/s² 166.714*MOTOR.PITCH mm/s² 166,714.191*MOTOR.PITCH µm/s² 833.571 (custom units)/s²
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1088	Yes	64 bit	No	M_01-05-11-000

Description

Sets the maximum commanded acceleration rate when the speed is increased. Set MOVE.ACC prior to initiating the move. You can update MOVE.ACC during a move by executing an MOVE.GOUPDATE statement.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
'This example sets MOVE.ACC to 10,000 rpm/sec
and does a
'relative move of 10 motor revolutions.
MOVE.RUNSPEED = 1000
MOVE.ACC = 10000
MOVE.DEC = 10000
MOVE.RELATIVEDIST = 655360
MOVE.GOREL
```

Related Topics

MOVE.DEC

6.24.3 MOVE.DEC

General Information	
Type	R/W
Description	Sets the maximum commanded deceleration rate when the speed is decreased.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s², (custom units)/s² , rad/s² Linear: counts/s² , mm/s² , µm/s² , (custom units)/s²
Range	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s² 0.155 to 4,166,666.752 (custom units)/s² 0.012 to 5,235,987.968 rad/s² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s² 30.995*MOTOR.PITCH to 2,147,483.647*MOTOR.PITCH µm/s² 0.155 to 2,147,483.647 (custom units)/s²
Default Value	Note: The range and default values of (custom units)/s² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s² 833.333 (custom units)/s² 1,047.2 rad/s² Linear: 715,840,000.000 counts/s² 166.714*MOTOR.PITCH mm/s² 166,714.191*MOTOR.PITCH µm/s² 833.571 (custom units)/s²
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1092	Yes	64 bit	No	M_01-05-11-000

Description

Sets the maximum commanded deceleration rate when the speed is decreased. Set MOVE.DEC prior to initiating the move. You can update MOVE.DECduring a move by executing an MOVE.GOUPDATE statement.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
'This example sets MOVE.DEC to 5,000 rpm/sec
and does a
'relative move of 10 motor revolutions.
MOVE.RUNSPEED = 1000
MOVE.ACC = 10000
MOVE.DEC = 10000
MOVE.RELATIVEDIST = 655360
MOVE.GOREL
```

Related Topics

MOVE.ACC

6.24.4 MOVE.DIR

General Information	
Type	R/W parameter
Description	MOVE.DIR specifies the direction the motor turns when a MOVE.GOVEL statement is executed.
Units	none
Range	0 or 1
Default Value	0
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1096	No	32 bit	No	M_01-05-11-000

Description

MOVE.DIR specifies the direction the motor turns when a MOVE.GOVEL statement is executed. It has no effect on any other motion statements. If MOVE.DIR = 0, the motor turns in the positive direction. If MOVE.DIR = 1, the motor turns in the negative direction.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

6.24.5 MOVE.DWELLTIME

General Information	
Type	R/W
Description	Adds a dwell time at the end of a MOVE.GOREL or MOVE.GOABS move that will elapse before MOVE.MOVING is set equal to 0.
Units	msecs
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1182	No	32 bit	No	M_01-05-11-000

Description

MOVE.DWELLTIME will add a pause after a Relative or absolute move is complete before the MOVE.MOVING flag is set to 0.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
MOVE.RUNSPEED = 1000
MOVE.DWELLTIME = 5000
MOVE.RELATIVEDIST = 65536
MOVE.GOREL
While     MOVE.MOVING = 1 : Wend     'Wait for
move to complete and pause 5 seconds
```

Related Topics

MOVE.GOABS | MOVE.GOREL

6.24.6 MOVE.GOABS

General Information	
Type	Command
Description	MOVE.GOABS moves the motor to the position specified by MOVE.TARGETPOS.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1098	No	Command	No	M_01-05-11-000

Description

MOVE.GOABS (Go to Absolute Position) causes the motor to move to the position specified by MOVE.TARGETPOS. This is an absolute position referenced to the position where PL.FB = 0.

Program execution continues with the line immediately following the MOVE.GOABS statement as soon as the move is initiated. Program execution does not wait until the move is complete.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.POSCOMMAND | MOVE.ACC | MOVE.DEC | MOVE.RUNSPEED | MOVE.ABORT | MOVE.GOHOME | MOVE.MOVING | MOVE.INPOSITION | Statement Table

6.24.7 MOVE.GOHOME

General Information	
Type	Command
Description	MOVE.GOHOME causes the motor to move to the position specified where PL.FB = 0.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1102	No	Command	No	M_01-05-11-000

Description

MOVE.GOHOME causes the motor to move to the position specified where PL.FB = 0. MOVE.GOHOME is identical to MOVE.GOABS with MOVE.TARGETPOS = 0. The motor speed follows a velocity profile as specified by MOVE.ACC, MOVE.DEC, and MOVE.RUNSPEED. This profile may be modified during the move using MOVE.GOUPDATE.

Program execution continues with the line immediately following the MOVE.GOHOME statement as soon as the move is initiated. Program execution does not wait until the move is complete.

The drive must be enabled in order for any motion to take place.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.POSCOMMAND | MOVE.ACC | MOVE.DEC | MOVE.RUNSPEED |MOVE.ABORTMOVE.GOABS | MOVE.MOVING | MOVE.INPOSITION

6.24.8 MOVE.GOREL

General Information	
Type	Command
Description	MOVE.GOREL moves the motor a distance specified by MOVE.RELATIVEDIST.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1106	No	Command	No	M_01-05-11-000

Description

MOVE.GOREL (Go Relative) moves the motor a distance specified by MOVE.RELATIVEDIST.

The motor speed follows a velocity profile as specified by MOVE.ACC, MOVE.DEC, and MOVE.RUNSPEED. This profile may be modified during the move using MOVE.GOUPDATE.

Program execution continues with the line immediately following the MOVE.GOREL statement as soon as the move is initiated. Program execution does not wait until the move is complete.

The drive must be enabled in order for any motion to take place.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.POSCOMMAND | MOVE.ACC | MOVE.DEC | MOVE.RUNSPEED | MOVE.ABORT | MOVE.GOABS | MOVE.MOVING | MOVE.INPOSITION | Statement Table

6.24.9 MOVE.GOUPDATE

General Information	
Type	Command
Description	MOVE.GOUPDATE updates a move in progress with new move parameters.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1108	No	Command	No	M_01-05-11-000

Description

MOVE.GOUPDATE updates a move in progress with new move parameters. This allows you to change motion on-the-fly. MOVE.GOUPDATE updates MOVE.ACC, MOVE.DEC, MOVE.DIR, and MOVE.RUNSPEED.

Program execution continues with the line immediately following the MOVE.GOUPDATE statement as soon as the move is initiated. Program execution does not wait until the move is complete. The drive must be enabled in order for any motion to take place.

MOVE.GOUPDATE does not initiate motion if there is no move in progress, the MOVE.GOUPDATE statement is ignored.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.GOREL | MOVE.GOABS | MOVE.GOVEL | MOVE.GOHOME

6.24.10 MOVE.GOVEL

General Information	
Type	Command
Description	MOVE.GOVEL moves the motor at a constant speed specified by MOVE.RUNSPEED and direction specified by MOVE.DIR.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1110	No	Command	No	M_01-05-11-000

Description

MOVE.GOVEL (Go at Velocity) moves the motor at a constant speed specified by MOVE.RUNSPEED and direction specified by MOVE.DIR.

The motor speed follows a velocity profile as specified by MOVE.ACC, MOVE.DEC, and MOVE.RUNSPEED. This profile may be modified during the move using MOVE.GOUPDATE.

Program execution continues with the line immediately following MOVE.GOVEL as soon as the move is initiated. Program execution does not wait until the move is complete.

The drive must be enabled in order for any motion to take place.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.GOREL | MOVE.GOABS | MOVE.GOVEL | MOVE.GOHOME | Statement Table

6.24.11 MOVE.INPOSITION

General Information	
Type	R/O
Description	Indicates whether or not the motor has achieved command position.
Units	none
Range	0 or 1
Default Value	none
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1112	No	32 bit	No	M_01-05-11-000

Description

MOVE.INPOSITION is used for monitoring move commands to ensure that the desired motion has been completed. MOVE.INPOSITION is always 0 (False) or 1 (True).

MOVE.INPOSITION is 1 (True) if all the following are true:

- Drive is enabled.
- MOVE.MOVING = 0
- PL.ERR less than MOVE.INPOSLIMIT

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.MOVING

6.24.12 MOVE.INPOSLIMIT

General Information	
Type	R/W
Description	Specifies the tolerance of Position Error (PL.ERR) within which the MOVE.INPOSITION flag will be set to 1 (True).
Units	Position units
Range	TBD
Default Value	TBD
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1114	Yes	64 bit	Yes	M_01-05-11-000

Description

Specifies the tolerance of Position Error (PL.ERR) within which the MOVE.INPOSITION flag will be set to 1 (True). Set MOVE.INPOSLIMIT before using MOVE.INPOSITION.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.INPOSITION

6.24.13 MOVE.MOVING

General Information	
Type	R/O
Description	Indicates whether or not the commanded motion profile is complete.
Units	N/A
Range	0 or 1
Default Value	0
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1118	No	32 bit	No	M_01-05-11-000

Description

MOVE.MOVING only indicates whether or not the commanded motion profile for MOVE.GOREL, MOVE.GOABS, MOVE.GOHOME, and MOVE.GOVEL is complete. Even when the commanded motion profile is completed (MOVE.MOVING = 0), there may still be motor motion as the result of settling time and/or electronic gearing. MOVE.MOVING is not applicable for EGEAR commands.

0 - commanded motion complete

1 - move in progress

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
MOVE.RELATIVEDIST = 10000
MOVE.GOREL
While                MOVE.MOVING : Wend
Pause(0.5)
MOVE.RELATIVEDIST = -MOVE.RELATIVEDIST
MOVE.GOREL
```

Related Topics

MOVE.INPOSITION | MOVE.INPOSITION | MOVE.GOREL | MOVE.GOABS | MOVE.G-OHOME | MOVE.GOVEL

6.24.14 MOVE.POSCOMMAND

General Information	
Type	R/W
Description	Current Position Command from Trajectory Generator.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.A-CCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1120	Yes	64 bit	Yes	M_01-05-11-000

Description

MOVE.POSCOMMAND contains the current position command from the trajectory generator.

The value of MOVE.POSCOMMAND is affected by MOVE.POSMODULO and MOVE.P-OSPOLARITY.

MOVE.POSCOMMAND can be used to determine the position being commanded. You can write to MOVE.POSCOMMAND at any time; to establish a new home position (where MOVE.POSCOMMAND = 0). Writing to MOVE.POSCOMMAND will not affect motor motion.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
'set electrical home position when DIN1.STATE
 goes to 0.
'--------------------------------------
MOVE.DIR = 0 : MOVE.RUNSPEED = 100 : MOVE.GOVEL
When              DIN1.STATE = 0, Continue
MOVE.ABORT
While              MOVE.MOVING : Wend
MOVE.POSCOMMAND = 0
```

6.24.15 MOVE.RELATIVEDIST

General Information	
Type	R/W
Description	Specifies the distance the motor turns during a relative move (MOVE.GOREL).
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1134	Yes	64 bit	Yes	M_01-05-11-000

Description

MOVE.RELATIVEDIST specifies the distance the motor moves during a relative move (MOVE.GOREL). Specify MOVE.RELATIVEDIST before initiating MOVE.GOREL.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Example

```
'This example sets Move.RelativeDist to 655360
(10 motor revolu
'tions, assuming units is 16 bit position units
or 65536
'counts/rev) and does a relative move.
MOVE.RUNSPEED  = 1000
MOVE.ACC = 10000
MOVE.DEC =  5000
MOVE.RELATIVEDIST = 655360
MOVE.GOREL
```

Related Topics

MOVE.ACC | MOVE.DEC | MOVE.RUNSPEED

6.24.16 MOVE.RUNSPEED

General Information	
Type	R/W
Description	Sets the maximum speed allowed during a relative (MOVE.GOREL) or absolute (MOVE.GOABS) move, and sets the commanded speed during a velocity move (MOVE.GOVEL).
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1138	Yes	64 bit	No	M_01-05-11-000

Description

Sets the maximum speed allowed during an incremental (MOVE.GOREL) or absolute (MOVE.GOABS) move, and sets the commanded speed during a velocity move (MOVE.GOVEL). Specify MOVE.RUNSPEED before initiating any move commands.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.GOVEL | MOVE.GOREL

6.24.17 MOVE.SCURVETIME

General Information	
Type	RW
Description	Sets the amount of S-curve smoothing applied to all velocity profiles.
Units	0 (trapezoidal profile)
Range	0 to 0.256 (0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.128, 0.256)
Default Value	TBD
Data Type	Float

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1142	No	32 bit	No	M_01-05-11-000

Description

MOVE.SCURVETIME sets the amount of S-curve smoothing applied to all velocity profiles. The greater the value of MOVE.SCURVETIMEe, the smoother (lower jerk) the profile.

Specifying a non-zero value for MOVE.SCURVETIME increases move time by MOVE.SCURVETIME. For example, a trapezoidal move (MOVE.SCURVETIME = 0) that takes 0.500 seconds to complete, takes 0.756 seconds to complete if MOVE.SCURVETIME is set to 0.256.

Change MOVE.SCURVETIME only when the motor is not moving (MOVE.MOVING = 0). If you attempt to change MOVE.SCURVETIME while the motor is moving a RunTime error is produced.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.ACC | MOVE.DEC | MOVE.GOREL | MOVE.GOABS | MOVE.GOHOME | MOVE.GOVEL

6.24.18 MOVE.TARGETPOS

General Information	
Type	R/W
Description	MOVE.TARGETPOS specifies the target position for an absolute (MOVE.GOABS) move.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1146	Yes	64 bit	Yes	M_01-05-11-000

Description

MOVE.TARGETPOS specifies the target position for an absolute (MOVE.GOABS) move. MOVE.TARGETPOS is an absolute position referenced to the electrical home position (the position where PL.FB = 0).

Set MOVE.TARGETPOS before initiating a MOVE.GOABS.

NOTE MOVE parameters are only valid when DRV.OPMODE = 3 and DRV.CMDSOURCE = 5.

Related Topics

MOVE.GOABS

6.25 PL Parameters

This section describes the PL parameters.

6.25.1 PL.CMD

General Information	
Type	R/O Parameter
Description	Reads the position command directly from the entry to the position loop.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Float
See Also	PL.FB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	570	Yes	64 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

PL.CMD reads the position command as it is received in the position loop entry.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.2 PL.ERR

General Information	
Type	R/O Parameter
Description	Reads the position error present when the drive is controlling the position loop.
Units	counts, rad, deg, (custom units)
Range	N/A
Default Value	N/A
Data Type	Float
See Also	PL.FB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	35C5h/0 60F4h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	574	Yes	64 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

PL.ERR reads the position error present when the drive is controlling the position loop. PL.ERR is the difference between the actual position of the motor shaft (PL.FB) and the commanded position of the drive (PL.CMD). If the drive is not in the position operating mode (DRV.OPMODE = 2), then the PL.ERR value is not generated by the drive and this parameter is read as 0.

AKD BASIC Notes

NOTE When you enable the position error interrupt (by setting INTR.PL.ERR = 1), the Position Error fault is disabled. In situations where this fault would have occurred, a position error interrupt is generated instead.

Related Topics

PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.3 PL.ERRFTHRESH

General Information	
Type	NV Parameter
Description	Sets the maximum position error.
Units	Depends on UNIT.ACCROTARY or UNIT.A-CCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	Rotary: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 7,495,067.136 rad 0.000 to 429,436,076.032 deg 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts Linear: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 1,192,877.952*MOTOR.PITCH mm 0.000 to 1,192,878,014.464*MOTOR.PITCH µm 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts
Default Value	Rotary: 42,949,672,960.000 counts 62.832 rad 3,600.000 deg 50.000 (custom units) 655,360.000 16-bit counts Linear: 42,949,672,960.000 counts 10.000*MOTOR.PITCHMOTOR.PITCH mm 10,000.000*MOTOR.PITCH µm 50.000 (custom units) 655,360.000 16-bit counts
Data Type	Float
See Also	PL.ERR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	35C7h/0 6065h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	580	Yes	64 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

This parameter sets the maximum position error. If the position error PL.ERR is larger than PL.ERRFTHRESH the drive generates a fault. If PL.ERRFTHRESH is set to 0, the maximum position error is ignored.

Example

Set position rotary units to 2 (degrees). Setting PL.ERRFTHRESH to 1000 states that is the position error is larger than 1000 degrees, the drive will generate a fault.

```
UNIT.PROTARY 2
PL.ERRFTHRESH 1000
```

Related Topics

PL.ERR | PL.ERRMODE | PL.ERRWTHRESH

6.25.4 PL.ERRMODE

General Information	
Type	NV Parameter
Description	Sets the type of following error warning and fault usage.
Units	0- Standard following error 1-Enhanced following error
Range	0 to 1
Default Value	0
Data Type	Boolean
See Also	PL.ERR , PL.ERRFTHRESH , PL.ERRWTHRESH
Start Version	M_01-02-09-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	578	No	8 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

PL.ERRMODE sets the type of following error warning and fault usage.

Mode 0 - following error magnitude fault

In Mode 0, the values of PL.ERRFTHRESH and PL.ERRWTHRESH are compared against the value of PL.ERR. If the absolute value of PL.ERR is larger than PL.ERRWTHRESH, then a warning is generated. If the absolute value of PL.ERR is larger than PL.ERRFTHRESH, then a fault is generated.

Mode 1 - deviation from predicted trajectory fault

In Mode 1, the values of PL.ERRFTHRESH and PL.ERRWTHRESH are compared against the following value:

<error> = abs(PL.ERR - [(VL.CMD – 1*VL.FF) / PL.KP])

If the absolute value of <error> is larger than PL.ERRWTHRESH for a consecutive period of 100 ms, then a warning is generated. If the absolute value of <error> is larger than PL.ERR-FTHRESH for a consecutive period of 100 ms, then a fault is generated.

In mode 1, if PL.KI is not 0 then the following error prediction mechanism is turned off. When the drive is disabled, the following error limit tests are turned off and the warnings are cleared. A value of 0 in PL.ERRFTHRESH or PL.ERRWTHRESH disables the respective functionality.

Example

Assuming

PL.ERRMODE = 0, PL.ERRFTHRESH=1.2, PL.ERRWTHRESH=1, then PL.ERR reads 1.1.

In this case the warning is generated, but the fault is not.

Assuming PL.ERRMODE = 0, PL.ERRFTHRESH=1.2, PL.ERRWTHRESH=1, then PL.ERR reads 1.3.

In this case the warning is generated, as well as the fault.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRWTHRESH

6.25.5 PL.ERRWTHRESH

General Information	
Type	NV Parameter
Description	Sets the position error warning level.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.A-CCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	Rotary: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 7,495,067.136 rad 0.000 to 429,436,076.032 deg 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts Linear: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 1,192,877.952*MOTOR.PITCH mm 0.000 to 1,192,878,014.464*MOTOR.PITCH µm 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts
Default Value	0.000 deg
Data Type	Float
See Also	PL.ERR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3483h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	584	Yes	64 bit	No	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

If this value is not equal 0 and the position error PL.ERR is larger than this value, the drive will generate a warning.

If PL.ERRWTHRESH is set to 0 the warning is not issued.

Example

Set position rotary units to 2 degrees. If you set PL.ERRWTHRESH to 100 and the position error is larger than 100 degrees, then the drive will generate a warning.

UNIT.PROTARY 2

PL.ERRWTHRESH 100

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE

6.25.6 PL.FB

General Information	
Type	R/O Parameter
Description	Reads the position feedback value.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.A-CCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Float
See Also	FB1.OFFSET
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	6064h /0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	588	Yes	64 bit	Yes	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

PL.FB returns the position feedback value.

Note that this value is not the pure feedback value read from the feedback device, but also includes the value of the FB1.OFFSET and an internal offset set by the user. If a new value is written to MOVE.POSCOMMAND then PL.FB will be automatically changed such that PL.ERROR (the difference between them) is unchanged.

AKD BASIC Example

```
Print PL.FB, MOVE.POSCOMMAND
MOVE.POSCOMMAND = 0
Print PL.FB, MOVE.POSCOMMAND
```

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.7 PL.FBSOURCE

General Information	
Type	NV Parameter
Description	Sets the feedback source for the position loop.
Units	N/A
Range	Range will differ depending on drive model. 0 to 1 (for AKD-x-xxxxx-NAxx-xxxx) 0 to 2 (for AKD-x-xxxxx-NBxx-xxxx)
Default Value	0
Data Type	Integer
See Also	VL.FBSOURCE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	592	No	8 bit	No	M_01-03-00-000

Description

This parameter determines the feedback source that the position loop uses. A value of 0 for this parameter selects the primary feedback, a value of 1 selects the secondary feedback. If you use the secondary feedback as the source for the position loop, then FB2.MODE mode should be set as 0 (A/B signals). A/B signals are the only supported feedback type as secondary feedback into the position loop. Other settings for FB2.MODE are intended as pulse inputs or a gearing command when PL.FBSOURCE remains 0.

0	Primary Feedback connected to X10.
1	Secondary Feedback (DRV.HANDWHEEL) connected to X7 or X9.
2	Tertiary Feedback connected to X9 (only supported with AKD-x-xxxxx-NBxx-xxxx).

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.8 PL.INTINMAX

General Information	
Type	NV Parameter
Description	Limits the input of the position loop integrator by setting the input saturation.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	Rotary: 0.000 to 18,446,744,073,709.000 counts 0.000 to 26,986.052 rad 0.000 to 1,546,188.288 deg 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts Linear: 0.000 to 18,446,744,073,709.000 counts 0.000 to 4,294.968*MOTOR.PITCH mm 0.000 to 4,294,967.296*MOTOR.PITCH µm 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts
Default Value	Rotary: 3,999,989,760.000 counts 5.852 rad 335.275 deg 4.657 (custom units) 61,035.000 16-bit counts Linear: 3,999,989,760.000 counts 0MOTOR.PITCH mm 9MOTOR.PITCH µm 4.657 (custom units) 61,035.000 16-bit counts
Data Type	Float
See Also	PL.FB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3481h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	594	Yes	64 bit	No	M_01-03-00-000

Description

PL.INTINMAX limits the input of the position loop integrator by setting the input saturation. When used in concert with PL.INSATOUT, this variable allows you to make the position loop

integrator effective near the target position. Far from the target position, however,the integrator is not dominant in the loop dynamics.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.9 PL.INTOUTMAX

General Information	
Type	NV Parameter
Description	Limits the output of the position loop integrator by setting the output saturation.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	Rotary: 0.000 to 18,446,744,073,709.000 counts 0.000 to 26,986.052 rad 0.000 to 1,546,188.288 deg 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 counts16 bit Linear: 0.000 to 18,446,744,073,709.000 counts 0.000 to 4,294.968*MOTOR.PITCH mm 0.000 to 4,294,967.296*MOTOR.PITCH µm 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts
Default Value	Rotary: 3,999,989,760.000 counts 5.852 rad 335.275 deg 4.657 (custom units) 61,035.000 16-bit counts Linear: 3,999,989,760.000 counts 0MOTOR.PITCH mm 9MOTOR.PITCH µm 4.657 (custom units) 61,035.000 16-bit counts
Data Type	Float
See Also	PL.INTINMAX
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3481h/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	598	Yes	64 bit	No	M_01-03-00-000

Description

PL.INTOUTMAX limits the output of the position loop integrator by setting the output saturation.

When used in concert with PL.INTINMAX, this variable allows you to make the position loop integrator effective near the target position. Far from the target position, however, the integrator is not dominant in the loop dynamics.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.10 PL.KI

General Information	
Type	NV Parameter
Description	Sets the integral gain of the position loop.
Units	Hz
Range	0 to 250 Hz
Default Value	0 Hz
Data Type	Float
See Also	PL.KP, PL.KD
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3480h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	602	No	32 bit	No	M_01-03-00-000

Description

PL.KI sets the integral gain of the position regulator PID loop.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.11 PL.KP

General Information	
Type	NV Parameter
Description	Sets the proportional gain of the position regulator PID loop.
Units	(rev/s)/rev
Range	0 to 2,147,483.008 (rev/s)/rev
Default Value	100 rps/rev
Data Type	Float
See Also	PL.KI , PL.KD
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3542h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	604	No	32 bit	No	M_01-03-00-000

Description

PL.KP sets the proportional gain of the position regulator PID loop.

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.12 PL.MODP1

General Information	
Type	R/W parameter
Description	Sets modulo range parameter.
Units	Depends on UNIT.PROTARY and UNIT.PLI-NEAR
Range	N/A
Default Value	N/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3637h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	604	Yes	64 bit	Yes	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

This parameter is either the beginning or the end of the modulo range, depending on whether this value is smaller or larger than PL.MODP2. If you set PL.MODP1 equal to PL.MODP2, an error message occurs.

Condition	Beginning of the modulo-range	End of the modulo-range
PL.MODP1 < PL.MODP2	PL.MODP1	PL.MODP2
PL.MODP2 < PL.MODP1	PL.MODP2	PL.MODP1

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.13 PL.MODP2

General Information	
Type	R/W Parameter
Description	Sets the beginning or end modulo range parameter.
Units	Depends on UNIT.PROTARY and UNIT.PLINEAR .
Range	N/A
Default Value	N/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3638h/0	M_01-00-00-000
Modbus	610 (64-bit)	M_01-03-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	610	Yes	64 bit	Yes	M_01-03-00-000

AKD BASIC Information	
Data Type	Integer

Description

This parameter is either the beginning or the end of the modulo range, depending on whether this value is smaller or larger than PL.MODP1 .

Condition	Beginning of the modulo range	End of the modulo range
PL.MODP1 < PL.MODP2	PL.MODP1	PL.MODP2
PL.MODP2 < PL.MODP1	PL.MODP2	PL.MODP1

Related Topics

PL.ERR | PL.ERRFTHRESH | PL.ERRMODE | PL.ERRWTHRESH

6.25.14 PL.MODPDIR

General Information	
Type	R/W Parameter
Description	Sets the direction for absolute motion tasks.
Units	N/A
Range	0 to 2
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3430h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	614	No	8 bit	No	M_01-03-00-000

Description

This parameter defines the direction of an absolute motion task when the modulo position has been activated. For more details about absolute motion tasks, see .0.1 Motion Tasks. For absolute motion tasks, you can only select a target position within the modulo range.

PL.MODPDIR Settings

Value	Motion	Description
0	Inside Range	The motor moves in a negative direction if the target position of the absolute motion task is less than the current position. The motor moves in positive direction if the target position of the absolute motion task is greater than the current position.
1	Positive	The motor always moves in a positive direction relative to the target position of the absolute motion task.
2	Negative	The motor always moves in a negative direction relative to the target position of the absolute motion task.
3	Shortest Distance	The motor always moves the shortest distance in order to reach the target position within the modulo-range.

6.25.15 PL.MODPEN

General Information	
Type	R/W Parameter
Description	Enables the modulo position.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	35CFh/0	M_01-00-00-000

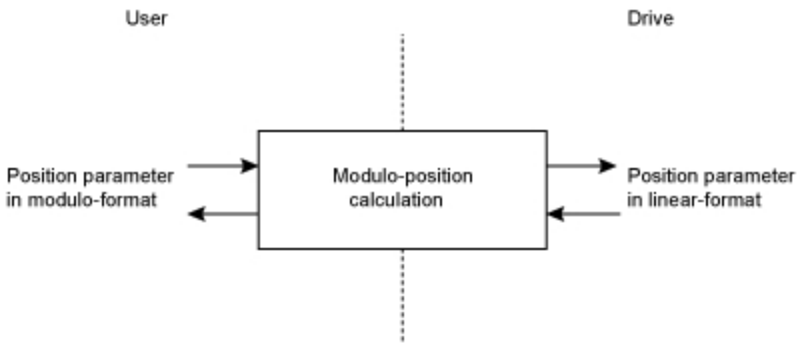
Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	616	No	8 bit	No	M_01-03-00-000

Description

A value of 0 disables the modulo-position and a value of 1 enables the modulo-position feature. The modulo-position feature can be used for circular applications such as round tables.

The position loop of the drive uses always a linear position variable but the data-exchange between the user and the drive uses the modulo-position calculation in order to convert values from linear format into modulo format and vice versa.

The following figure shows the interface between the user and the drive for PL.MODPEN=1:



6.26 PLS Parameters

This section describes the PLS parameters.

6.26.1 PLS.EN

General Information	
Type	R/W Parameter
Description	Enables programmable limit switch (PLS).
Units	N/A
Range	0 to 255
Default Value	0
Data Type	Integer
See Also	PLS.MODE , PLS.RESET, PLS.STATE, PLS.UNITS, PLS.P1 to PLS.P8, PLS.WIDTH1 to PLS.WIDTH8, PLS.T1 to PLS.T8
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	34A3h/1	M_01-02-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	618	No	16 bit	No	M_01-03-00-000

Description

PLS.EN is a bit variable which determines the mode of an individual PLS. Eight PLSs are available in the drive.

Example

Bit Value	Behavior
Bit 0 = 0	Disables PLS 1
Bit 0 = 1	Enables PLS 1
Bit 7 = 0	Disables PLS 8
Bit 7 = 1	Enables PLS 8

6.26.2 PLS.MODE

General Information	
Type	R/W Parameter
Description	Selects programmable limit switch mode.
Units	N/A
Range	0 to 255
Default Value	0
Data Type	Integer
See Also	PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.P1 to PLS.P8, PLS.WIDTH1 to PLS.WIDTH8, PLS.T1 to PLS.T8
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	34A3h/3	M_01-02-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	620	No	16 bit	No	M_01-03-00-000

Description

PLS.MODE is a bit variable which determines the mode of an individual PLS. Eight PLSs are available in the drive.

Example

Bit Value	Behavior
Bit 0 = 0	PLS 1 is monitored continuously.
Bit 0 = 1	PLS 1 is monitored until it is triggered once (single-shot method). The PLS observation can be re-armed using the PLS.RESET command.
Bit 7 = 0	PLS 8 is monitored continuously.
Bit 7 = 1	PLS 8 is monitored until it is triggered once (single-shot method). The PLS observation can be re-armed using the PLS.RESET command.

6.26.3 PLS.P1 TO PLS.P8

General Information	
Type	R/W Parameter
Description	Sets the trigger point for programmable limit switches.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR
Range	N/A
Default Value	0
Data Type	Float
See Also	UNIT.PROTARY
Start Version	M_01-02-03-000

AKD BASIC Information	
Data Type	Integer

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	34A0h/1	PLS.P1	M_01-02-03-000
	34A0h/2	PLS.P2	
	34A0h/3	PLS.P3	
	34A0h/4	PLS.P4	
	34A0h/5	PLS.P5	
	34A0h/6	PLS.P6	
	34A0h/7	PLS.P7	
	34A0h/8	PLS.P8	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	622	PLS.P1	Yes	64 bit	Yes	M_01-03-00-000
	626	PLS.P2				
	630	PLS.P3				
	634	PLS.P4				
	638	PLS.P5				
	642	PLS.P6				
	646	PLS.P7				
	650	PLS.P8				

Description

PLS.P1 to PLS.P8 define the trigger point of the PLS. For further information about how these parameters affect PLS behavior, see the PLS.UNITS parameter description.

6.26.4 PLS.RESET

General Information	
Type	R/O Parameter
Description	Resets programmable limit switch.
Units	N/A
Range	0 to 255
Default Value	N/A
Data Type	Integer
See Also	PLS.EN, PLS.MODE, PLS.STATE, PLS.UNITS, PLS.Px (x=1…8), PLS.WIDTHx (x=1…8), PLS.Tx (x=1…8)
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	34A3h/2	M_01-02-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	654	No	16 bit	No	M_01-03-00-000

Description

This parameter is a bit variable and is used in order to re-arm the corresponding PLS.STATE observation for another single-shot PLS use (see also PLS.MODE).

Example

Bit Value	Behavior
Bit 0 = 0	The PLS 1 observation (PLS.STATE bit 0) is not re-armed.
Bit 0 = 1	The PLS 1 observation (PLS.STATE bit 0) is re-armed.
Bit 7 = 0	The PLS 8 observation (PLS.STATE bit 7) is not re-armed.
Bit 7 = 1	The PLS 8 observation (PLS.STATE bit 7) is re-armed.

6.26.5 PLS.STATE

General Information	
Type	R/O Parameter
Description	Reads the programmable limit switch state.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	PLS.EN, PLS.RESET, PLS.UNITS, PLS.MODE, PLS.P1 TO PLS.P8, PLS.WIDTH1 TO PLS.WIDTH8, PLS.T1 TO PLS.T8
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	34A3h/4	M_01-02-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	656	No	16 bit	No	M_01-03-00-000

Description

This parameter is a bit variable and displays the current status of the individual programmable limit switches.

Example

Bit 0 = 0: Programmable Limit Switch 1 (PLS 1) is not active.

Bit 0 = 1: Programmable Limit Switch 1 (PLS 1) is active.

Bit 7 = 0: Programmable Limit Switch 8 (PLS 8) is not active.

Bit 7 = 1: Programmable Limit Switch 8 (PLS 8) is not active.

6.26.6 PLS.T1 TO PLS.T8

General Information	
Type	R/W parameter
Description	Sets programmable limit switch time
Units	ms
Range	0 to 65,536 ms
Default Value	500 ms
Data Type	Integer
See Also	PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.MODE, PLS.WIDTH1 TO PLS.WIDTH8, PLS.P1 TO PLS.P8
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	34A2h /1	PLS.T1	M_01-02-03-000
	34A2h /2	PLS.T2	
	34A2h /3	PLS.T3	
	34A2h /4	PLS.T4	
	34A2h /5	PLS.T5	
	34A2h /6	PLS.T6	
	34A2h /7	PLS.T7	
	34A2h /8	PLS.T8	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	658	PLS.T1	No	16 bit	No	M_01-03-00-000
	660	PLS.T2				
	662	PLS.T3				
	664	PLS.T4				
	666	PLS.T5				
	668	PLS.T6				
	670	PLS.T7				
	672	PLS.T8				

Description

These parameters define the time of the PLS pulse for time-based PLS handling.

For further information about the PLS functionality, especially the meaning of the PLS.T1 to PLS.T8 parameter, refer to the PLS.UNITS parameter.

6.26.7 PLS.UNITS

General Information	
Type	R/W parameter
Description	Sets programmable limit switch (PLS) units.
Units	N/A
Range	0 to 255
Default Value	0
Data Type	Integer
See Also	PLS.EN , PLS.RESET , PLS.STATE , PLS.MODE , PLS.P1 TO PLS.P8 PLS.WIDTH1 TO PLS.WIDTH8 , PLS.T1 TO PLS.T8
Start Version	M_01-02-03-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	34A4h/0	M_01-02-03-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	674	No	8 bit	No	M_01-03-00-000

Description

PLS.UNITS is a bit variable which determines the behavior of the eight PLSs available in the drive. This parameter is used to select the units for the PLS pulse.

Examples

Bit Value	Behavior
Bit 0 = 0 Position-based PLS handling.	The PLS.STATE parameter displays an active PLS 1 when the position is within the range of PLS.P1 + PLS.WIDTH1 (PLS.P1 <= PL.FB <= PLS.P1+PLS.WIDTH1). When the parameter PLS.WIDTH1 has been set to the value of 0, this bit will be activated as soon as PLS.FB >= PL.P1.
Bit 0 = 1 Time-based PLS handling.	After PLS.P1 is crossed, the PLS.STATE parameter displays an active PLS 1 for a PLS.T1 ms period of time.
Bit 7 = 0 Position-based PLS handling.	The PLS.STATE parameter displays an active PLS 8 when the position is within the range of PLS.P8 + PLS.WIDTH8 (PLS.P8 <= PL.FB <= PLS.P8+PLS.WIDTH8). When the parameter PLS.WIDTH8 has been set to the value of 0, this bit will be activated as soon as PLS.FB >= PL.P8.
Bit 7 = 1 Time-based PLS handling.	After PLS.P8 has been crossed. the PLS.STATE parameter displays an active PLS 8 for a PLS.T8 ms period of time.

Continuous position-based PLS handling

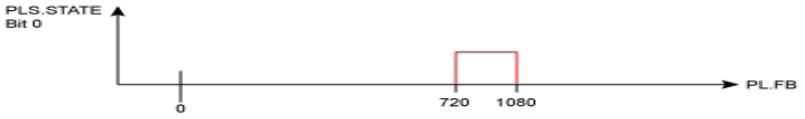
PLS.P1 = 720

PLS.WIDTH1 = 360

PLS.UNITS bit 0 (for PLS 1) = low; PLS.T1 is not considered.

PLS.EN bit 0 (for PLS 1) = high

PLS.MODE bit 0 (for PLS 1) = low



Time-based PLS handling

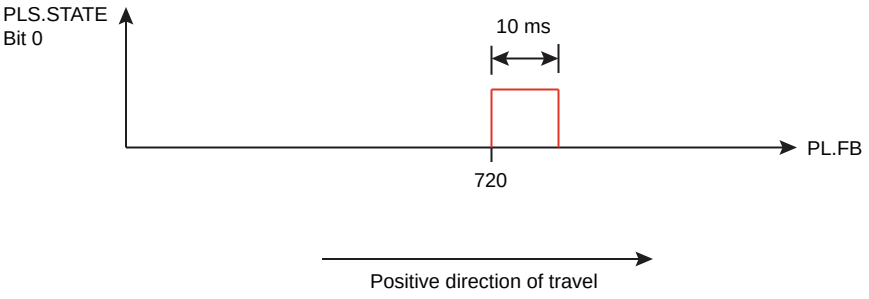
PLS.P1 = 720

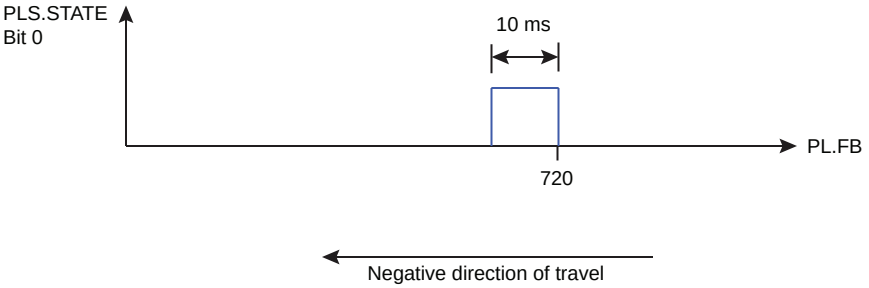
PLS.T1 = 10

PLS.UNITS bit 0 (for PLS 1) = low; PLS.WIDTH1 is not considered.

PLS.EN bit 0 (for PLS 1) = high

PLS.MODE bit 0 (for PLS 1) = low





6.26.8 PLS.WIDTH1 TO PLS.WIDTH8

General Information	
Type	R/W parameter
Description	Programmable Limit Switch Width
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR
Range	N/A
Default Value	0
Data Type	Float
See Also	PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.MODE, PLS.P1 TO PLS.P8, PLS.T1 TO PLS T8
Start Version	M_01-02-03-000

AKD BASIC Information	
Data Type	Integer

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	34A1h/1	PLS.WIDTH1	M_01-02-03-000
	34A1h/2	PLS.WIDTH2	
	34A1h/3	PLS.WIDTH3	
	34A1h/4	PLS.WIDTH4	
	34A1h/5	PLS.WIDTH5	
	34A1h/6	PLS.WIDTH6	
	34A1h/7	PLS.WIDTH7	
	34A1h/8	PLS.WIDTH8	
Modbus	676 (64-Bit)	PLS.WIDTH1	M_01-03-00-000
	680 (64-Bit)	PLS.WIDTH2	
	684 (64-Bit)	PLS.WIDTH3	
	688 (64-Bit)	PLS.WIDTH4	
	692 (64-Bit)	PLS.WIDTH5	
	696 (64-Bit)	PLS.WIDTH6	
	700 (64-Bit)	PLS.WIDTH7	
	704 (64-Bit)	PLS.WIDTH8	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	676	PLS.WIDTH1	Yes	64 bit	Yes	M_01-03-00-000
	680	PLS.WIDTH2				
	684	PLS.WIDTH3				
	688	PLS.WIDTH4				
	692	PLS.WIDTH5				
	696	PLS.WIDTH6				
	700	PLS.WIDTH7				
	704	PLS.WIDTH8				

Description

These parameter define the width of the PLS pulse for position-based PLS handling. For further information about the PLS functionality, especially the meaning of the PLS.WIDTH1 to PLS.WIDTH8 parameter, refer to the PLS.UNITS parameter.

6.27 REC Parameters

This section describes the REC parameters.

6.27.1 REC.ACTIVE

General Information	
Type	R/O Parameter
Description	Indicates if data recording is in progress (active).
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	REC.DONE , REC.OFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	708	No	8 bit	No	M_01-03-00-000

Description

REC.ACTIVE indicates whether or not data recording is in progress. Recording is in progress if the trigger was met and the recorder is recording all data.

6.27.2 REC.DONE

General Information	
Type	R/O Parameter
Description	Checks whether or not the recorder has finished recording.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	REC.ACTIVE, REC.OFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	710	No	8 bit	No	M_01-03-00-000

Description

REC.DONE indicates that the recorder has finished recording. This value is reset to 0 when the recorder trigger is set. The drive also resets this value when the recording has finished or when REC.OFF is executed.

6.27.3 REC.OFF

General Information	
Type	R/W Parameter
Description	Turns the recorder OFF.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	REC.ACTIVE, REC.DONE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	716	No	Command	No	M_01-03-00-000

Description

REC.OFF turns the recorder off. In order to set the recorder again, the recorder must first be armed and then a trigger set.

6.27.4 REC.TRIG

General Information	
Type	Command
Description	Triggers the recorder.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	720	No	Command	No	M_01-03-00-000

Description

REC.TRIG starts the trigger according to the trigger type defined by REC.TRIGTYPE.

REC.TRIG sets the value of REC.DONE to 0.

After calling REC.TRIG, the data that was recorded by previous recording is deleted and cannot be retrieved.

No REC parameters can be set after a call to REC.TRIG until the recorder has finished or until REC.OFF is executed.

6.28 REGEN Parameters

This section describes the REGEN parameters.

6.28.1 REGEN.POWER

General Information	
Type	R/O parameter
Description	Reads regen resistor's calculated power.
Units	Watt
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3416h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	734	Yes	64 bit	No	M_01-03-00-000

Description

This parameter reads regen resistor's calculated power, which is determined as follows:

$(V^2 / R) * DutyCycle$

6.28.2 REGEN.REXT

General Information	
Type	NV Parameter
Description	Sets the external, user-defined regen resistor resistance.
Units	Ω
Range	0 to 255 Ω
Default Value	0 Ω
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	35C2h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	738	No	16 bit	No	M_01-03-00-000

Description

REGEN.REXT sets the external user-defined regen resistor resistance. This variable is needed for the regen resistor temperature estimation algorithm.

6.28.3 REGEN.TEXT

General Information	
Type	R/W Parameter
Description	Sets the external regen resistor thermal protection time constant.
Units	s
Range	0.1 to 1,200 s
Default Value	100 s
Data Type	Float
See Also	REGEN.WATTEXT , REGEN.REXT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3415h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	740	No	32 bit	No	M_01-03-00-000

Description

REGEN.TEXT is a thermal time constant used to protect an external regeneration (regen) resistor from overheating and failing. Its value is the time-to-fault when input power steps from 0 to 150% of REGEN.WATTEXT . The drive's regen resistor protection algorithm continuously calculates the power dissipated in the resistor and processes that power value through a single pole low pass filter to model the regen resistor's thermal inertia. When the filtered regen power on the output of the filter exceeds REGEN.WATTEXT, a fault occurs. REGEN.TEXT sets the time constant of this thermal inertia filter.

REGEN.TEXT can often be found directly on power resistor data sheets. On the data sheets, find the peak overload curve and then find the safe allowed time to be at 150% of the regen resistor's continuous power rating. Another way regen resistor peak overload capability is often specified is by giving the energy rating in joules of the resistor. If you have the energy rating E then:

REGEN.TEXT = (1.1)*((joule limit)/REGEN.WATTEXT)

Example

The external regen resistor is rated for 250 W continuous, is 33 ohm, and has a joule rating of 500 joules. To use this resistor, the drive settings become:

REGEN.TYPE = -1 (External Regen)

REGEN.REXT = 33

REGEN.WATTEXT = 250

REGEN.TEXT = (1.1)*(500 j)/(250 W) = 2.2 sec

6.28.4 REGEN.TYPE

General Information	
Type	NV Parameter
Function	Sets the regen resistor type.
WorkBench Location (Screen/Dialog Box)	Power/Regen Resistor Type
Units	N/A
Range	–1 to 0
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3412h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	742	No	8 bit	Yes	M_01-03-00-000

Description

You can specify a user-defined external regen resistor, select an internal regen resistor, or choose from a list of predefined regen resistors. The values for REGEN.TYPE are shown below:

Type	Description
–1	External user-defined regen resistor
0	Internal regen resistor

If you specify a user-defined regen resistor, then you must also define this resistor's resistance (REGEN.REXT), heatup time (REGEN.REXT), and power (REGEN.WATTEXT).

6.28.5 REGEN.WATTEXT

General Information	
Type	R/W parameter
Description	Sets the regen resistor's power fault level for an external regen resistor.
Units	W
Range	0 to 62,000 W
Default Value	1000 W
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3414h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	744	No	16 bit	No	M_01-03-00-000

Description

Sets the regen resistor's power fault level for an external regen resistor (when REGEN.TYPE = -1).

Above this fault level, the regen resistor's PWM wil be 0 and a fault will be issued.

6.29 STO Parameters

This section describes the STO parameters.

6.29.1 STO.STATE

General Information	
Type	R/O Parameter
Description	Returns the status of the safe torque off.
Units	N/A
Range	0 to 1
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	762	No	8 bit	No	M_01-03-00-000

Description

STO.STATE returns the status of the safe torque off.

1 - Safe torque on (no safe torque off fault).

0 - Safe torque off (safe torque off fault).

Related Topics

1 Limits

0.1 Safe Torque Off (STO)

6.30 SWLS Parameters

This section describes the SWLS parameters.

6.30.1 SWLS.EN

General Information	
Type	NV Parameter
Description	Enables and disables software travel limit switches.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	U8
See Also	DRV.MOTIONSTAT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	764	No	16 bit	No	M_01-03-00-000

Description

This parameter enables the software travel limit switches. The software limit switches are only active if the axis is homed.

Example

Bit 0 = 0: Disable SWLS.LIMIT0

Bit 0 = 1: Enable SWLS.LIMIT0

Bit 1 = 0: Disable SWLS.LIMIT1

Bit 1 = 1: Enable SWLS.LIMIT1

Related Topics

1 Limits

1 Homing

HOME Parameters

6.30.2 SWLS.LIMIT0

General Information	
Type	NV Parameter
Description	Sets the position of the software travel limit switch 0.
Units	Position units
Range	-9,007,199,254,740,992 to 9,007,199,254,740,991
Default Value	0
Data Type	S64
See Also	UNIT.PROTARY , UNIT.PLINEAR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	607Dh/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	766	Yes	64 bit	Yes	M_01-03-00-000

Description

This parameter sets the compare register for the software limit switch 0. This value can be either the lower or the upper software limit switch register, depending on the configuration of the software limit switches. Whichever switch is set largest is the positive limit switch; the other switch becomes the negative limit switch. These switches can be used in addition to hardware limit switches. The software limit switches are only active if the axis is homed. For more information about homing, please refer to the HOME Parameters and DRV.MOTIONSTAT.

6.30.3 SWLS.LIMIT1

General Information	
Type	NV Parameter
Description	Sets the position of the software travel limit switch 0.
Units	Position units
Range	-9,007,199,254,740,992 to 9,007,199,254,740,991
Default Value	1,048,576.000 counts, 16-bit (firmware versions M_01-02-00-000 and above) 68,719,476,736 counts (for firmware version M_01-01-00-000)
Data Type	S64
See Also	UNIT.PROTARY , UNIT.PLINEAR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	607Dh/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	770	Yes	64 bit	Yes	M_01-03-00-000

Description

This parameter sets the compare register for the software limit switch 1. This value can be either the lower or the upper software limit switch register, depending on the configuration of the software limit switches. Whichever switch is set largest is the positive limit switch; the other switch becomes the negative limit switch. These switches can be used in addition to hardware limit switches. The software limit switches are only active if the axis is homed. For more information about homing, please refer to the HOME Parameters and DRV.MOTIONSTAT.

6.30.4 SWLS.STATE

General Information	
Type	R/O Parameter
Description	Reads the actual status of software limit switches.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	774	No	16 bit	No	M_01-03-00-000

Description

This parameter reads the status word of the software limit switches. The status word indicates the result of the compare between the software limit switch compare register and the actual position of the position loop.

Example

Bit 0 = 0: SWLS.LIMIT0 is not active.

Bit 0 = 1: SWLS.LIMIT0 is active.

Bit 1 = 0: SWLS.LIMIT1 is not active.

Bit 1 = 1: SWLS.LIMIT1 is active.

Bits 2 to 7 are currently not in use.

Related Topics

1 Limits

1 Homing

HOME Parameters

6.31 UNIT Parameters

This section describes the UNIT parameters.

6.31.1 UNIT.ACCLINEAR

General Information	
Type	NV Parameter
Description	Sets the linear acceleration/deceleration units.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	Integer
See Also	DRV.ACC , DRV.DEC , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	776	No	8 bit	No	M_01-03-00-000

Description

UNIT.ACCLINEAR sets the units type for the acceleration and deceleration parameters, when the motor type (MOTOR.TYPE) is linear.

Type	Description
0	[custom units]/s^2
1	millimeters per second squared (mm/s^2)
2	micrometers per second squared ($\mu m/s^2$)
3	Feedback counts/s^2

6.31.2 UNIT.ACCROTARY

General Information	
Type	NV Parameter
Description	Sets the rotary acceleration/deceleration units.
Units	rpm/s, rps/s, deg/s^2 , [custom units]/s^2
Range	0 to 3 rpm/s
Default Value	0 rpm/s
Data Type	Integer
See Also	DRV.ACC , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3659h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	778	No	8 bit	No	M_01-03-00-000

Description

UNIT.ACCROTARY sets the acceleration/deceleration units when the motor type (MOTOR.TYPE) is rotary.

Type	Description
0	rpm/s
1	rps/s
2	deg/s^2
3	(custom units)/s^2

6.31.3 UNIT.LABEL

General Information	
Type	NV Parameter
Description	Sets user-defined name for user-defined position units.
Units	N/A
Range	Maximum 16 characters, no spaces
Default Value	custom units
Data Type	String
See Also	UNIT.PLINEAR , UNIT.POUT
Start Version	M_01-00-00-000

Description

If you define a special position unit with UNIT.PLINEAR and UNIT.POUT , then you can give this unit a descriptive name. You can name the unit anything you wish, as long as the name is limited to 16 characters and includes no spaces. The label used for velocity and acceleration are in terms of this descriptive name.

This parameter is descriptive only and does not influence drive internal functions in any way.

6.31.4 UNIT.PIN

General Information	
Type	NV Parameter
Description	Sets gear IN for the unit conversion.
Units	User units
Range	0 to 4,294,967,295
Default Value	100
Data Type	Integer
See Also	UNIT.POUT
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	35CAh/0 6092h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	780	No	32 bit	No	M_01-03-00-000

Description

UNIT.PIN is used in conjunction with UNIT.POUT to set application specific units. This parameter is used as follows in the drive unit conversion:

- For position, this parameter sets the units as [custom units]/rev.
- For velocity, this parameter sets the units as [custom units]/s.
- For acceleration/deceleration, this parameter sets the units as [custom units]/s^2.

6.31.5 UNIT.PLINEAR

General Information	
Type	NV Parameter
Description	Sets the linear position units.
Units	N/A
Range	0 to 4
Default Value	0
Data Type	Integer
See Also	PL.FB , PL.CMD , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	782	No	8 bit	No	M_01-03-00-000

Description

UNIT.PLINEAR sets the units type for the position parameters when the motor type (MOTOR.TYPE) is linear.

Type	Description
0	32-bit counts
1	Millimeters (mm)
2	Micrometers (µm)
3	(PLINEAR/POUT) per revolution
4	16-bit counts

6.31.6 UNIT.POUT

General Information	
Type	NV Parameter
Description	Sets gear out for the unit conversion.
Units	User units.
Range	0 to 4,294,967,295
Default Value	20
Data Type	Integer
See Also	UNIT.PLINEAR
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	35CBh/0 6092h/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	784	No	32 bit	No	M_01-03-00-000

Description

UNIT.POUT is used in conjunction with UNIT.PLINEAR to set application specific units in UNIT.POUT. This parameter is used as follows in the drive unit conversion:

- For position, this parameter sets the units as [custom units]/rev.
- For velocity, this parameter sets the units as [custom units]/s.
- For acceleration/deceleration, this parameter sets the units as [custom units]/s^2.

6.31.7 UNIT.PROTARY

General Information	
Type	NV Parameter
Description	Sets the position units when the motor type (MOTOR.TYPE) is rotary.
Units	counts, rad, deg, custom units, 16-bit counts
Range	0 to 4
Default Value	4 16-bit counts (for firmware versions M_01-02-00-000 and above) 0 counts (for firmware version M_01-01-00-000)
Data Type	Integer
See Also	PL.FB , PL.CMD , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3660h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	786	No	8 bit	No	M_01-03-00-000

Description

UNIT.PROTARY sets the position units when the motor type (MOTOR.TYPE) is rotary.

Value	Units
0	counts
1	radians
2	degrees
3	custom units
4	16-bit counts

6.31.8 UNIT.VLINEAR

General Information	
Type	NV Parameter
Description	Sets the linear velocity units.
Units	N/A
Range	0 to 3
Default Value	0
Data Type	Integer
See Also	VL.FB , VL.CMDU , VL.CMD , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	788	No	8 bit	No	M_01-03-00-000

Description

UNIT.VLINEAR sets the units type for the velocity parameters when the motor type (MOTOR.TYPE) is linear.

Type	Description
0	(custom units) per second
1	Micrometers per second
2	Millimeters per second
3	Counts per second

6.31.9 UNIT.VROTARY

General Information	
Type	NV Parameter
Description	Sets the velocity units when the motor type (MOTOR.TYPE) is rotary.
Units	rpm, rps, deg/s, (custom units)/s
Range	0 to 3
Default Value	0 rpm
Data Type	Integer
See Also	VL.FB , VL.CMDU , VL.CMD , MOTOR.TYPE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	365Fh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	790	No	8 bit	No	M_01-03-00-000

Description

UNIT.VROTARY sets the velocity units when the motor type (MOTOR.TYPE) is rotary.

Value	Units
0	rpm
1	rps
2	deg/s
3	(custom units)/s

6.32 VBUS Parameters

This section describes the VBUS parameters.

6.32.1 VBUS.OVFTHRESH

General Information	
Type	R/O Parameter
Description	Reads the over voltage fault level.
Units	Vdc
Range	0 to 900 Vdc
Default Value	N/A
Data Type	Integer
See Also	VBUS.UVFTHRESH
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	794	No	16 bit	No	M_01-03-00-000

Description

VBUS.OVFTHRESH reads the over voltage fault level for the DC bus.

This value is read from the drive EEPROM and varies according to the drive type.

6.32.2 VBUS.OVWTHRESH

General Information	
Type	NV Parameter
Description	Sets voltage level for over voltage warning.
Units	Vdc
Range	0 to 900 Vdc
Default Value	0 Vdc (warning disabled)
Data Type	U16
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	796	No	16 bit	No	M_01-03-00-000

Description

If VBUS.VALUE value exceeds VBUS.OVWTHRESH, then a warning is generated.

6.32.3 VBUS.RMSLIMIT

General Information	
Type	R/O Parameter
Description	Reads the limit for the bus capacitors load.
Units	Vrms
Range	N/A
Default Value	N/A
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	798	No	8 bit	No	M_01-03-00-000

Description

This parameter reads the limit of the bus capacitor load. When the bus capacitor loads exceeds this limit, the drive generates fault F503.

Excessive bus capacitor load may indicate a disconnected main supply phase.

6.32.4 VBUS.UVFTHRESH

General Information	
Type	R/W Parameter
Description	Sets the under voltage fault level.
Units	Vdc
Range	90 to 420 Vdc
Default Value	90 Vdc
Data Type	Integer
See Also	VBUS.OVFTHRESH
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	800	No	16 bit	No	M_01-03-00-000

Description

VBUS.UVFTHRESH sets the undervoltage fault level of the DC bus.

The default value is read from the EEPROM, but can by modified by the user and stored on the NV RAM. This value varies according to drive type.

6.32.5 VBUS.UVMODE

General Information	
Type	N/V Parameter
Description	Indicates undervoltage (UV) mode.
Units	N/A
Range	0 to 1
Default Value	1
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	802	No	8 bit	No	M_01-03-00-000

Description

This parameter indicates undervoltage (UV) mode.

When VBUS.UVMODE - 0, an undervoltage fault is issued whenever the DC bus goes below the undervoltage threshold.

When VBUS.UVMODE = 1, an undervoltage fault is issued whenever the DC bus goes below the

under voltage threshold and the controller attempts to enable the drive (software or hardware enable).

6.32.6 VBUS.UVWTHRESH

General Information	
Type	NV Parameter
Description	Sets voltage level for undervoltage warning.
Units	Vdc
Range	0 to 900 Vdc
Default Value	10 volts above the default value of the under voltage fault threshold (VBUS.UVFTHRESH). The default value of VBUS.UVFTHRESH is hardware dependent.
Data Type	U16
See Also	VBUS.UVFTHRESH
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	804	No	16 bit	No	M_01-03-00-000

Description

If VBUS.VALUE value drops below VBUS.UVWTHRESH, then a warning is generated.

6.32.7 VBUS.VALUE

General Information	
Type	R/O Parameter
Description	Reads DC bus voltage.
Units	Vdc
Range	0 to 900 Vdc
Default Value	N/A
Data Type	Float
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	361Ah/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	806	No	32 bit	No	M_01-03-00-000

Description

VBUS.VALUE reads the DC bus voltage.

6.33 VL Parameters

This section describes the VL parameters.

6.33.1 VL.ARPF1 TO VL.ARPF4

General Information	
Type	R/W Parameter
Description	Sets the natural frequency of the pole (denominator) of anti-resonance (AR) filters 1, 2, 3, and 4; active in opmodes 1 (velocity) and 2 (position) only.
Units	Hz
Range	5 to 5,000 Hz
Default Value	500 Hz
Data Type	Float
See Also	VL.ARPQ1 TO VL.ARPQ4 , VL.ARZF1 TO VL.ARZF4 , Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only.
Start Version	M_01-02-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3406h/1	VL.ARPF1	M_01-02-00-000
	3406h/2	VL.ARPF2	
	3406h/3	VL.ARPF3	
	3406h/4	VL.ARPF4	

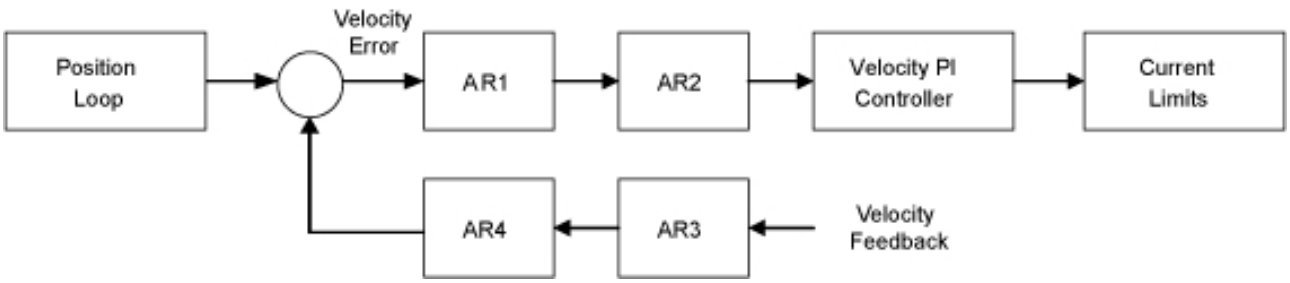
Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	808	VL.ARPF1	No	32 bit	No	M_01-03-00-000
	810	VL.ARPF2				
	812	VL.ARPF3				
	814	VL.ARPF4				

Description

VL.ARPF1 sets the natural frequency of the pole (denominator) of AR filter 1. This value is F_P in the approximate transfer function of the filter:

$$ARx(\mathbf{s}) = [\mathbf{s}^2/(2\pi F_Z)^2 + \mathbf{s}/(Q_Z 2\pi F_Z) + 1]/\ [\mathbf{s}^2/(2\pi F_P)^2 + \mathbf{s}/(Q_P 2\pi F_P) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$\mathbf{s} \approx (1-z^{-1})/t$, where $t = 62.5\ \mu s$

The poles are prewarped to F_P and the zeros are prewarped to F_Z.

6.33.2 VL.ARPQ1 TO VL.ARPQ4

General Information	
Type	R/W Parameter
Description	Sets the Q of the pole (denominator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only.
Units	None
Range	0.2 to 20
Default Value	0.5
Data Type	Float
See Also	VL.ARPF1 TO VL.ARPF4 , VL.ARZF1 TO VL.ARZF4 , VL.ARZQ1 TO VL.ARZQ4
Start Version	M_01-02-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3406h/5	VL.ARPQ1	M_01-02-00-000
	3406h/6	VL.ARPQ2	
	3406h/7	VL.ARPQ3	
	3406h/8	VL.ARPQ4	

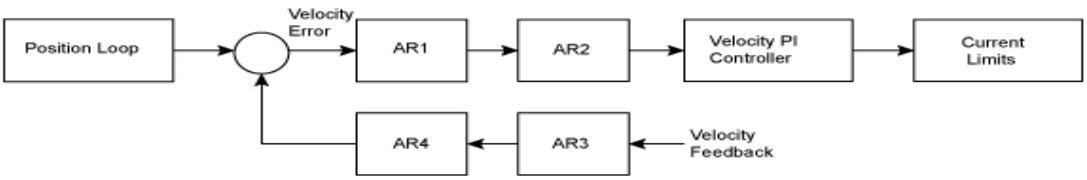
Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	816	VL.ARPQ1	No	32 bit	No	M_01-03-00-000
	818	VL.ARPQ2				
	820	VL.ARPQ3				
	822	VL.ARPQ4				

Description

VL.ARPQ1 sets the Q (quality factor) of the pole (denominator) of AR filter 1. This value is Q_P in the approximate transfer function of the filter:

$$ARx(\mathbf{s}) = [\mathbf{s}^2/(2\pi F_Z)^2 + \mathbf{s}/(Q_Z 2\pi F_Z) + 1] / [\mathbf{s}^2/(2\pi F_P)^2 + \mathbf{s}/(Q_P 2\pi F_P) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$\mathbf{s} \approx (1-z^{-1})/t$, where $t = 62.5\ \mu s$

The poles are prewarped to F_P and the zeros are prewarped to F_Z.

6.33.3 VL.ARTYPE1 TO VL.ARTYPE4

General Information	
Type	NV Parameter
Description	Indicates the method used to calculate BiQuad coefficients; active in opmodes 1 (velocity) and 2 (position) only.
Units	N/A
Range	0
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3405h /1	VL.ARTYPE1	M_01-00-00-000
	3405h /2	VL.ARTYPE2	
	3405h /3	VL.ARTYPE3	
	3405h /4	VL.ARTYPE4	

Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	824	VL.ARTYPE1	No	8 bit	No	M_01-03-00-000
	826	VL.ARTYPE2				
	828	VL.ARTYPE3				
	830	VL.ARTYPE4				

Description

These parameters indicate the method used to calculate the biquad coefficients VL.ARPFx, VL.ARPQx, VL.ARZFx, and VL.ARZQx. A value of 0 indicates that the coefficients are set directly. This parameter has no effect on the filter itself, but is only used to determine the original design parameters. Currently, only the value of 0 is supported.

6.33.4 VL.ARZF1 TO VL.ARZF4

General Information	
Type	R/W Parameter
Description	Sets the natural frequency of the zero (numerator) of anti-resonance (AR)filter 1; active in opmodes 1 (velocity) and 2 (position) only.
Units	Hz
Range	5 to 5,000 Hz
Default Value	500 Hz
Data Type	Float
See Also	VL.ARPF1 TO VL.ARPF4 ,VL.ARPQ1 TO VL.ARPQ4 , VL.ARZQ1 TO VL.ARZQ4
Start Version	M_01-02-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3406h /9	VL.ARZF1	M_01-02-00-000
	3406h /A	VL.ARZF2	
	3406h /B	VL.ARZF3	
	3406h /C	VL.ARZF4	

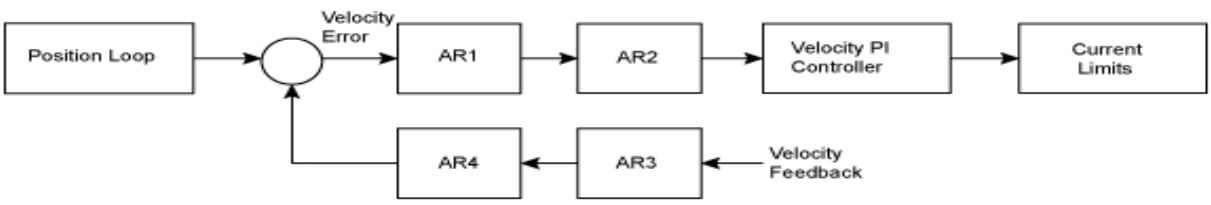
Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	832	VL.ARZF1	No	32 bit	No	M_01-03-00-000
	834	VL.ARZF2				
	836	VL.ARZF3				
	838	VL.ARZF4				

Description

VL.ARZF1 sets the natural frequency of the zero (numerator) of AR filter 1. This value is F_Z in the approximate transfer function of the filter:

$$ARx(\mathbf{s}) = [\mathbf{s}^2/(2\pi F_Z)^2 + \mathbf{s}/(Q_Z 2\pi F_Z) + 1]/\ [\mathbf{s}^2/(2\pi F_P)^2 + \mathbf{s}/(Q_P 2\pi F_P) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$\mathbf{s} \approx (1-z^{-1})/t$, where t = 62.5 µs

The poles are prewarped to F_P and the zeros are prewarped to F_Z.

6.33.5 VL.ARZQ1 TO VL.ARZQ4

General Information	
Type	R/W Parameter
Description	Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only.
Units	N/A
Range	0.1 to 5
Default Value	0.5
Data Type	Float
See Also	VL.ARPF1 TO VL.ARPF4 , VL.ARPQ1 TO VL.ARPQ4 , VL.ARZF1 TO VL.ARZF4
Start Version	M_01-02-00-000

Fieldbus	Index/Subindex		Object Start Version
EtherCAT COE and CANopen	3406h/D	VL.ARZQ1	M_01-02-00-000
	3406h/E	VL.ARZQ2	
	3406h/F	VL.ARZQ3	
	3406h /10	VL.ARZQ4	

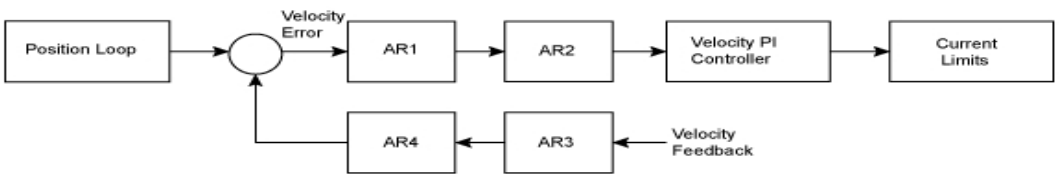
Fieldbus	Index/Subindex		Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	840	VL.ARZQ1	No	32 bit	No	M_01-03-00-000
	842	VL.ARZQ2				
	844	VL.ARZQ3				
	846	VL.ARZQ4				

Description

VL.ARZQ1 sets the Q (quality factor) of the zero (numerator) of AR filter 1. This value is Q_Z in the approximate transfer function of the filter:

$$AR1(\mathbf{s}) = [\mathbf{s}^2/(2\pi F_Z)^2 + \mathbf{s}/(Q_Z 2\pi F_Z) + 1]/\ [\mathbf{s}^2/(2\pi F_P)^2 + \mathbf{s}/(Q_P 2\pi F_P) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3 and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$\mathbf{s} \approx (1-z^{-1})/t$, where t = 62.5 µs.

The poles are prewarped to F_P and the zeros are prewarped to F_Z.

6.33.6 VL.BUSFF

General Information	
Type	R/O Parameter
Description	Displays the velocity loop feedforward value injected by the field-bus; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	0.0 to VL.LIMITP
Default Value	0.0
Data Type	Float
See Also	VL.FF , VL.KBUSFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	60B1h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	848	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

This parameter displays the velocity loop feedforward value injected by the fieldbus.

6.33.7 VL.CMD

General Information	
Type	R/O Parameter
Description	Reads the actual velocity command; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float
See Also	VL.FB , VL.CMDU , VL.LIMITP , VL.LIMITN
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	606Bh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	850	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.CMD returns the actual velocity command as it is received in the velocity loop entry after all velocity limits (such as VL.LIMITN and VL.LIMITP). See velocity loop design diagram for more details.

6.33.8 VL.CMDU

General Information	
Type	R/W Parameter
Description	Sets the user velocity command; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s
Range	Rotary -15,000.000 to 15,000.000 rpm -250.000 to 250.000 rps -90000.000 to 90000.000 deg/s -1250.000 to 1250.000 custom units/s -1570.796 to 1570.796 rad/s Linear -1,073,741,824,000.000 to 1,073,741,824,000.000 counts/s -8,000.000 to 8,000.000 mm/s -8,000,000.000 to 8,000,000.000 µm/s -1,250.000 to 1,250.000 custom units/s
Default Value	0
Data Type	Float
See Also	VL.FB , VL.CMD , DRV.OPMODE , DRV.CMDSOURCE , VL.LIMITN , VL.LIMITP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	60FFh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	852	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.CMDU sets the user velocity command.

When DRV.OPMODE is set to 1 (velocity loop) and DRV.CMDSOURCE is set to 0 (TCP/IP channel), then setting this value when the drive is enabled will cause the drive to rotate at the required velocity.

6.33.9 VL.ERR

General Information	
Type	R/O Parameter
Description	Sets the velocity error; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float
See Also	VL.CMD , VL.FB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3407h/4	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	854	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.ERR sets the velocity error. It is calculated in the velocity loop as the difference between VL.CMD and VL.FB .

6.33.10 VL.FB

General Information	
Type	R/O Parameter
Description	Reads the velocity feedback; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float
See Also	VL.CMDU
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3618h/0 606Ch/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	856	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.FB returns the velocity feedback as it is received in the velocity loop, after passing through Filter 3 and Filter 4.

6.33.11 VL.FBFILTER

General Information	
Type	R/O Parameter
Description	Filters VL.FB value; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float
See Also	VL.FB
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3407h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	858	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

This parameter returns the same value as VL.FB , filtered through a 10 Hz filter.

6.33.12 VL.FBSOURCE

General Information	
Type	NV Parameter
Description	Sets feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	PL.FBSOURCE
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	860	No	8 bit	No	M_01-03-00-000

Description

This parameter determines the feedback source to be used by the velocity loop. A value of 0 selects the primary feedback, 1 selects the secondary feedback.

6.33.13 VL.FBUNFILTERED

General Information	
Type	R/O Parameter
Description	Reads the velocity feedback.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR, UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	N/A
Default Value	N/A
Data Type	Float
See Also	VL.FB , VL.FBFILTER
Start Version	M_01-03-06-000

Description

VL.FBUNFILTERED reads the raw velocity feedback before any filters affect the value of this feedback.

6.33.14 VL.FF

General Information	
Type	R/O Parameter
Description	Displays the velocity loop overall feedforward value; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, µm/s, (custom units)/s
Range	0 to VL.LIMITP
Default Value	0
Data Type	Float
See Also	VL.KBUSFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	862	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

This parameter displays the velocity loop overall feedforward value.

6.33.15 VL.GENMODE

General Information	
Type	NV Parameter
Description	Selects mode of velocity generation (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	Integer
See Also	N/A
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	864	No	16 bit	No	M_01-03-00-000

Description

This parameter is used to select the velocity generator mode.

Mode	Description
0	d/dt mode: The derivative of the mechanical angle of the drive is fed to a first order low pass.
1	Luenberger Observer mode

6.33.16 VL.KBUSFF

General Information	
Type	R/W Parameter
Description	Sets the velocity loop acceleration feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only.
Units	NA
Range	0.0 to 2.0
Default Value	0.0
Data Type	Float
See Also	VL.BUSFF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3407h/3	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	866	No	32 bit	No	M_01-03-00-000

Description

This parameter sets the gain for the acceleration feedforward (a scaled second derivative of the position command is added to the velocity command value).

The nominal feedforward value can be multiplied by this gain value.

This will have affect only when using position mode (DRV.OPMODE = 2).

6.33.17 VL.KI

General Information	
Type	NV Parameter
Description	Sets the velocity loop integral gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only.
Units	Hz
Range	0 to 1,000 Hz
Default Value	160 Hz
Data Type	Float
See Also	VL.KP
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	354Dh/0	M_01-00-00-000

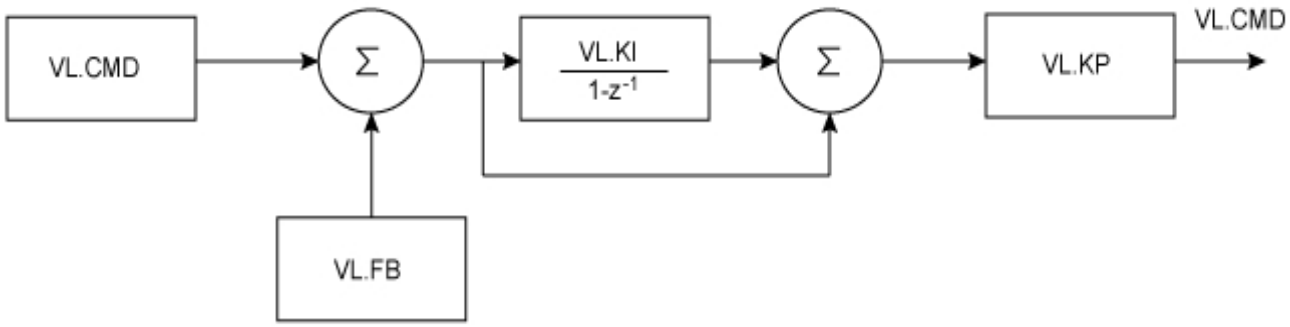
Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	868	No	32 bit	No	M_01-03-00-000

Description

VL.KI sets the integral gain of the velocity loop.

A factor of 2π is included in the time calculation, therefore a PI velocity loop with a constant error of 1 rps in which VL.KI is set to 160 and VL.KP is set to 1, will take (1000/160)*2π ms to increase the integral gain to 1. Therefore, the total gain is 2 at this time (see velocity loop structure below).

Velocity Loop Structure



6.33.18 VL.KP

General Information	
Type	NV Parameter
Description	Sets velocity loop proportional gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only.
Units	A/(rad/sec)
Range	0.001 to 2,147,483.008
Default Value	1
Data Type	Float
See Also	VL.KI
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3548h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	872	No	32 bit	No	M_01-03-00-000

Description

VL.KP sets the proportional gain of the velocity loop.

The idealized velocity loop bandwidth in Hz is:

Rotary motor:

Bandwidth (Hz) = VL.KP *K_t/ (2π *J_m)

Where:

K_t= motor torque constant, in units of Nm/Arms

J_m = motor inertia, in units of kg*m²

Linear motor:

Bandwidth (Hz) = VL.KP * K_t / (Motor Pitch (mm) * J_m)

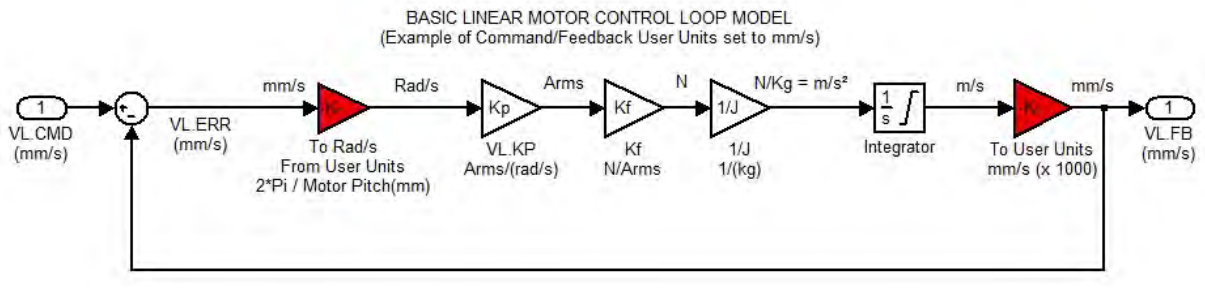
Where:

K_t= motor torque constant, in units of Nm/Arms

J_m = motor inertia, in units of kg

The drive uses the same control loop for both linear and rotary motors. VL.KP units are in Arms /(rad/s). If you want to tune in units of Arms/(mm/s), then you must manually convert the units.

The diagram below shows how linear motors are implemented at the control loop level.



The red blocks are automatically handled at the drive level.

2π radians is the linear equivalent of one full mechanical revolution of a rotary motor - and is equal to the MOTOR.PITCH of a linear motor.

Example

To convert VL.KP = 0.320 Arms/(rad/s) to Arms/(mm/s), where MOTOR.PITCH is 32 mm:

```
VL.KP = 0.320 Arm /rad/s * (2π rad / 32mm MOTOR.PITCH)
VL.KP = 0.32 * 2π / 32 = 0.063 Arms / (mm/s)
```

6.33.19 VL.KVFF

General Information	
Type	R/W Parameter
Description	Sets the velocity loop velocity feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only.
Units	NA
Range	0.0 to 2.0
Default Value	0.0
Data Type	Float
See Also	VL.FF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3407h/2 354Bh/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	874	No	32 bit	No	M_01-03-00-000

Description

This parameter sets the gain for the velocity feedforward (a scaled derivative of the position command is added to the velocity command value). The nominal feedforward value can be multiplied by this gain value.

This parameter is only used in the position mode (DRV.OPMODE = 2).

6.33.20 VL.LIMITN

General Information	
Type	NV Parameter
Description	Sets the velocity lower limit; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s
Range	Rotary: -15,000.000 to 0.000 rpm -250.000 to 0.000 rps -90,000.000 to 0.000 deg/s -1,250.000 to 0.000 custom units/s -1570.796 to 0.000 rad/s Linear: -1,073,741,824,000.000 to 0.000 counts/s -250.000*MOTOR.PITCH to 0.000 mm/s -250,000.000*MOTOR.PITCH to 0.000 µm/sec -1,250.000 to 0.000 custom units/s
Default Value	Rotary: -3,000.000 rpm -50.000 rps -18,000.002 deg/s -250.000 (custom units)/s -314.159 rad/s Linear: -0.050 counts/s -50*MOTOR.PITCH mm/s -50,000.004*MOTOR.PITCH µm/sec -250.000 custom units/s
Data Type	Float
See Also	VL.LIMITP , VL.CMD
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3623h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	876	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.LIMITN sets the velocity command negative limit.

If the input to the velocity loop is lower than VL.LIMITN, then the actual velocity command VL.CMD is limited by the value of VL.LIMITN.

6.33.21 VL.LIMITP

General Information	
Type	NV Parameter
Description	Sets the velocity high limit; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s
Range	Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/sec 0.000 to 250,000.000*MOTOR.PITCH µm/s 0.000 to 1,250.000 custom units/s
Default Value	Rotary: 3,000.000 rpm 50.000 rps 18,000.002 deg/s 250.000 (custom units)/s 314.159 rad/s Linear: 0.050 counts/s 50.000*MOTOR.PITCH mm/sec 50,000.004*MOTOR.PITCH µm/sec 250.000 custom units/s
Data Type	Float
See Also	VL.LIMITN , VL.CMD
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3622h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	878	Yes	low 32 bit word	No	M_01-03-00-000

Description

VL.LIMITP sets the velocity command positive limit.

If the input to the velocity loop is higher than VL.LIMITP, then the actual velocity command VL.CMD is limited by the value of VL.LIMITP.

6.33.22 VL.LMJR

General Information	
Type	NV Parameter
Description	Sets the ratio of the estimated load moment of inertia relative to the motor moment of inertia; active in opmodes 1 (velocity) and 2 (position) only.
Units	NA
Range	0 to 100.0
Default Value	0
Data Type	Float
See Also	IL.FF
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	880	No	32 bit	No	M_01-03-00-000

Description

This parameter is used in the internal calculation of the current loop acceleration feed forward gain value.

6.33.23 VL.THRESH

General Information	
Type	NV Parameter
Description	Sets the over speed fault value; active in opmodes 1 (velocity) and 2 (position) only.
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s
Range	Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCHMOTOR.PITCH µm/s 0.000 to 1,250.000 custom units/s
Default Value	Rotary: 3,600 rpm 60 rps 21,600.000 deg/s 300.000 custom units/s 376.991 rad/s Linear: 0.060 counts/s 60.000*MOTOR.PITCH mm/s 60,000.04*MOTOR.PITCHMOTOR.PITCH µm/s 300.000 custom units/s
Data Type	Float
See Also	VL.CMD , VL.CMDU
Start Version	M_01-00-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3627h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	888	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

VL.THRESH sets the threshold for the velocity over which an over speed fault is generated.

The value is considered as an absolute value, hence it applies for both negative and positive velocities.

Example

VL.THRESH is set to 600 rpm. A velocity (VL.FB) of 700 rpm will generate an over speed fault.

6.34 VM Parameters

This section describes the VM parameters.

6.34.1 VM.AUTOSTART

General Information	
Type	R/W
Description	VM.AUTOSTART specifies whether or not the program in the AKD BASIC starts executing automatically when AC power is applied.
Units	none
Range	0 to 1
Default Value	0
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1152	No	32 bit	No	M_01-05-11-000

Description

VM.AUTOSTART specifies whether or not the program in the AKD BASIC starts executing automatically when AC power is applied.

- 0 = Program does not start automatically
- 1 = Program starts automatically

Set VM.AUTOSTART to 0 or 1 in the terminal in WorkBench and execute an NVSave.

6.34.2 VM.ERR

General Information	
Type	R/W
Description	Indicates what caused the most recent Runtime Error.
Units	TBD
Range	TBD
Default Value	TBD
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1162	No	32 bit	No	M_01-05-11-000

Description

Runtime errors are caused by the program running on the AKD BASIC trying to do something that is not allowed. For example, runtime errors occur when you attempt to write a value that is too high or too low to a particular variable. We try to catch as many errors as possible when the program is compiled, but some errors are only detected when the program is running.

Determine the particular problem causing Runtime Error (F4 Fault) by looking at the value of the VM.ERR variable. Use the Variables Window to find the value of VM.ERR.

The table below shows what each value of VM.ERR means.

Value of VM_ERR	Error Caused by
1	Division by zero in arithmetic
2	Stack is full
3-5	(not used)
6	Out of Memory
7-10	(not used)
11	Attempt to use Feature not available in this firmware
12	Internal firmware error
13	Invalid Parameter ID Number
14	Attempt to write to a Read-Only Variable
15	DSP Read Error
16	DSP Write Error
17	DSP Command Error
18-21	(not used)
22	No interrupt handler defined
23	(not used)
24	PACLAN Transmit Error
25	PACLAN Response Timeout
26	PACLAN Response Error
27	Interrupt error
28	Maximum String Length Exceeded
29	String Overflow

Value of VM_ERR	Error Caused by
30	Array Index Bounds Error
31	Invalid Axis in PACLAN Message
32	No LAN Interrupt Handler
33	LAN Interrupt Queue is full
34	LAN Interrupt is not available
35	LAN Interrupt: Destination is busy
36	ModBus: Attempt to do nested master functions
37	ModBus: Attempt to use master without setting RuntimeProtocol
38	ModBus: Illegal Slave Address (255)
39	AB DF1: Invalid PLC Address (0-255)
40	AB DF1: Invalid PLC File Number Specified
41	AB DF1: Invalid PLC Element Number Specified
42	AB DF1: too many unresolved messages outstanding
43	AB DF1: Attempt to use AB DF1 without setting RunTimeProtocol
44	AB DF1: Transmit queue overflow
45	$DeclareCam: Invalid Cam Number specified
46	$DeclareCam: Too many points specified.
47	CAM.CREATE: Tried to create a new cam before finished creating the first one.
48	CAM.CREATE: Tried to create cam without declaring it.
49	CAM.ADDPOINT: Tried to add more points than declared.
50	CAM.ADDPOINT: Starting Master position is non-zero.
51	CAM.ADDPOINT: Used CAM.ADDPOINT outside a CAM.CREATE block.
52	CAM.CREATE: EndList without Create
53	CAM.CREATE: Tried to create a cam with less than three points.
54	CAM.ADDPOINT: Used the same master position for two points or master position was negative
55	CAM.CREATE: Tried to create the CAM.ACTIVATE.
56	CAM.ACTIVATE: Tried to activate a cam that was not created.
57	CAM.ACTIVATE: Tried to activate a cam while it is being created.
58	CAM.ACTIVATE: Tried to activate a cam while MOVE.RUNSPEED =0.
59	CAM.ACTIVATE: Tried to activate a cam with master position outside the cam table.

6.34.3 VM.INTRTIMER

General Information	
Type	R/W
Description	Sets a number of milliseconds before INTR.TIMER executes after it is called.
Units	Milliseconds
Range	100 to 1,000,000,000
Default Value	
Data Type	Integer

Description

Sets a number of milliseconds before INTR.TIMER executes after it is called. The real-time timer is reset each time INTR.TIMER = 1. VM.INTRTIMER can be changed in the Timer Interrupt Service Routine. To avoid run-time problems, the user needs to keep the Timer ISR short (must be less than VM.INTRTIMER).

Related Topics

Interrupt {Source}

6.34.4 VM.RESTART

General Information	
Type	Command
Description	Restart AKD BASIC virtual machine.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1154	No	Command	No	M_01-05-11-000

Description

Restarts the AKD BASIC virtual machine if it has been stopped.

Note: This keyword is executed when the continue icon is pressed in the Program view.

Related Topics

VM.START | VM.STOP

6.35 WHEN Parameters

This section describes the WHEN parameters.

6.35.1 When

General Information	
Type	Statement
Description	The WHEN statement is used for very fast response to certain input conditions.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
Start Version	TBD

Description

The WHEN statement is used for very fast response to certain input conditions. Upon encountering and executing the WHEN statement, program execution waits until the specified condition is satisfied. When the condition is satisfied, the when-action is executed immediately and the program continues at the next line after the WHEN statement. The syntax is as follows:

```
When when-condition, when-action
```

The following tables lists the possible when-conditions and when actions

When-conditions
PL.FB < value
PL.FB > value
PL.CMD < value
PL.CMD > value
DRV.HANDWHEEL < value
DRV.HANDWHEEL > value
DRV.TIME > value
DIN1.STATE = 0 or 1
DIN2.STATE = 0 or 1
DIN3.STATE = 0 or 1
DIN4.STATE = 0 or 1
DIN5.STATE = 0 or 1
DIN6.STATE = 0 or 1
DIN7.STATE = 0 or 1
FB3.P < value
FB3.P > value
MOVE.MOVING = 0 or 1
MOVE.INPOSITION = 0 or 1

When-actions
Continue

When-actions
DOUT1.STATEU = 0 or 1
DOUT2.STATEU = 0 or 1
EGEAR.RATIO = value
MOVE.ABORT
MOVE.GOABS
MOVE.GOREL
MOVE.GOVEL
MOVE.GOABSREG
MOVE.GORELREG
MOVE.GOUPDATE
MOVE.GOHOME

Interrupts are active and will be serviced during the execution of a WHEN statement. The execution of an interrupt service routine will not affect how quickly the when-action is executed after the when-condition is satisfied.

The When condition is checked every 1 millisecond. At the instant (within 1 msec) that the when-condition is satisfied, the values of the following variables are strobed into special When variables:

Variable	When variable
DRV.HANDWHEEL	WHEN.DRVHANDWHEEL
DRV.TIME	WHEN.DRVTIME
FB1.MECHPOS	WHEN.FB1MECHPOS
PL.CMD	WHEN.PLCMD
PL.FB	WHEN.PLFB

Example

```
When                DIN1.STATE = 1, Continue
        ...
When                DRV.HANDWHEEL > 10000,
DOUT1.STATEU = 1
        ...
When                DRV.TIME > 5.6, EGEAR.RATIO
= -2.2
```

Related Topics

WHEN.DRVHANDWHEEL | WHEN.FB1MECHPOS | WHEN.DRVTIME
WHEN.PLCMD | WHEN.PLFB

6.35.2 WHEN.DRVHANDWHEEL

General Information	
Type	R/O
Description	Records the value of DRV.HANDWHEEL when the when-condition is satisfied.
Units	1/4,294,967,296 rev
Range	0 to 4,294,967,295 rev
Default Value	0 rev
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1170	No	32 bit	No	M_01-05-11-000

Description

Records the value of DRV.HANDWHEEL when the when-condition is satisfied. The when-condition is checked once per millisecond.

Related Topics

DRV.HANDWHEEL | WHEN.ENCPOS | When

6.35.3 WHEN.DRVTIME

General Information	
Type	R/O
Description	Records the value of Time when the when-condition is satisfied.
Units	Seconds
Range	0 - 2,147,483 (~24.8 days)
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Object Start Version
Modbus	1172	M_01-05-11-000

Description

Records the value of DRV.RUNTIME when the when-condition is satisfied. The when-condition is checked once per millisecond.

Related Topics

DRV.RUNTIME | WHEN.TIME | When

6.35.4 WHEN.FB1MECHPOS

General Information	
Type	R/O
Description	Records the value of FB1.MECHPOS when the when-condition is satisfied.
Units	Position units
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1164	No	32 bit	No	M_01-05-11-000

Description

Records the value of FB1.MECHPOS when the when-condition is satisfied. The when-condition is checked once per millisecond.

Related Topics

FB1.MECHPOS | WHEN.RESPOS | When

6.35.5 WHEN.PLCMD

General Information	
Type	R/O
Description	Records the value of PL.CMD when the when-condition is satisfied.
Units	Position units
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1174	Yes	64 bit	No	M_01-05-11-000

Description

Records the value of PL.CMD when the when-condition is satisfied. The when-condition is checked once per millisecond.

Related Topics

PL.CMD | WHEN.POSCOMMAND | When

6.35.6 WHEN.PLFB

General Information	
Type	R/O
Description	Records the value of Position when the when-condition is satisfied.
Units	Depends on UNIT.PROTARY or UNIT.PLINEAR UNIT.A-CCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, µm, (custom units), 16-bit counts
Range	N/A
Default Value	N/A
Data Type	Integer

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	1178	Yes	64 bit	Yes	M_01-05-11-000

Description

Records the value of Position when the when-condition is satisfied. The when-condition is checked once per millisecond.

Related Topics

PL.FB | When

6.36 WS Parameters

This section describes the WS parameters.

6.36.1 WS.ARM

General Information	
Type	Command
Description	Sets wake and shake to start at the next drive enable.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3494h/6	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	890	No	Command	No	M_01-03-00-000

Description

This command sets wake and shake to start at the next drive enable. Feedback type is not relevant for this command. If WS.STATE is 0 and the drive is disabled, then WS.STATE will change to 1 after issuing WS.ARM. With this command, wake and shake can be repeated if desired.

6.36.2 WS.DISARM

General Information	
Type	Command
Description	Cancels ARM requests and resets wake and shake to the IDLE state.
Units	N/A
Range	N/A
Default Value	N/A
Data Type	N/A
See Also	N/A
Start Version	M_01-04-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	N/A	
Modbus	N/A	

Description

This command disables wake and shake immediately. Feedback type is not relevant for this command. If WS.ARM has been issued, the request to execute the wake and shake algorithm at the next enable is cancelled. WS.STATE is set to IDLE.

6.36.3 WS.DISTMAX

General Information	
Type	R/W Parameter
Description	Sets maximum movement allowed for wake and shake.
Units	deg (position units)
Range	0 to 90 deg
Default Value	15 deg
Data Type	S64
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3494h/2	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	892	Yes	64 bit	Yes	M_01-03-00-000

Description

This parameter sets the maximum movement that is allowed for finding commutation. If this value is too small, F475, "Wake and Shake. Too much movement", may occur before wake and shake is finished. The bigger this value, the more movement is allowed for wake and shake. This value is application dependent.

6.36.4 WS.DISTMIN

General Information	
Type	R/W Parameter
Description	Sets the minimum movement required for wake and shake.
Units	Actual position units
Range	0 to 90 deg
Default Value	1 deg
Data Type	S64
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	36D1h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	896	Yes	64 bit	Yes	M_01-03-00-000

Description

This parameter sets the minimum movement that is required for commutation finding. If this value is too small, the the commutation finding might fail if too little current is used. The larger this value, the more movement is needed in order to avoid F473: "Wake and Shake: Too little movement".

6.36.5 WS.IMAX

General Information	
Type	R/W Parameter
Description	Sets maximum current used for wake and shake.
Units	Arms
Range	0 to (lower value of MOTOR.IPEAK and DRV.IPEAK) Arms
Default Value	(half of maximum) Arms
Data Type	U16
See Also	MOTOR.IPEAK , DRV.IPEAK
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3494h/1	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	900	No	32 bit	Yes	M_01-03-00-000

Description

This parameter defines the maximum current used for wake and shake. If the selected current is too low, the minimum required movement may not occur. If the selected current is too high, the movement may be too fast (overspeed) or too large (over maximum movement).

The maximum of this parameter is the lower value of MOTOR.IPEAK and DRV.IPEAK. The default value of this parameter is the half of its maximum. This value depends on the specific application.

6.36.6 WS.MODE

General Information	
Type	R/W Parameter
Description	Sets the method used for wake and shake.
Units	N/A
Range	0 to 1
Default Value	0
Data Type	U8
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	902	No	8 bit	No	M_01-03-00-000

Description

This parameter sets the method used for finding commutation.

0 = Standard wake and shake

Two iterations are used to find the correct angle in this mode. Coarse (current mode) and fine (velocity mode) iterations are done in a loop (WS.NUMLOOPS times). The average angle of all loops is calculated and used.

1 = Commutation alignment by fixed commutation vector (Zero Method)

The motor poles are set to 0, current mode is activated, and WS.IMAX is applied. The angle in which the motor settles is used for commutation. Other settings are restored (such as motor poles and operation mode).

6.36.7 WS.NUMLOOPS

General Information	
Type	R/W Parameter
Description	Sets the number of repetitions for wake and shake.
Units	counts
Range	0 to 20 counts
Default Value	5 counts
Data Type	U8
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	36E2h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	904	No	8 bit	No	M_01-03-00-000

Description

This parameter sets the maximum number of wake and shake repetitions. MOTOR.PHASE is calculated as mean value of all wake and shake repetitions.

6.36.8 WS.STATE

General Information	
Type	R/O Parameter
Description	Reads wake and shake status
Units	N/A
Range	N/A
Default Value	Only valid before the first enable occurs. 11 - for feedback types that do not require wake and shake 1 - for feedback types that require wake and shake
Data Type	U8
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3494h/5	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	906	No	8 bit	No	M_01-03-00-000

Description

WS switches different current vectors and records position feedback in order to establish commutation alignment.

WS.STATE 0 = wake and shake successful (DONE).

WS.STATE 1 = wake and shake configured and will be done at next enable (ARMED).

WS.STATE 2 = wake and shake running. (ACTIVE)

WS.STATE 10 = error occurred during wake and shake (ERROR).

WS.STATE 11 = wake and shake not required (IDLE).

6.36.9 WS.T

General Information	
Type	R/W Parameter
Description	Sets wake and shake current-vector appliance time
Units	ms
Range	1 to 200 ms
Default Value	2 ms
Data Type	U8
See Also	WS.IMAX , WS.DISTMAX
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	36D0h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	908	No	16 bit	No	M_01-03-00-000

Description

This parameter defines the duration for each different current-vector while the coarse angle calculation. The move distance is proportional to the WS.T and WS.IMAX value.

6.36.10 WS.TDELAY1

General Information	
Type	NV Parameter
Description	Delay for wake and shake timing
Units	ms
Range	0 to 200 ms
Default Value	5 ms
Data Type	U8
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3683h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	910	No	16 bit	No	M_01-03-00-000

Description

WS.TDELAY1 defines the delay time of the wake and shake function. This time is a delay time between the switching of different current vectors during the wake and shake procedure. This time should be increased in the case of movement interferences between single current vectors.

6.36.11 WS.TDELAY2

General Information	
Type	NV Parameter
Description	Sets the delay for wake and shake timing.
Units	ms
Range	0 to 200 ms
Default Value	50 ms
Data Type	U8
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3685h/0	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	912	No	16 bit	No	M_01-03-00-000

Description

WS.TDELAY2 defines the delay between switching from coarse angle calculation to fine angle calculation during the wake and shake procedure. This time should be increased in the case of interferences between the coarse calculation done in current mode and the fine calculation done in velocity mode. Choosing too large a value increases the wake and shake duration.

6.36.12 WS.TDELAY3

General Information	
Type	NV Parameter
Description	Sets the delay for wake and shake between loops in mode 0.
Units	ms
Range	0 to 2,000 ms
Default Value	100 ms
Data Type	U16
See Also	N/A
Start Version	M_01-01-00-102, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CAN-open	3494h/3	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	914`	No	16 bit	No	M_01-03-00-000

Description

WS.TDELAY3 defines the delay between complete loops in mode 0 only. Decreasing this value makes the wake and shake procedure faster, but may lead to problems if the motor moves too long. Increasing this value will make wake and shake significantly longer.

6.36.13 WS.VTHRESH

General Information	
Type	NV Parameter
Description	Defines the maximum allowed velocity for Wake & Shake
Units	Depends on UNIT.VROTARY or UNIT.VLINEAR UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s
Range	Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 degree/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 µm/s 0.000 to 1,250.000 custom units/s
Default Value	100 rpm
Data Type	U16
See Also	N/A
Start Version	M_01-01-00-101, M_01-02-00-000

Fieldbus	Index/Subindex	Object Start Version
EtherCAT COE and CANopen	3494h/4	M_01-00-00-000

Fieldbus	Index/Subindex	Is 64 bit?	Attributes	Signed?	Object Start Version
Modbus	916	Yes	low 32 bit word	Yes	M_01-03-00-000

Description

This parameter defines the maximum allowed velocity that occurs while commutation finding is active. This supervision runs in real time, but only while wake and shake is active (WS.STATE 2 or greater, for Mode 0). If at any time while wake and shake is running a velocity higher than this value is detected, fault F478 is generated. Setting WS.VTHRESH to zero disables this feature. For Mode 1, WS.VTHRESH is only used after the initial phase-finding.

This page intentionally left blank.

About Kollmorgen

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For assistance with your application needs, visit www.kollmorgen.com or contact us at:

North America
KOLLMORGEN
203A West Rock Road
Radford, VA 24141 USA

Internet	www.kollmorgen.com
E-Mail	support@kollmorgen.com
Tel.:	+1 - 540 - 633 - 3545
Fax:	+1 - 540 - 639 - 4162

Europe
KOLLMORGEN Europe GmbH
Pempelfurtstraße 1
40880 Ratingen, Germany

Internet	www.kollmorgen.com
E-Mail	technik@kollmorgen.com
Tel.:	+49 - 2102 - 9394 - 0
Fax:	+49 - 2102 - 9394 - 3155

Asia
KOLLMORGEN
Rm 2205, Scitech Tower, China
22 Jianguomen Wai Street

Internet	www.kollmorgen.com
E-Mail	sales.asia@kollmorgen.com
Tel.:	+86 - 400 666 1802
Fax:	+86 - 10 6515 0263