

AKD Communications Application Note: Telnet and Modbus TCP Rev. A Jan/2016

Overview

Although all AKDs (except Profinet) support Telnet and Modbus TCP, this application note applies specifically to the Motion Tasking (AKD-P) drive.

The AKD-Pxxxxx-NBAN-0000 supports Telnet and Modbus TCP for communications over Ethernet.

Ethernet TCP/IP is a generic term used with Ethernet networks and is not synonymous with Ethernet IP an open protocol managed by the ODVA organization.

Telnet

Telnet is what is used in the Workbench communications to the drive. You can see an example when you use the Workbench terminal.

Essentially Telnet is ASCII over Ethernet but for more details to the standard here is a URL to more information.

<http://en.wikipedia.org/wiki/Telnet>

We do not have examples specific to any controller or software but there are a couple of documents and files on our KDN website that may be helpful. One is a VB example and the other is using Hyperterminal.

http://kdn.kollmorgen.com/query/telnet?f0=im_field_tags%3A252

VB Telnet Example

Attached is an example Visual Basic **Telnet** communications program to talk to the AKD. No ...
Last modified: Monday, December 8, 2014- 12:33 Dan.Wolke@Kollmorgen.com AKD **Telnet** VB
Visual Basic ...

AKD Connection using HyperTerminal

11:28 jcoleman02 AKD Ethernet Hyperterminal **Telnet** communication ...

Protocols Supported by AKD Drive

Current general market model numbers and their supported protocols Model Number **Telnet**
Modbus ... 2014- 08:34 Will Gaffga AKD CANopen EtherCAT Modbus Profinet **Telnet** model
number protocol English ...

Modbus TCP

An alternative to Telnet is Modbus TCP. There are a myriad of different products from PC based applications using a Modbus TCP driver plug-in (i.e. ActiveX, dll, etc.) to HMIs and PLCs.

Another option is Modbus TCP. Modbus TCP is an open protocol managed by its organization. Unlike Telnet which is ASCII over Ethernet based, Modbus TCP is a register/address based communications protocol.

<http://www.modbus.org>

Demonstration of Telnet

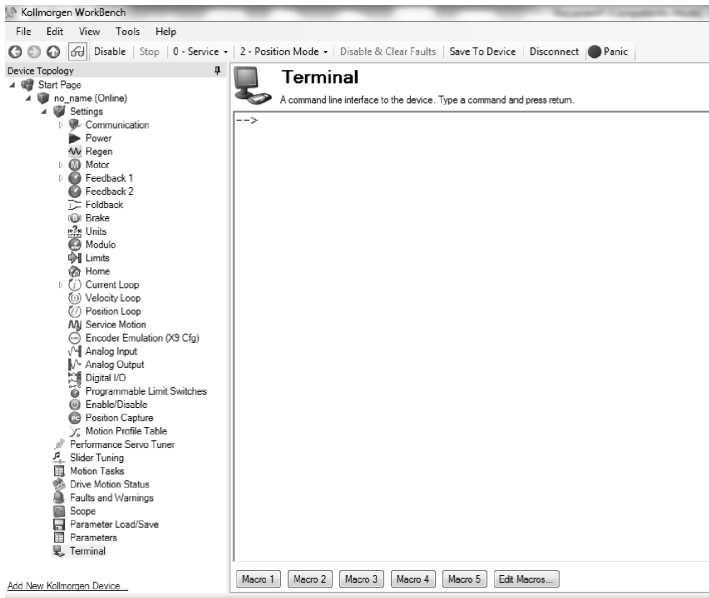
This application note will demonstrate setting up a motion task via Hyperterminal and also describe where to find the documentation on Modbus TCP.

In the Motion Tasking drive is the table of Motion Task as shown:

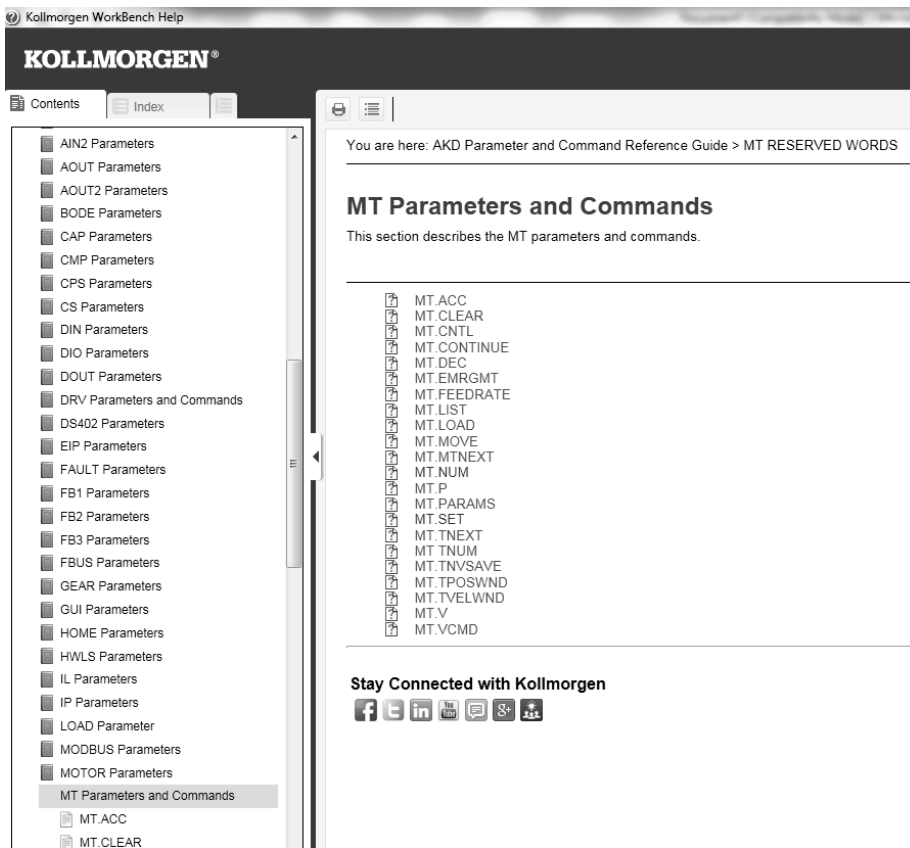
The screenshot shows the Kollmorgen WorkBench interface. On the left is a 'Device Topology' tree with various settings like Communication, Power, Regen, Motor, Feedback, Brake, Units, Modulo, Limits, Home, Current Loop, Velocity Loop, Position Loop, Service Motion, Encoder Emulation, Analog Input/Output, Digital I/O, Programmable Limit Switches, Enable/Disable, Position Capture, Motion Profile Table, Performance Servo Tuner, Slider Tuning, Motion Tasks, Drive Motion Status, and Faults and Warnings. The main window is titled 'Motion Tasks' and contains a 'Start' button and a status indicator 'Motion Task Running: Idle'. Below this is a table with the following columns: Position [Counts16Bit], Velocity [rpm], Acceleration [rpm/s], and Deceleration [rpm/s]. The table has 18 rows, with row 16 highlighted.

	Position [Counts16Bit]	Velocity [rpm]	Acceleration [rpm/s]	Deceleration [rpm/s]
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
▶ 16				
17				

Since Workbench has its own terminal I would suggest you start your application by issuing commands from the Workbench terminal. The advantage is you can set values using the command line (as you would in Hyperterminal or other 3rd party software) but also verify the values are written. This will give you a feel for our command/parameter naming convention.



From the Workbench help you will find the AKD Parameter and Command Reference Guide. These parameters are grouped by function. Not exclusive but you will initially find interest in the DRV, HOME, and MT groups (Drive, Homing, Motion Task).



The same commands can be done within Workbench but to demonstrate this can be done externally, I followed the setup for Hyperterminal from the app. note on the website.

This is purely for demonstrational purposes to indicate how one might control the drive by sending commands over Ethernet. Every application is different.

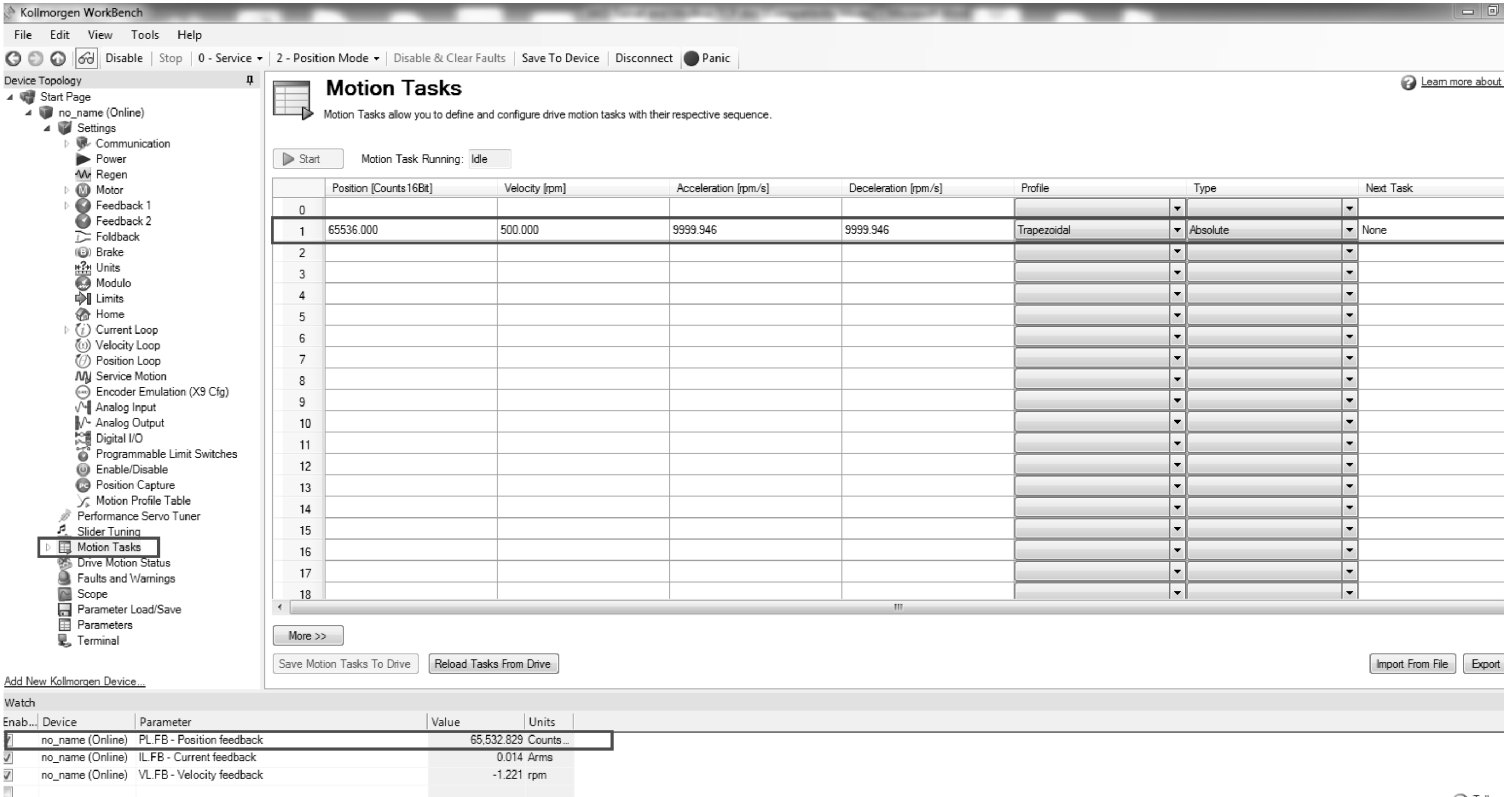
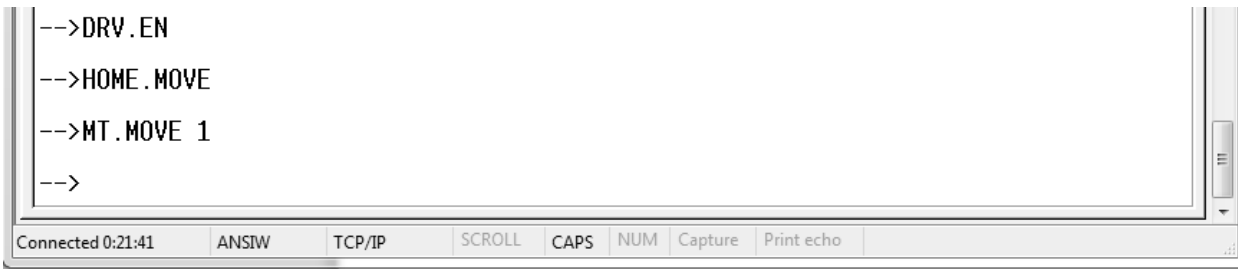
In this example, the drive is software disabled, enabled, the operation mode was changed to position mode (2), the drive was then disabled. Next I set the motion task number (pointer) to 1 which is motion task 1 (row one if you were to check it in Workbench). Units are setup in Workbench but are not the focus of this app. note. Once motion task 1 was selected, I set the position, velocity, accel, decel for the move. Note the MT.CNTL value 0 results in an absolute move. If other move types are required or other advanced features of the motion task needs to be set, the MT.CNTL must be set to a decimal equivalent value of the binary bits within the MT.CNTL. Please see the Workbench help for more details and the definition of this parameter. MT.SET moves the current values of the MT parameters into the current motion task based on the MT.NUM (1 in this case). This method could be repeated by setting MT.NUM to another number (i.e. MT.NUM 2) to setup other motion tasks. To save the motion tasks and all drive parameter settings, issue a DRV.NVSAVE so everything will be nonvolatile on power cycle of the drive.



```
-->DRV.DIS
-->DRV.EN
-->DRV.OPMODE 2
-->DRV.DIS

-->MT.NUM 1
-->MT.P 65536
-->MT.V 500
-->MT.ACC 10000
-->MT.DEC 10000
-->MT.CNTL 0
-->MT.SET
-->DRV.NVSAVE
```

Next I enabled the drive (software enable), commanded the drive to execute the home routine (see the Home screen in your project tree in Workbench for details). Finally I commanded Motion Task 1 to execute. Based on my units and scaling in the drive with 16 bit position units (65536=1rev), the motor moved 1 revolution.

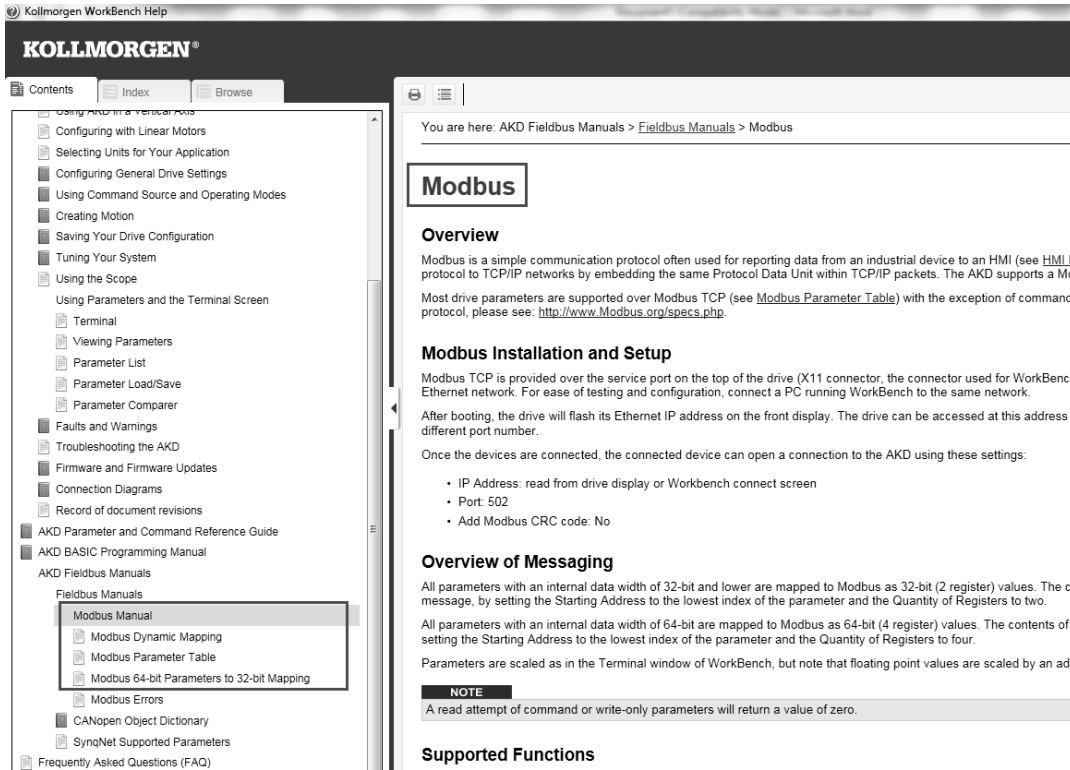


If your controller (PC, PLC, etc.) can emulate the same functionality as demonstrated with Hyperterminal then you can control the AKD using the same methodology. Note this doesn't go into details regarding communication errors or confirmation of read/write success (i.e. checksums).

Modbus TCP

An alternative to Telnet is Modbus TCP.

In Workbench Help there is a section for fieldbus manuals. Under that section is “Modbus Manual”. For users of AKD BASIC note that it also has a variable mapping range of 5000 to 5999 (see the AKD BASIC Programming manual for more details).



There are 2 tables with the drive parameters: The Modbus Parameter table (note the parameter, register address, and attribute/data types are given). There is also a Modbus 64bit to 32 bit mapping where some parameters that are in the first table dimensioned as a 64bit word are in the 32 bit mapping table redimensioned as a 32 bit word. In some cases this is easier to read/write data depending on your Modbus TCP drive/controller. As noted, Modbus is a register/address based protocol.

