

Overview

In the AKD there are multiple lists of parameters with various starting Modbus addresses.

The given ranges are:

1. Modbus Parameter Table (starting address 0)
2. Modbus 64-bit Parameters to 32-bit mapping (starting address 2000)
3. Modbus Dynamically Mapped Parameters (starting address 8192)
4. AKD BASIC Modbus variables (starting address 5000)
5. Modbus Errors Stack (starting address 4354)

The screenshot shows the Kollmorgen WorkBench Help interface. The left sidebar contains a navigation menu with 'Modbus Parameter Table' selected. The main content area displays the 'Modbus Parameter Table' with the following data:

Parameter	Modbus Register Address	Is 64-bit?	Attributes
AIN.CUTOFF	0		32-bit
AIN.DEADBAND	2		16-bit
AIN.ISCALE	4		32-bit
AIN.OFFSET	6		16-bit, signed
AIN.PSCALE	8	Yes	64-bit, signed

The screenshot shows the Kollmorgen WorkBench Help interface. The left sidebar contains a navigation menu with 'Modbus 64-bit Parameters to 32-bit Mapping' selected. The main content area displays the 'Modbus 64-bit Parameters to 32-bit Mapping' table with the following data:

Parameter	Address	64-bit	Attributes
AIN.PSCALE_32	2000	Yes	low 32-bit word, signed
AOUT.PSCALE_32	2002	Yes	low 32-bit word
AOUT.VALUE_32	2004	Yes	low 32-bit word, signed
AOUT.VALUEU_32	2006	Yes	low 32-bit word, signed
CAP0.PLFB_32	2008	Yes	low 32-bit word, signed
CAP1.PLFB_32	2010	Yes	low 32-bit word, signed

Kollmorgen WorkBench Help

KOLLMORGEN®

Contents Index

- Welcome Page
- Workbench User Manual
- AKD Parameter and Command Reference Guide
- AKD BASIC Programming Manual
 - AKD Fieldbus Manuals
 - Fieldbus Manuals
 - Modbus Manual
 - CANopen Object Dictionary
 - SynqNet Supported Parameters
 - Modbus Manual
 - Modbus Dynamic Mapping
 - Modbus Parameter Table
 - Modbus 64-bit Parameters to 32-bit Mapping
 - Modbus Errors
 - Modbus Dynamic Mapping
 - Modbus Parameter Table

You are here: AKD Fieldbus Manuals > [Fieldbus Manuals](#) > [Modbus Manual](#) > Modbus Dynamic Map

Modbus Dynamic Mapping

Modbus dynamic mapping allows you to map any of the fixed register addresses to a new register address becomes possible.

In general, all parameters are mapped as 32-bit values and occupy at least two Modbus registers. Parameters are mapped as 32-bit value (two registers) starting at register address 2000 (see [Modbus 64-bit Parameter](#))

Configuring Dynamic Mapping

The start address for dynamically mapped registers is 8192 (0x2000).

Mapping works as follows:

1. Enable dynamic mapping by writing a 1 (as a 32-bit value) to register address 4096 (0x1000). This is done in "run" mode, in which you can use the mapped parameter.
2. The mapping is now register oriented. Next, write the valid fixed register address of the parameter to map the corresponding two registers.

Example

Contents Index

- Welcome Page
- Workbench User Manual
- AKD Parameter and Command Reference Guide
- AKD BASIC Programming Manual
 - Using the Program View
 - Unsupported Parameters
 - AKD BASIC Language
 - AKD BASIC Program Structure
 - Program Memory Space
 - Program Sections
 - Main Program Subroutines Functions Interrupt Handling
 - Language Definition
 - Statements
 - Built in Functions
 - Expressions
 - Function Invocation
 - Access and Function Parameter Lists

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, see [Modbus Dynamic Mapping](#)

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
Dim flt1 as float
Dim long1 as long
MBInfo
$MMap32(5001, int2)
$MMap64(5003, long1)
$MMapfloat(5007, flt1)
End

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

KOLLMORGEN®

Contents Index

- Welcome Page
- Workbench User Manual
- AKD Parameter and Command Reference Guide
- AKD BASIC Programming Manual
- AKD Fieldbus Manuals
 - Fieldbus Manuals
 - Modbus Manual
 - CANopen Object Dictionary
 - SynqNet Supported Parameters
 - Modbus Manual
 - Modbus Dynamic Mapping
 - Modbus Parameter Table
 - Modbus 64-bit Parameters to 32-bit Mapping
 - Modbus Errors**
 - Modbus Dynamic Mapping
 - Modbus Parameter Table

Access Through Modbus

In Modbus the error list starts at register address 0x1102 (4354).

Modbus Address	Description
4354	Error count
4355	Error 1 : Register Address
4356	Error 1: Error Code
4357	Error 2 : Register Address
4358	Error 2 : Error Code
...	...
4603	Error 125 : Register Address
4604	Error 125 : Error Code

Modbus Error Descriptions

Error Code	Descriptions
0001	Unknown Fault!

The AKD supports Modbus Function Codes 03 (Read Holding Registers) and 16 (Write Multiple Registers).

What is Dynamic Mapping?

- Dynamic Mapping allows the user to take parameters from different (and non-consecutive) ranges and order and put them in consecutive order.

For example (The following selected parameters were arbitrary for demonstrational purposes):

Reads: Current Position

PL.FB_32	2072	Yes	low 32-bit word, signed
----------	------	-----	-------------------------

Current Velocity

VL.FB	856	Yes	low 32-bit word, signed
-------	-----	-----	-------------------------

Write: Home.Dist (offset move on homing)

HOME.DIST_32	2048	Yes	low 32-bit word, signed
--------------	------	-----	-------------------------

Home.Move (start a move)

HOME.MOVE	408		Command
-----------	-----	--	---------

Following Error Threshold

PL.ERRFTHRESH_32	2068	Yes	low 32-bit word
------------------	------	-----	-----------------

If Dynamic Mapping is not used then the above would require 2 individual reads and 3 individual writes. This is more taxing on the communications and the Ethernet port of the drive. If your list of parameters is more (i.e. 10 reads and 10 writes, etc.) eventually you may start seeing drive warnings of "Modbus Rate Too High". If Dynamic Mapping is used then it is possible to do one read for the 10 read parameters and 1 write for the 10 write parameters. This economizes your communications.

Besides communication efficiency another motivation for dynamic mapping is in the case where the Modbus Master's addressing and the drive's address is different in regards to high word/low word. By using Dynamic Mapping it is possible to do the word swapping in the drive instead of in the master which may clean up some of the programming code.

Using the example parameters above:

Reads

Parameter Description	Static Address	Dynamic Address
PL.FB	2072	8192
	2073	8193
VL.FB	856	8194
	857	8195

Writes

Parameter Description	Static Address	Dynamic Address
HOME.DIST	2048	8200
	2049	8201
HOME.MOVE	408	8202
	409	8203
PL.ERRFTHRESH_32	2068	8204
	2069	8205

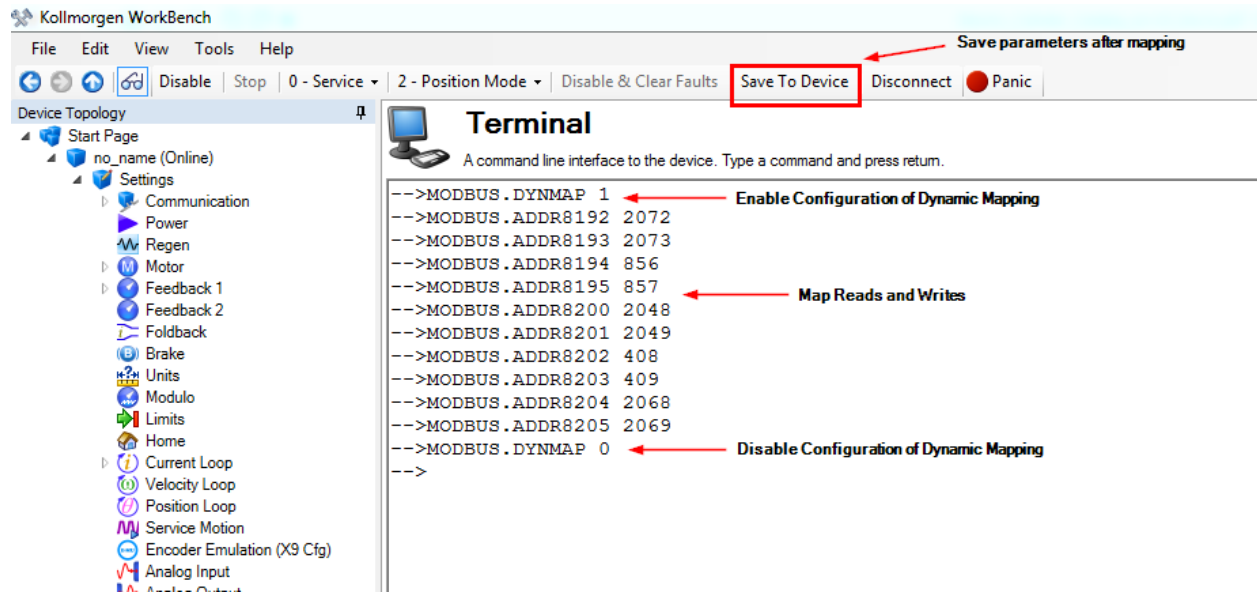
To implement use Workbench Terminal as follows:

Type “MODBUS.DYNMAP 1” to enable configuration mode of dynamic mapping.

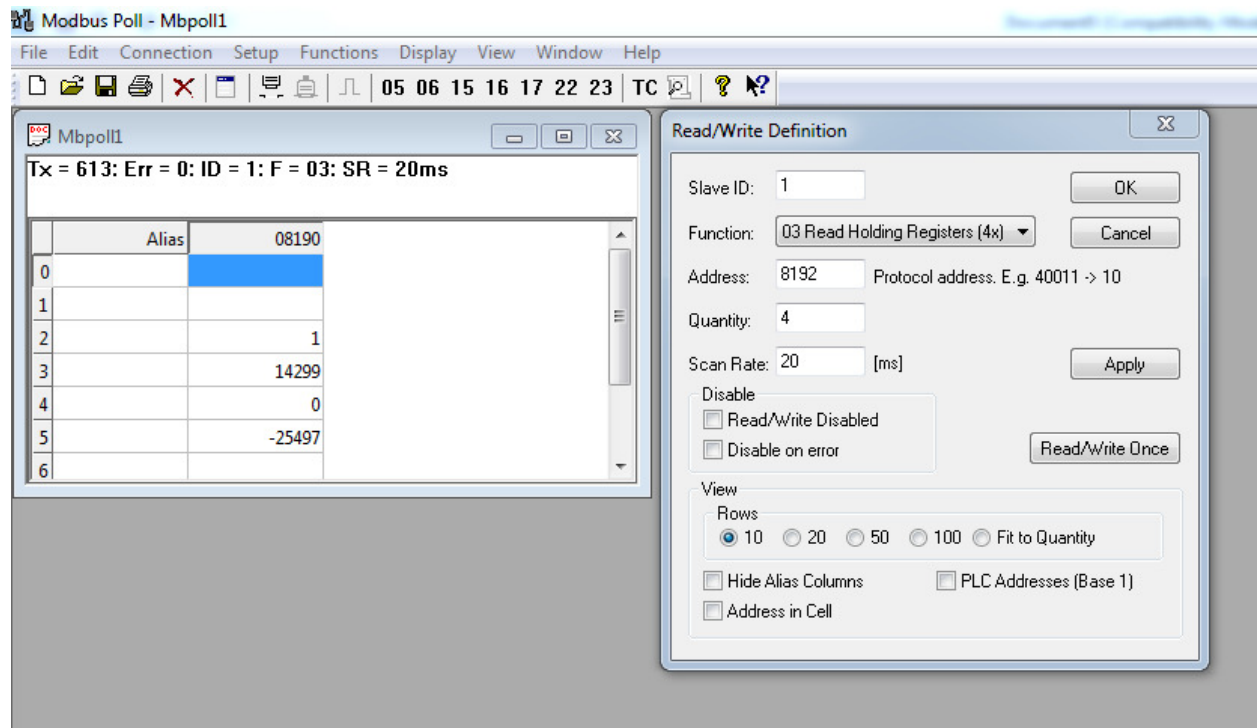
Use the MODBUS.ADDRxxxx yyyy format to map parameters where xxxx is the dynamic mapping range of addresses and yyyy is the address from any of the other ranges.

Type “MODBUS.DYNMAP 0” to disable configuration mode of dynamic mapping (values will begin to pass from the original parameters to the dynamically mapped addresses).

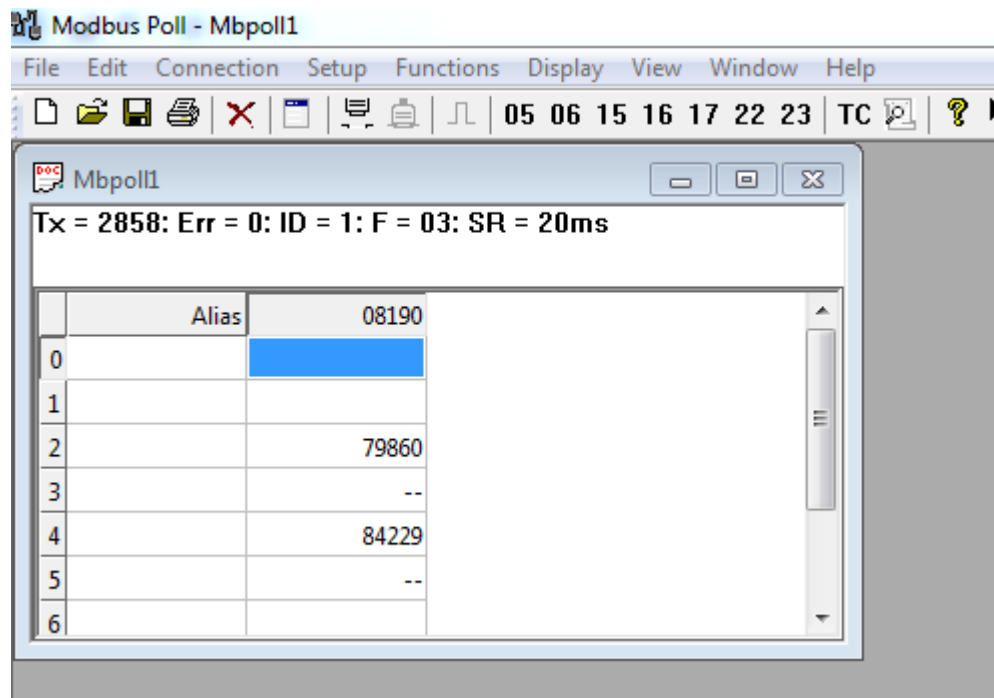
Save To Device when done so the mapping is saved to the drive’s non-volatile memory.



Now to look at the read data at the Modbus master (I'm using Modbus Poll).



Now to format the viewed data to 32 bits



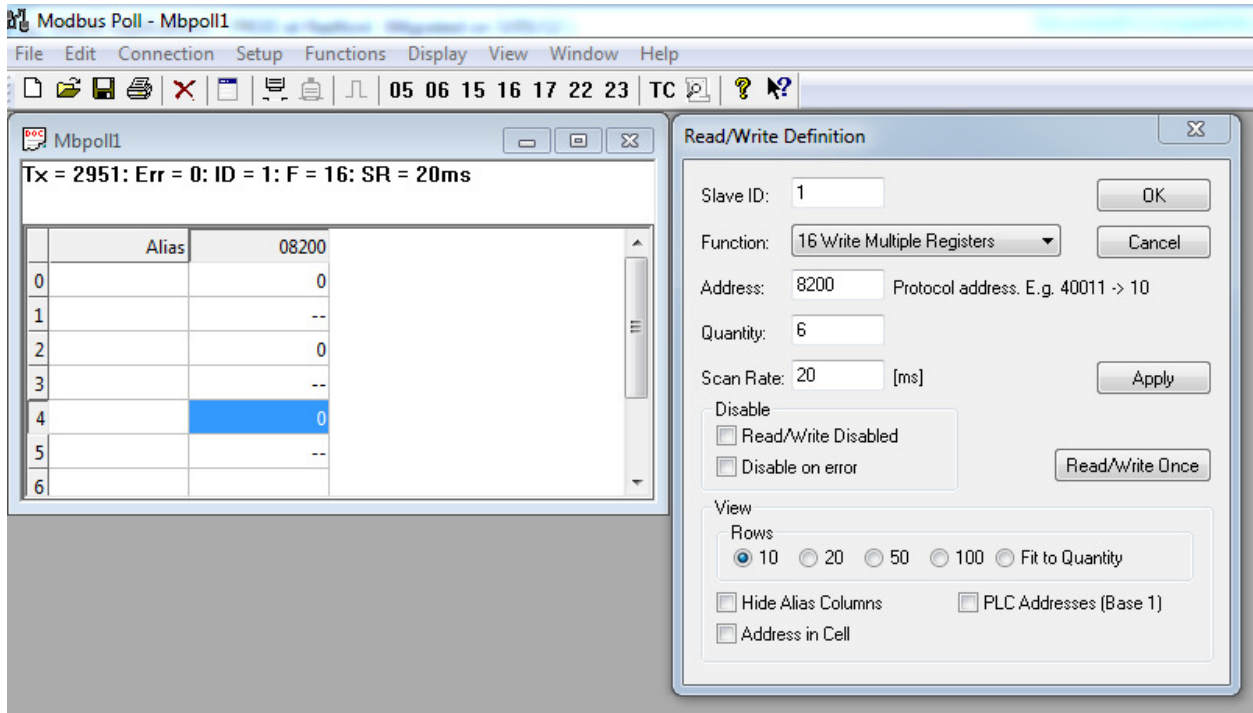
-->PL.FB

79.878 [counts]

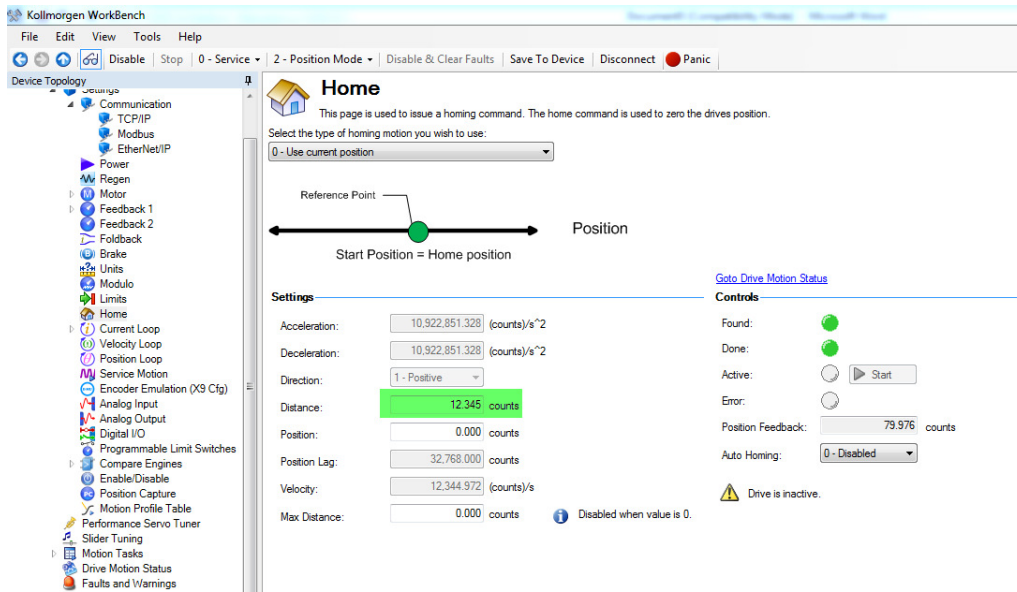
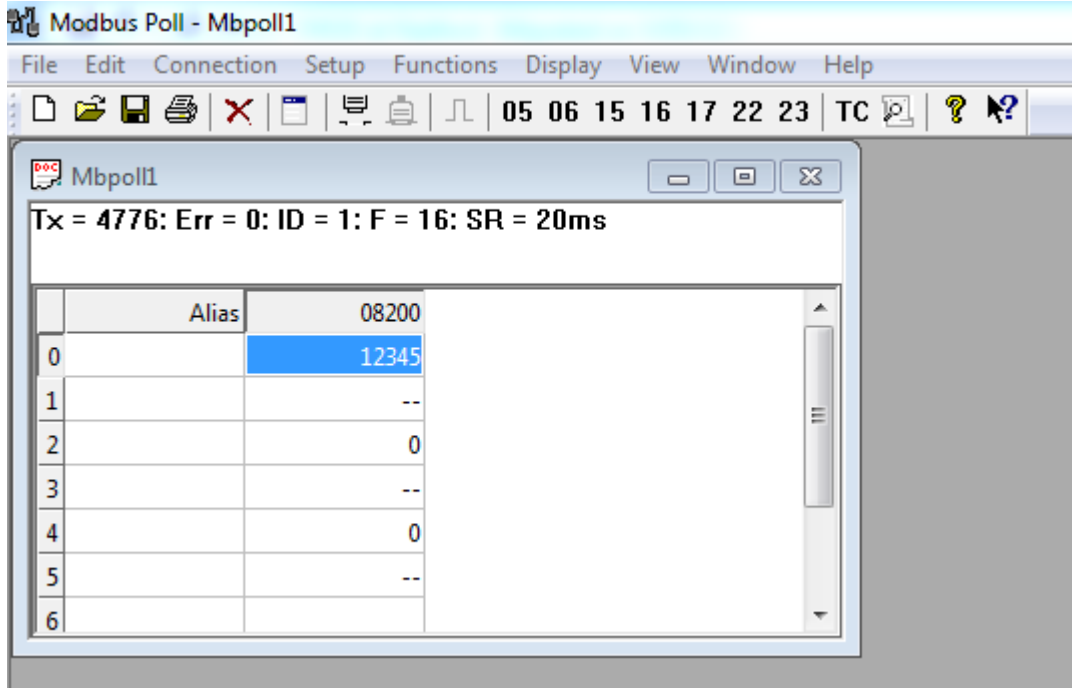
-->

VL.FB is going to vary in the value read.

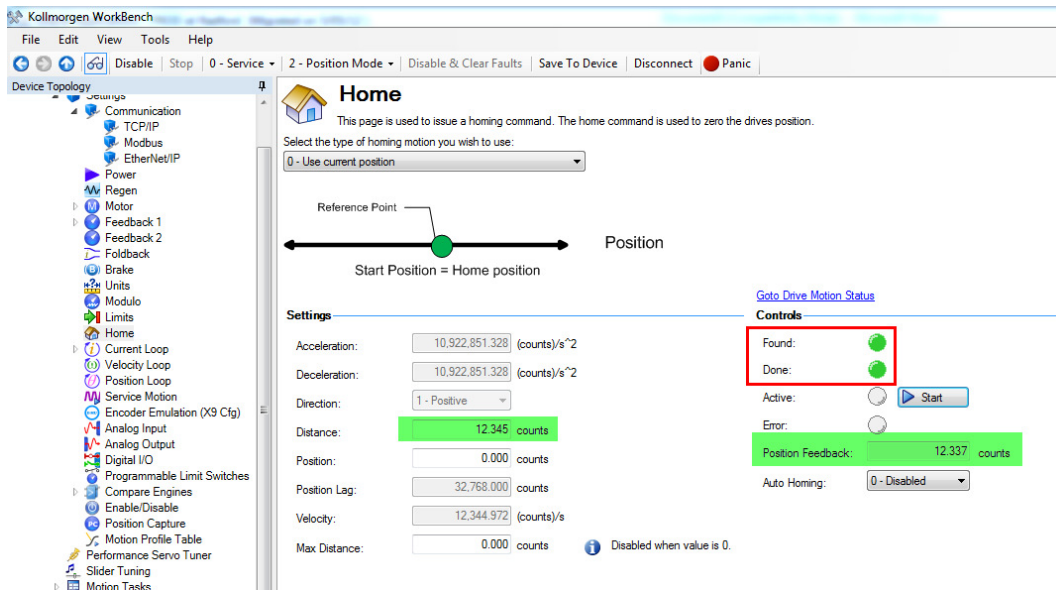
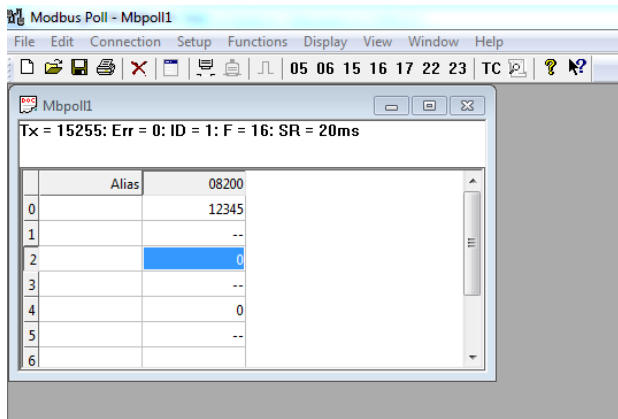
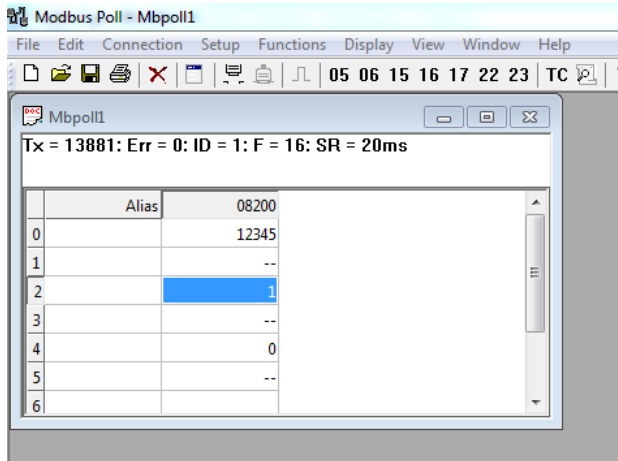
Now to validate the writes (I went ahead and formatted the viewed data to 32 bits):



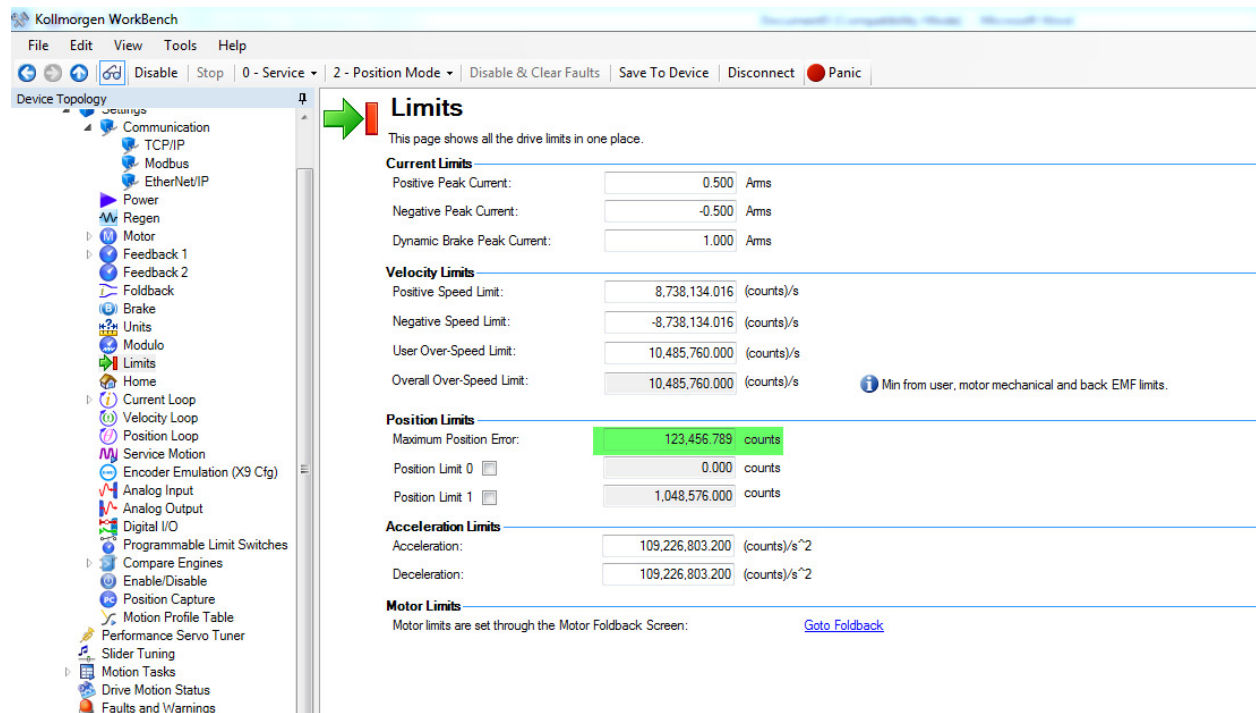
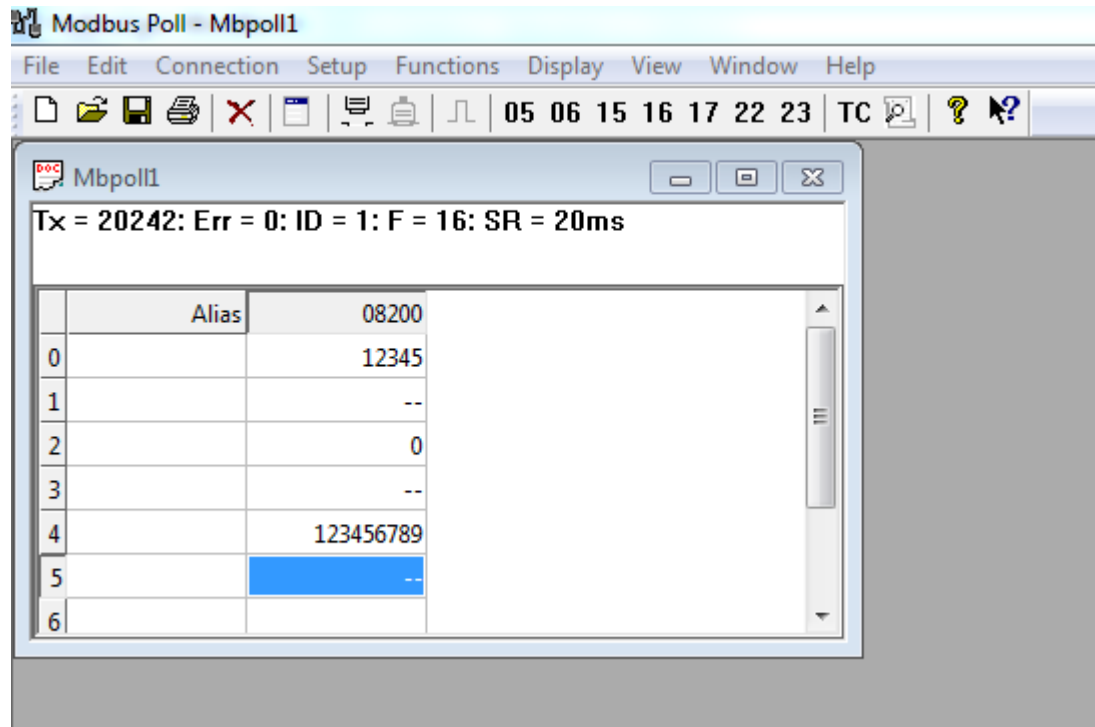
First, I set the HOME.DIST value to 12345.



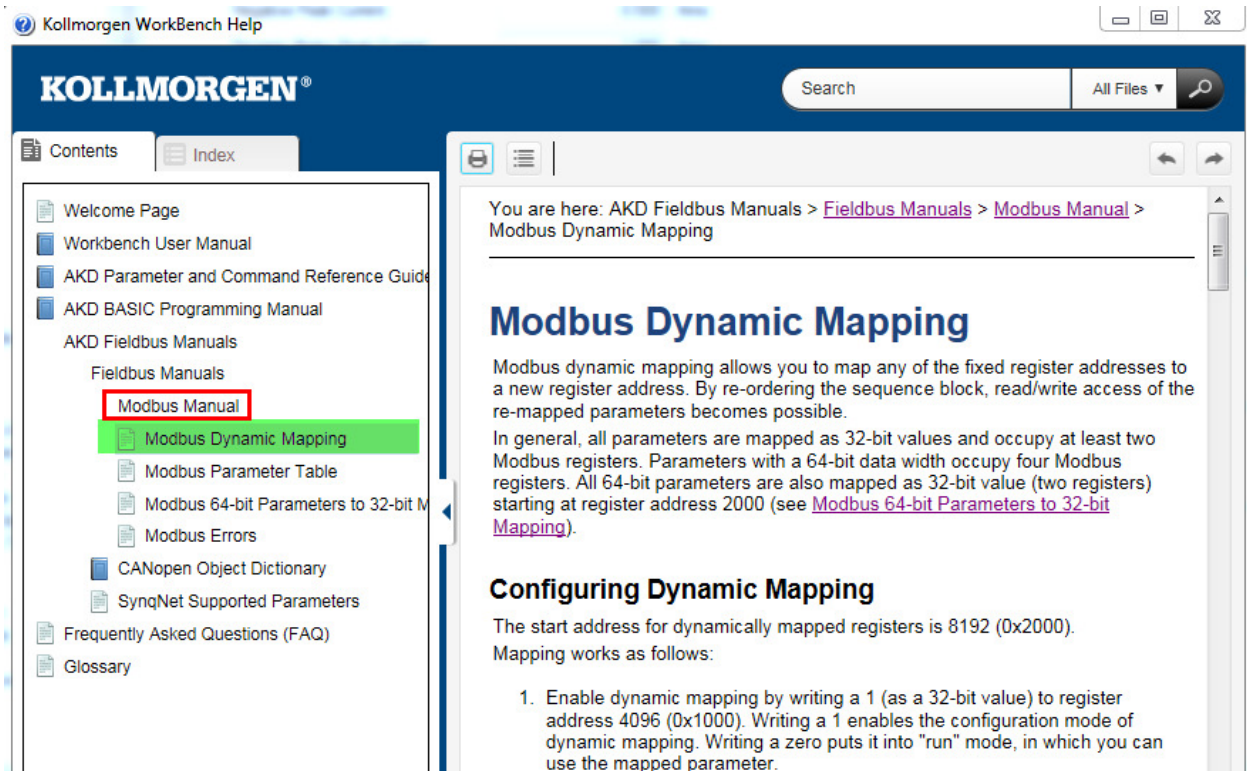
Next I executed a HOME.MOVE



Finally I set the following error threshold:



Refer to Workbench Help for more details on Modbus TCP and Dynamic Mapping:



The screenshot displays the Kollmorgen WorkBench Help interface. The top navigation bar includes the Kollmorgen logo, a search bar, and a file type dropdown set to 'All Files'. The left sidebar contains a 'Contents' menu with a tree structure: 'Welcome Page', 'Workbench User Manual', 'AKD Parameter and Command Reference Guide', 'AKD BASIC Programming Manual', 'AKD Fieldbus Manuals', 'Fieldbus Manuals', 'Modbus Manual' (highlighted with a red box), 'Modbus Dynamic Mapping' (highlighted with a green box), 'Modbus Parameter Table', 'Modbus 64-bit Parameters to 32-bit M', 'Modbus Errors', 'CANopen Object Dictionary', 'SynqNet Supported Parameters', 'Frequently Asked Questions (FAQ)', and 'Glossary'. The main content area shows the breadcrumb 'You are here: AKD Fieldbus Manuals > Fieldbus Manuals > Modbus Manual > Modbus Dynamic Mapping'. The page title is 'Modbus Dynamic Mapping'. The text explains that Modbus dynamic mapping allows mapping fixed register addresses to new ones, and that parameters are mapped as 32-bit values. A section titled 'Configuring Dynamic Mapping' states that the start address is 8192 (0x2000) and provides a list of steps to enable dynamic mapping.

Modbus Dynamic Mapping

Modbus dynamic mapping allows you to map any of the fixed register addresses to a new register address. By re-ordering the sequence block, read/write access of the re-mapped parameters becomes possible.

In general, all parameters are mapped as 32-bit values and occupy at least two Modbus registers. Parameters with a 64-bit data width occupy four Modbus registers. All 64-bit parameters are also mapped as 32-bit value (two registers) starting at register address 2000 (see [Modbus 64-bit Parameters to 32-bit Mapping](#)).

Configuring Dynamic Mapping

The start address for dynamically mapped registers is 8192 (0x2000). Mapping works as follows:

1. Enable dynamic mapping by writing a 1 (as a 32-bit value) to register address 4096 (0x1000). Writing a 1 enables the configuration mode of dynamic mapping. Writing a zero puts it into "run" mode, in which you can use the mapped parameter.