Micrologix 1400 Modbus TCP Sample Program for AKD BASIC

2/7/2017 Revision A

Typically AB users will use Ethernet/IP to communicate to an AKD (AKD-P-NxEI) drive but some have been motivated for various reasons to use a standard drive (AKD-P-NxAN) and use Modbus TCP instead. The sample program was originally written for an application that involved a Micrologix 1400 to the AKD BASIC hence the program will demonstrate writing to the AKD BASIC 5000 range Modbus registers. The method of writing and reading is not restricted to the AKD BASIC but applies to all the AKD drives that support Modbus TCP (all but the ProfiNET version AKD-P-NxPN).

The following application note will provide details on the setup and execution of the sample program.

The intention of the sample project and this application note is to allow the first time user to perform a bench test to establish a connection between the PLC and 1 AKD drive and verify Modbus TCP communications.

Included in the zip file along with the sample project are:

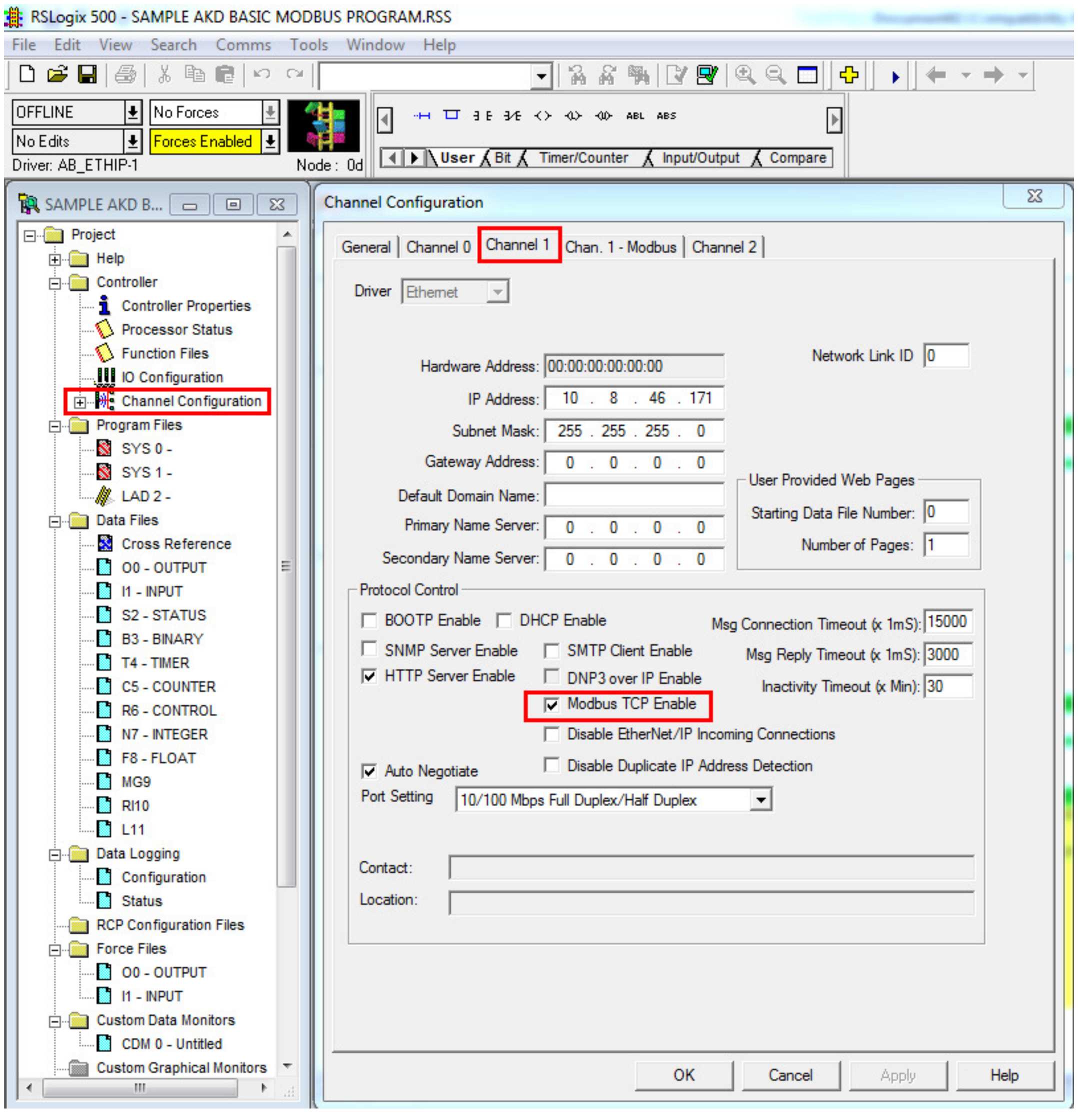
- *.BAS program
- *.AKD parameter file
- *.RSS RSLogix500 ladder project

Notes on using sample project

To use the sample project the communication channel must be setup for the target PLC.

For example I setup my Micrologix 1400 using the PLC's keypad on the front to set the address to 192.168.0.12. This application note will not go into the details; it is assumed the user is familiar with setting this.

Open the sample program offline and double-click the Channel Configuration in the project tree in RSLogix 500. The Channel Configuration window will appear.


RSLogix 500 - SAMPLE AKD BASIC MODBUS PROGRAM.RSS

Channel Configuration

General | Channel 0 | Channel 1 | Chan. 1 - Modbus | Channel 2

Driver: Ethernet

Hardware Address: 00:00:00:00:00:00

IP Address: 10 . 8 . 46 . 171

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

Network Link ID: 0

User Provided Web Pages
Starting Data File Number: 0
Number of Pages: 1

Protocol Control
- [ ] BOOTP Enable
- [ ] DHCP Enable
- [ ] SNMP Server Enable
- [ ] SMTP Client Enable
- [✓] HTTP Server Enable
- [ ] DNP3 over IP Enable
- [✓] Modbus TCP Enable
- [ ] Disable EtherNet/IP Incoming Connections
- [✓] Auto Negotiate
- [ ] Disable Duplicate IP Address Detection

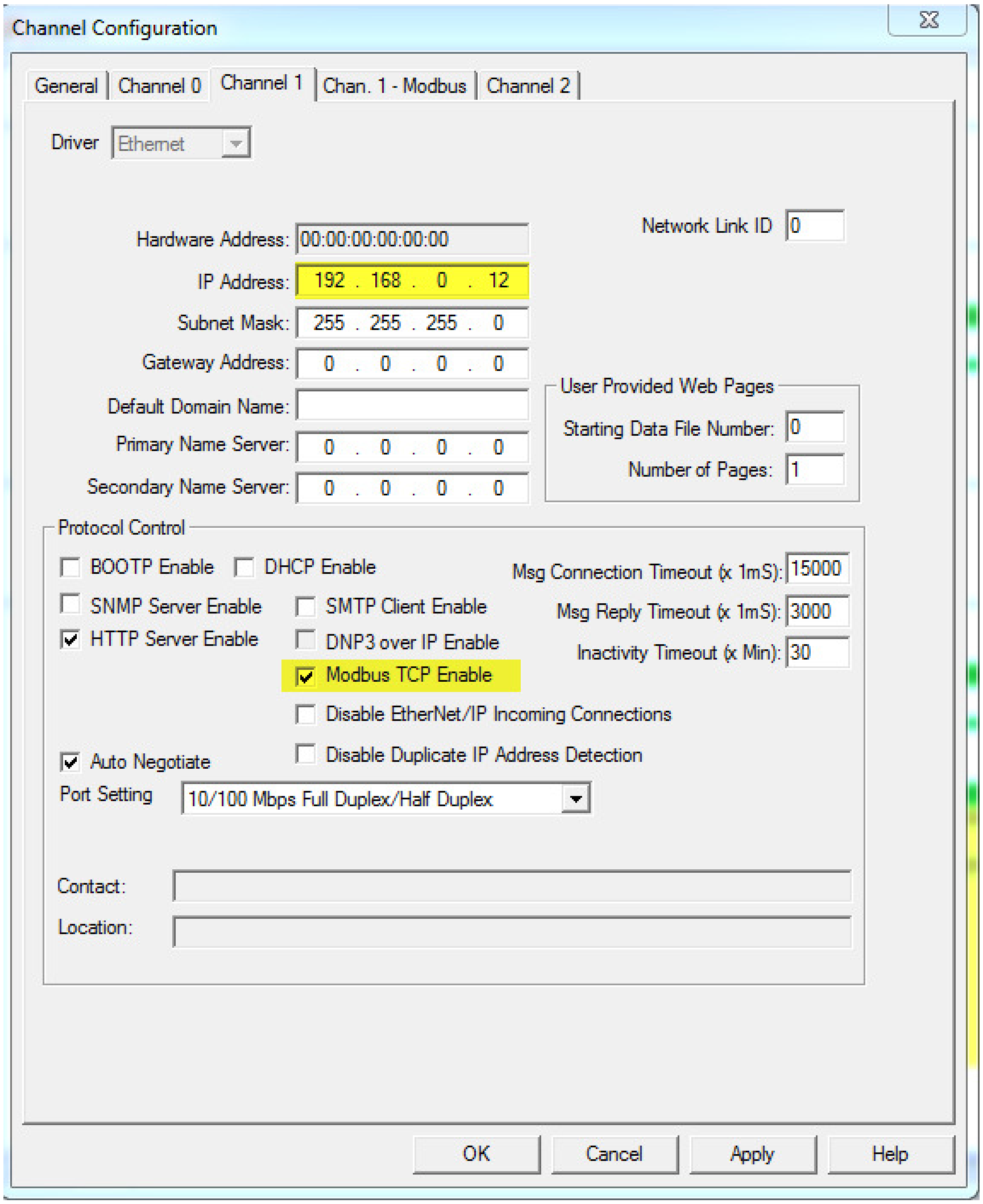
Msg Connection Timeout (x 1mS): 15000
Msg Reply Timeout (x 1mS): 3000
Inactivity Timeout (x Min): 30

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Contact:

Location:

I changed the IP Address to match my PLC's. Note the other settings but in particular the "Modbus TCP Enable". This will need to be checked to enable Modbus TCP protocol.



Channel Configuration

General | Channel 0 | Channel 1 | Chan. 1 - Modbus | Channel 2

Driver: Ethernet

Hardware Address: 00:00:00:00:00:00

IP Address: 192 . 168 . 0 . 12

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

Network Link ID: 0

User Provided Web Pages

Starting Data File Number: 0

Number of Pages: 1

Protocol Control

- [ ] BOOTP Enable
- [ ] DHCP Enable
- [ ] SNMP Server Enable
- [ ] SMTP Client Enable
- [✓] HTTP Server Enable
- [ ] DNP3 over IP Enable
- [✓] Modbus TCP Enable
- [ ] Disable EtherNet/IP Incoming Connections
- [✓] Auto Negotiate
- [ ] Disable Duplicate IP Address Detection

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

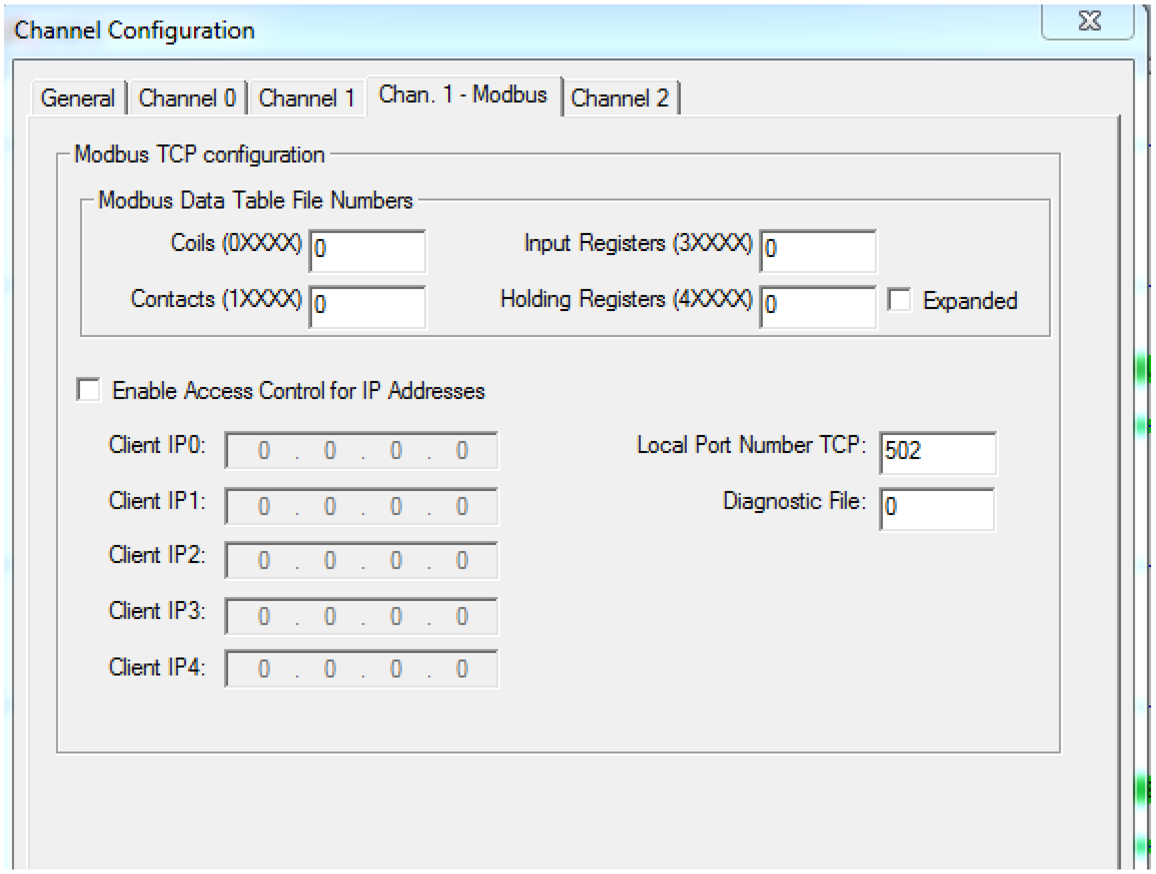
Inactivity Timeout (x Min): 30

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Contact:

Location:

You may have noticed there is a tab “Chan. 1-Modbus”. Other than the “Local Port Number TCP” being the standard 502 no other settings were utilized.



Channel Configuration

General | Channel 0 | Channel 1 | Chan. 1 - Modbus | Channel 2

Modbus TCP configuration

Modbus Data Table File Numbers

Coils (0XXXX)	0	Input Registers (3XXXX)	0
Contacts (1XXXX)	0	Holding Registers (4XXXX)	0 ☐ Expanded

☐ Enable Access Control for IP Addresses

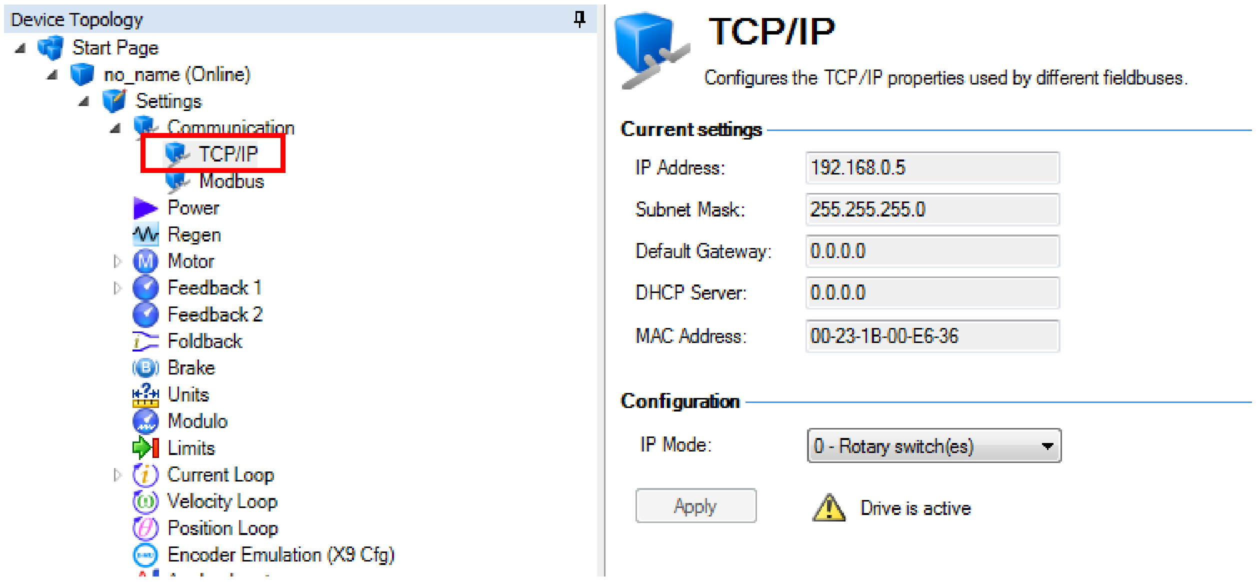
Client IP0:	0 . 0 . 0 . 0	Local Port Number TCP:	502
Client IP1:	0 . 0 . 0 . 0	Diagnostic File:	0
Client IP2:	0 . 0 . 0 . 0		
Client IP3:	0 . 0 . 0 . 0		
Client IP4:	0 . 0 . 0 . 0		

Click Apply and OK to close out the Channel Configuration window.

Next it is important to setup and know what the target IP address is; that is the AKD drive's IP Address.

The AKD can be setup with the 192.168.0.x convention using the rotary dials on the front of the drive or by using Workbench to set a static IP address for the drive.

In my case, online in Workbench you can see under the Online drive->Settings->Communication->TCP/IP screen that the IP mode of my drive is "0-Rotary Switches". Mode 1 could have also been used which is 1-Fixed IP Address.



TCP/IP

Configures the TCP/IP properties used by different fieldbuses.

Current settings

IP Address:	192.168.0.5
Subnet Mask:	255.255.255.0
Default Gateway:	0.0.0.0
DHCP Server:	0.0.0.0
MAC Address:	00-23-1B-00-E6-36

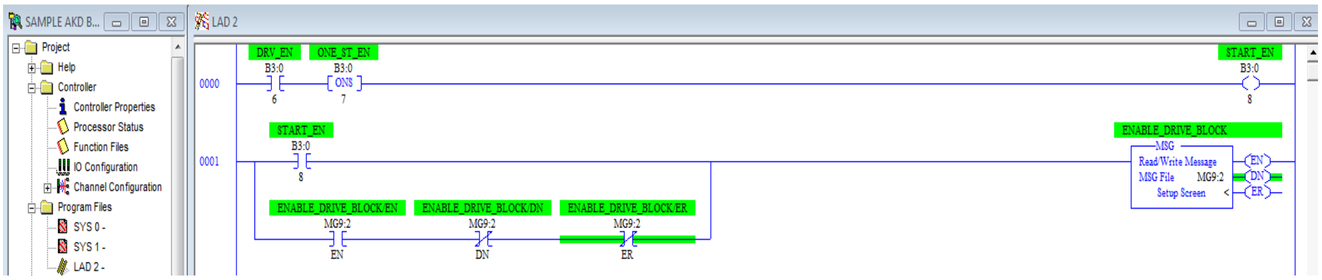
Configuration

IP Mode: 0 - Rotary switch(es)

Drive is active

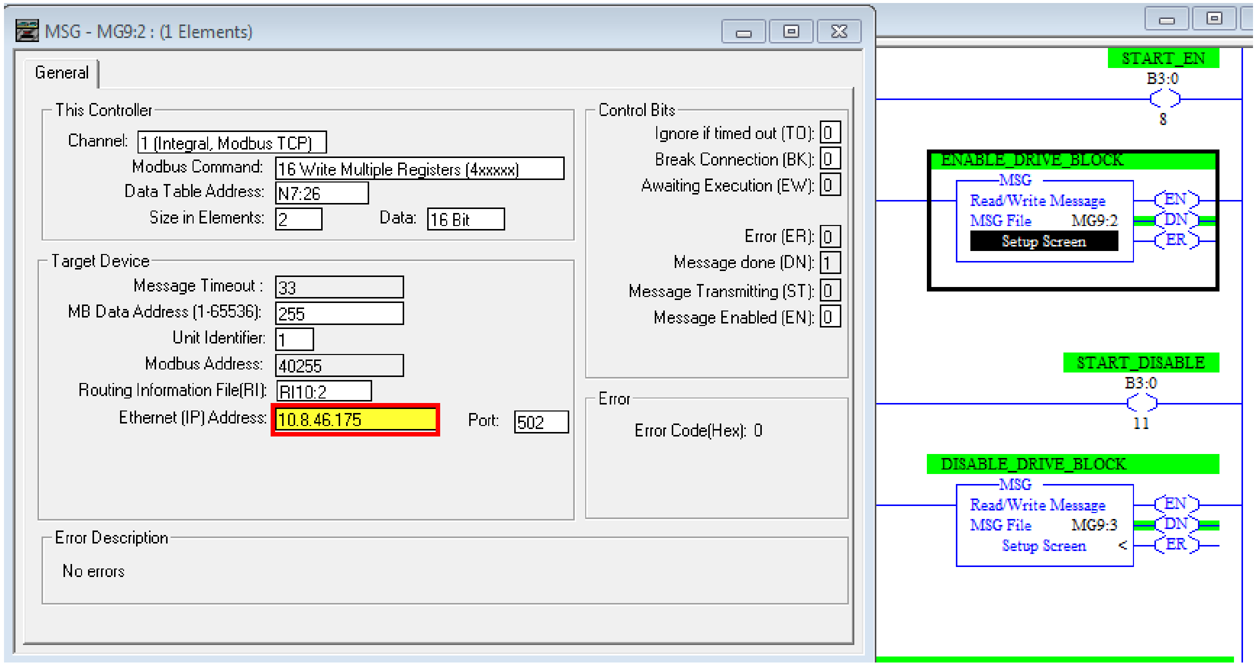
Now that the drive's IP address has been established, it is important to modify the sample project so the target IP address matches the AKD drive's. The Micrologix 1400 supports Modbus TCP via the MSG (Message) block. This means a MSG block must be triggered one at a time to read or write to an AKD drive parameter (more on this later) or a group of consecutive registers (multiple read or writes).

Begin by editing the offline program beginning with the first MSG block on rung 1.



Double-click on the “Setup Screen” inside the ENABLE_DRIVE_BLOCK MSG to bring up the MSG’s configuration screen.

The Ethernet IP address will need to be changed to the AKD drive’s IP Address.


MSG - MG9:2 : (1 Elements)

General

This Controller

Channel: 1 (Integral, Modbus TCP)
Modbus Command: 16 Write Multiple Registers (4xxxxx)
Data Table Address: N7:26
Size in Elements: 2 Data: 16 Bit

Target Device

Message Timeout : 33
MB Data Address (1-65536): 255
Unit Identifier: 1
Modbus Address: 40255
Routing Information File(RI): RI10:2
Ethernet (IP) Address: 10.8.46.175 Port: 502

Control Bits

Ignore if timed out (TO): 0
Break Connection (BK): 0
Awaiting Execution (EW): 0
Error (ER): 0
Message done (DN): 1
Message Transmitting (ST): 0
Message Enabled (EN): 0

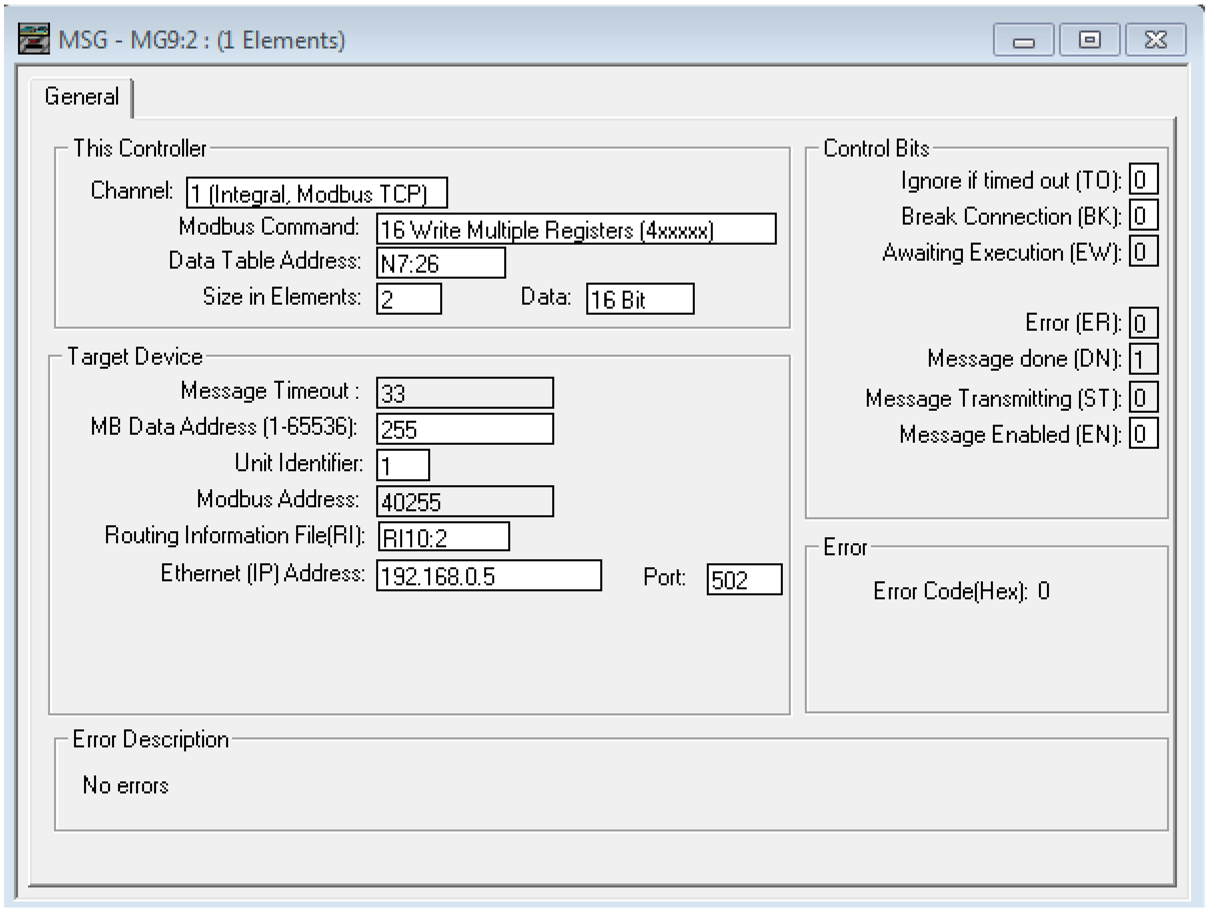
Error

Error Code(Hex): 0

Error Description

No errors

In my case (as shown before) my AKD's IP Address is 192.168.0.5.


MSG - MG9:2 : (1 Elements)

General

This Controller

Channel: 1 (Integral, Modbus TCP)

Modbus Command: 16 Write Multiple Registers (4xxxxx)

Data Table Address: N7:26

Size in Elements: 2

Data: 16 Bit

Target Device

Message Timeout : 33

MB Data Address (1-65536): 255

Unit Identifier: 1

Modbus Address: 40255

Routing Information File(RI): RI10:2

Ethernet (IP) Address: 192.168.0.5

Port: 502

Control Bits

Ignore if timed out (TO): 0

Break Connection (BK): 0

Awaiting Execution (EW): 0

Error (ER): 0

Message done (DN): 1

Message Transmitting (ST): 0

Message Enabled (EN): 0

Error

Error Code(Hex): 0

Error Description

No errors

Close the MSG Setup Screen->Right Click on the Edited Rung->Verify Rung.

This process must be repeated for every MSG block in the Sample Project.

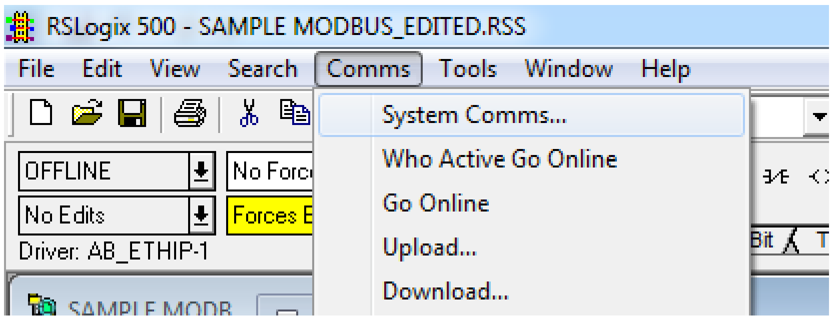
In summary these instances are:

Rung Number	**Name**	**Address**
1	ENABLE_DRIVE_BLOCK	MG9:2
3	DISABLE_DRIVE_BLOCK	MG9:3
5	WRITE_TO_DRIVE_BLOCK	MG9:0
7	READ_FROM_DRIVE_BLOC	MG9:1

After editing, either save the project or save the project under a different name and keep the original for your records (recommended).

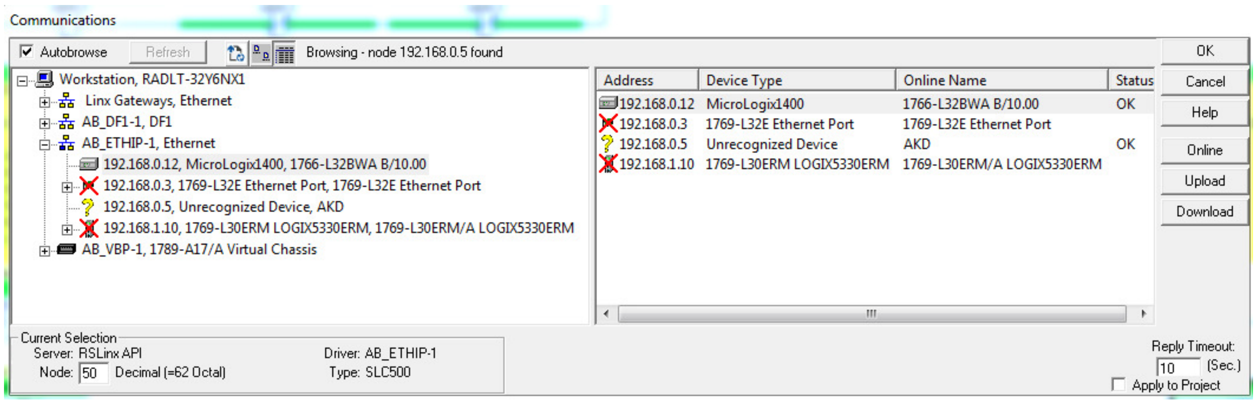
The project is ready to download and test.

From the Comms pull-down menu in RSLogix 500, select the Systems Comms…



In my case RSLinx is already running.

I highlighted the Micrologix 1400 and then clicked on "Download"



Browsing - node 192.168.0.5 found

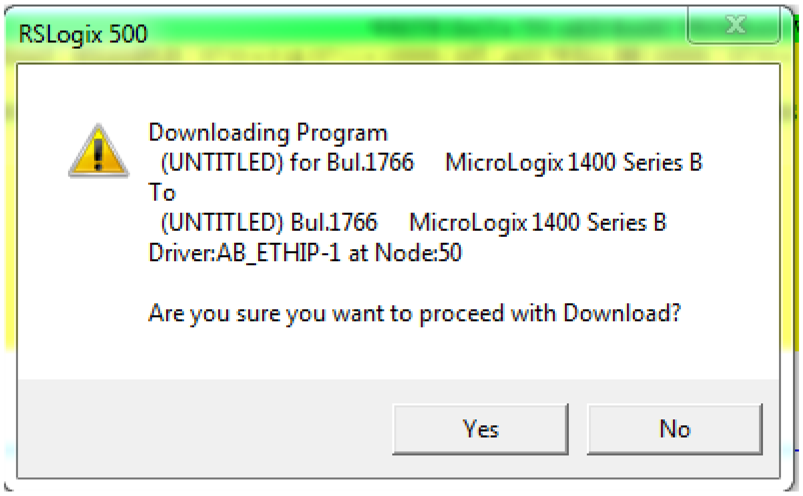
Workstation, RADLT-32Y6NX1
- Linx Gateways, Ethernet
- AB_DF1-1, DF1
- AB_ETHIP-1, Ethernet
  - 192.168.0.12, MicroLogix1400, 1766-L32BWA B/10.00
  - 192.168.0.3, 1769-L32E Ethernet Port, 1769-L32E Ethernet Port
  - 192.168.0.5, Unrecognized Device, AKD
  - 192.168.1.10, 1769-L30ERM LOGIX5330ERM, 1769-L30ERM/A LOGIX5330ERM
- AB_VBP-1, 1789-A17/A Virtual Chassis

Address	Device Type	Online Name	Status
192.168.0.12	MicroLogix1400	1766-L32BWA B/10.00	OK
192.168.0.3	1769-L32E Ethernet Port	1769-L32E Ethernet Port	
192.168.0.5	Unrecognized Device	AKD	OK
192.168.1.10	1769-L30ERM LOGIX5330ERM	1769-L30ERM/A LOGIX5330ERM	

Current Selection
Server: RSLinx API
Node: 50 Decimal (=62 Octal)
Driver: AB_ETHIP-1
Type: SLC500

Reply Timeout: 10 (Sec.)
Apply to Project

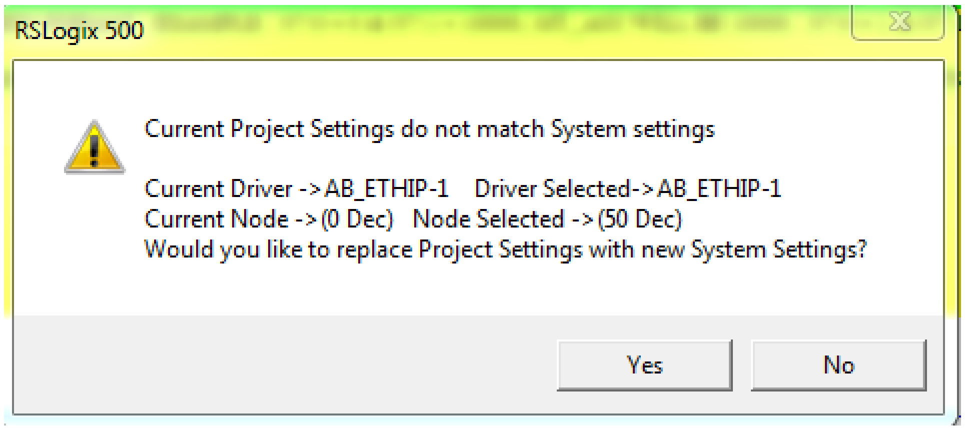
The following window will appear. Click Yes.



Downloading Program
(UNTITLED) for Bul.1766 MicroLogix 1400 Series B
To
(UNTITLED) Bul.1766 MicroLogix 1400 Series B
Driver:AB_ETHIP-1 at Node:50

Are you sure you want to proceed with Download?

The following window will appear. Click Yes.

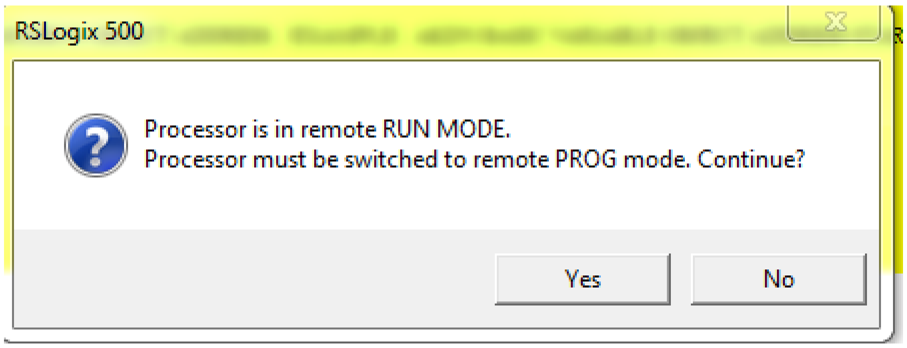


RSLogix 500

Current Project Settings do not match System settings

Current Driver ->AB_ETHIP-1 Driver Selected->AB_ETHIP-1
Current Node ->(0 Dec) Node Selected ->(50 Dec)
Would you like to replace Project Settings with new System Settings?

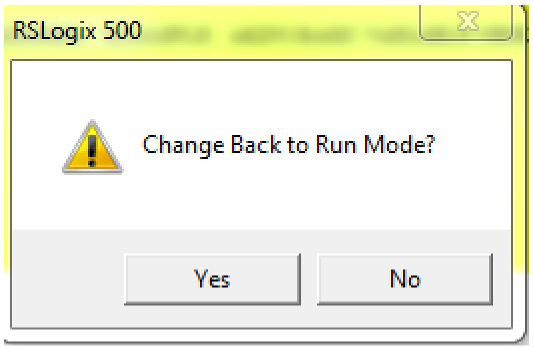
In my case the processor was in RUN MODE. I clicked Yes to switch to PROG mode.



RSLogix 500

Processor is in remote RUN MODE.
Processor must be switched to remote PROG mode. Continue?

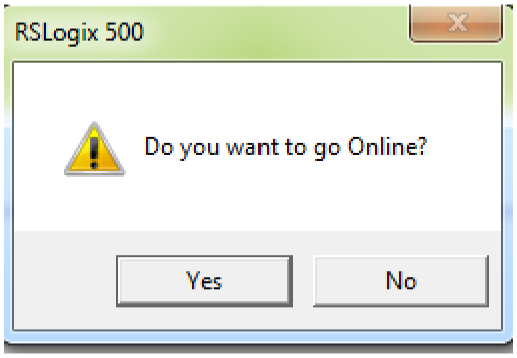
After downloading the following window appears. I clicked Yes to switch back to RUN MODE.



RSLogix 500

Change Back to Run Mode?

After going to RUN MODE the following window appears. I clicked Yes to go Online.


RSLogix 500

Do you want to go Online?

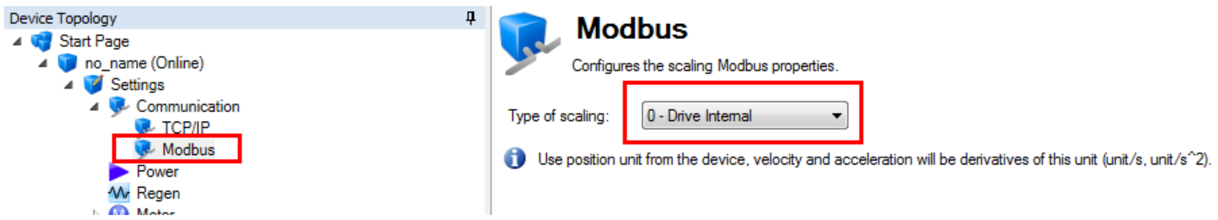
The zip file contains the AKD setup used to test the sample project. It will be necessary to change the motor and feedback to match your case.

To demonstrate the details of setup, I set my drive up with minimal settings and then will demonstrate key settings.

Before triggering any of the Modbus TCP read/writes to the drive verify a couple of settings in the AKD using Workbench.

The first setting is in the online drive's project tree under Settings->Communication->Modbus.

The default type of scaling is "1-Modbus Specific". Although I've seen on occasion someone interested in using it, it is quite rare. In general, the most popular setup is to set the type of scaling to "0-Drive Internal". So that no additional scaling is performed between Workbench units and the values read or written over Modbus TCP.



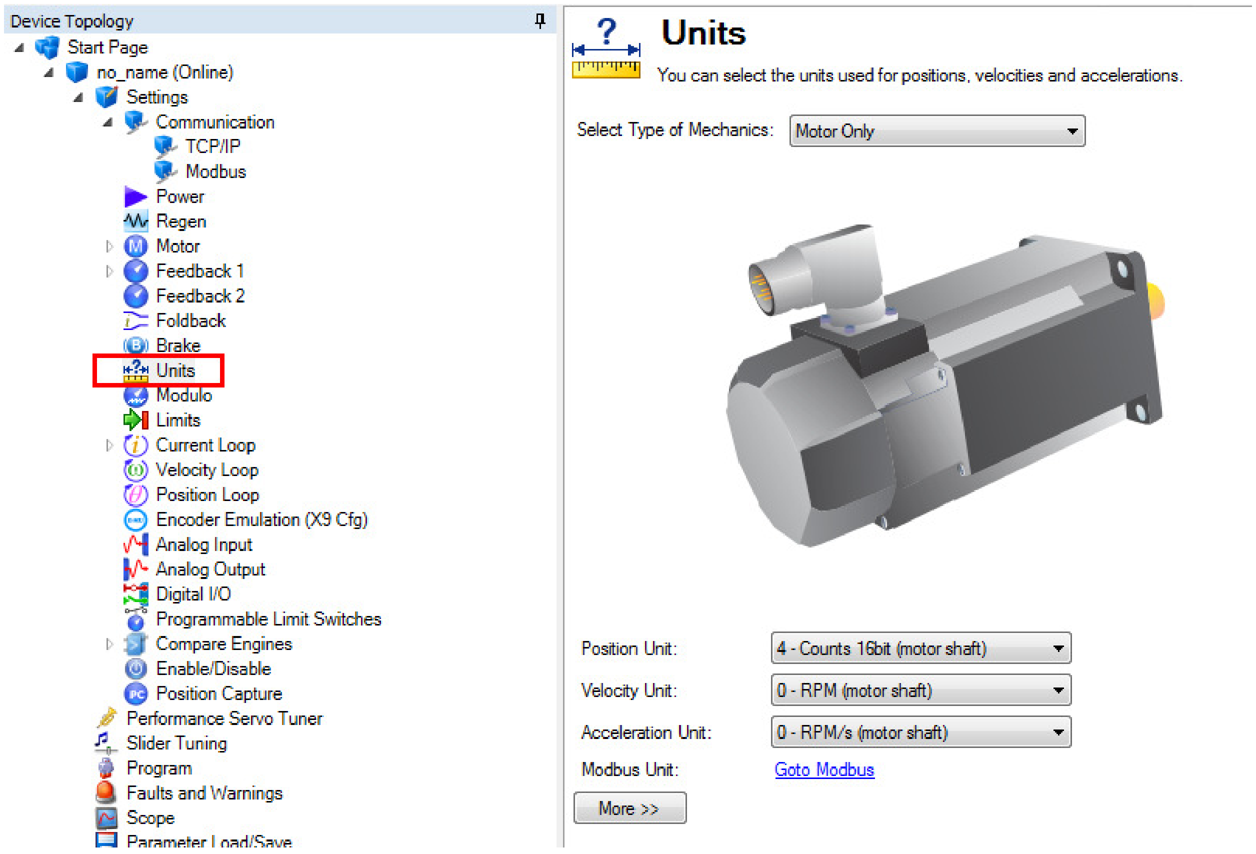
Modbus

Configures the scaling Modbus properties.

Type of scaling: 0 - Drive Internal

Use position unit from the device, velocity and acceleration will be derivatives of this unit (unit/s, unit/s^2).

Next I setup the units for the bench test. This assumes the motor is free to turn and not connected to any load. Although the units can be set for inches, mm, deg, etc. for this test I left the Units to the default (Motor Only).



You can select the units used for positions, velocities and accelerations.

Select Type of Mechanics: Motor Only

Position Unit: 4 - Counts 16bit (motor shaft)

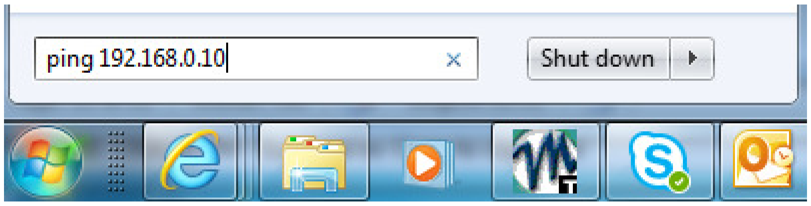
Velocity Unit: 0 - RPM (motor shaft)

Acceleration Unit: 0 - RPM/s (motor shaft)

Modbus Unit: Goto Modbus

It is always a good idea when establishing communications to ping your PC, the PLC, and the AKD drive to make sure everyone replies.

This is done in search programs and files box in Windows 7 when you click on the “Start” icon.

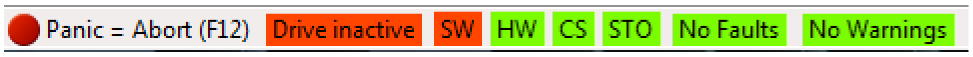


```
ping 192.168.0.10
```

I repeated the method and got replies for all 3 IP Addresses.

To prove the PLC is communicating to the drive, the AKD drive should be healthy and ready to enable with the exception of the softare enable. This application note assumes you've already gone through the process of wiring and basic drive/motor setup per the AKD Installation Manual, Quick Start, Workbench Help, and other support documentation.

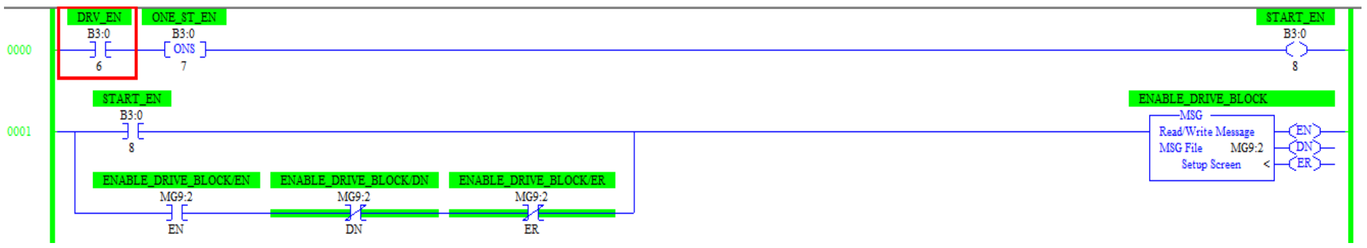
From the bottom status bar of Workbench:



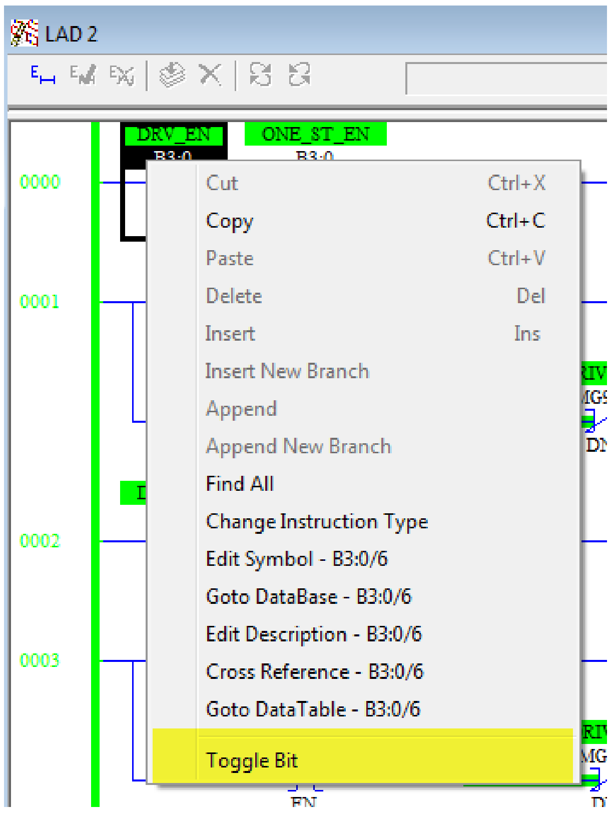
Since the enable/disable from the PLC is independent of the AKD BASIC sample program we can prove PLC to drive communications using the Enable and Disable MSG blocks.

Now we're ready to enable the drive via the Micrologix 1400 over Modbus TCP.

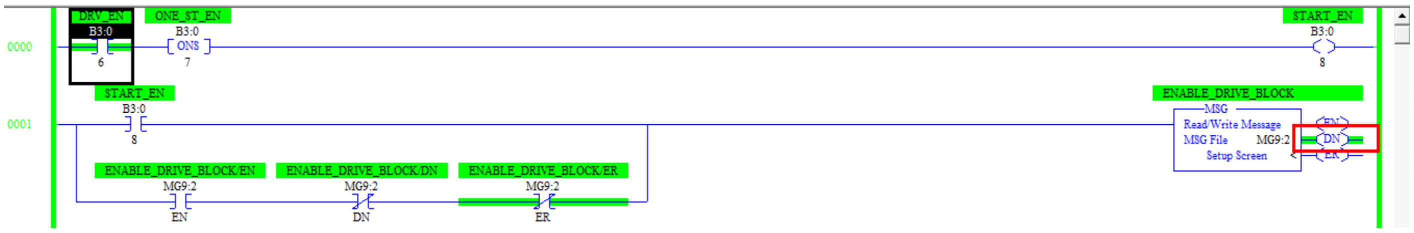
The methodology used in the ladder for triggering the MSG blcoks will be explained in greater detail but first right click on the DRV_EN contact in rung 0.



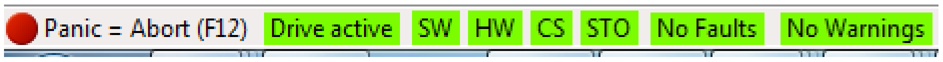
From the choices, select "Toggle Bit" to trigger the logic (note the drive should enable).

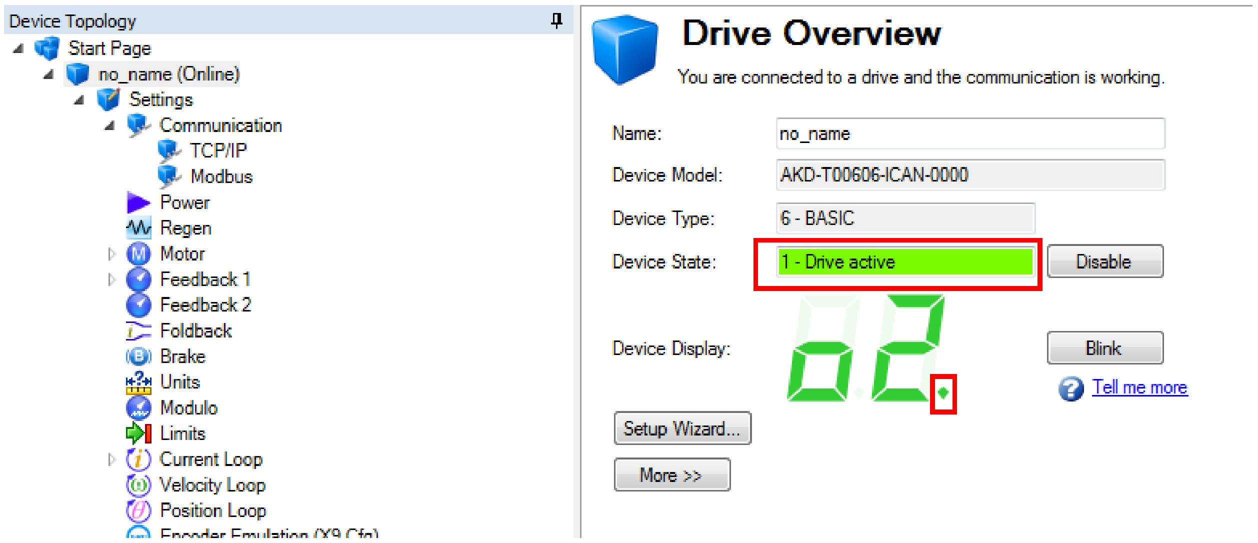


If the MSG block executes without error the DN (done) bit should turn on, on the output of the MSG block in rung 1.



From the Workbench software:



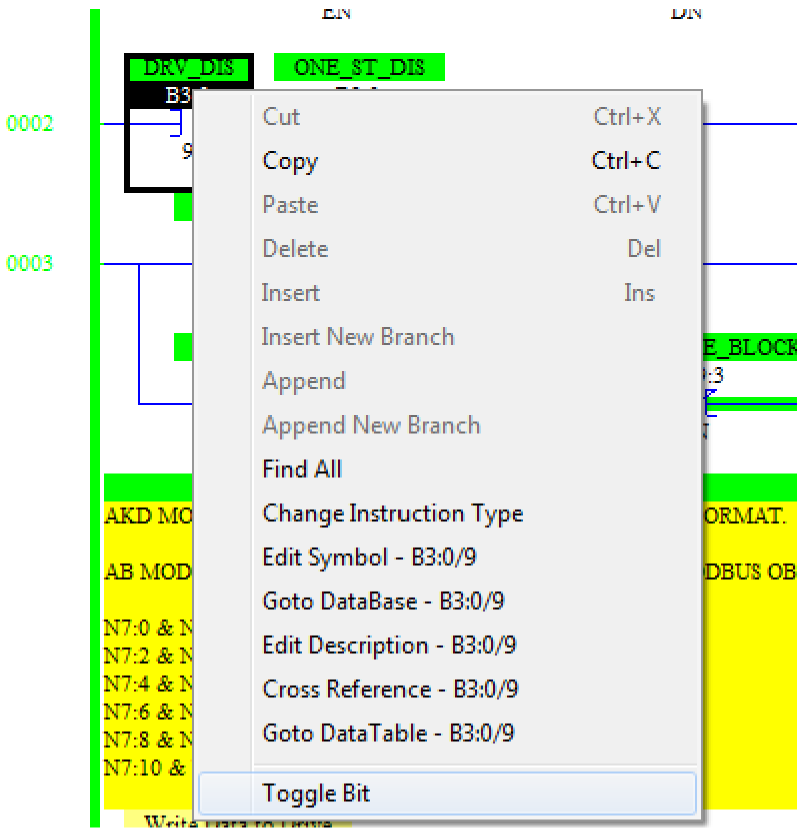


You are connected to a drive and the communication is working.

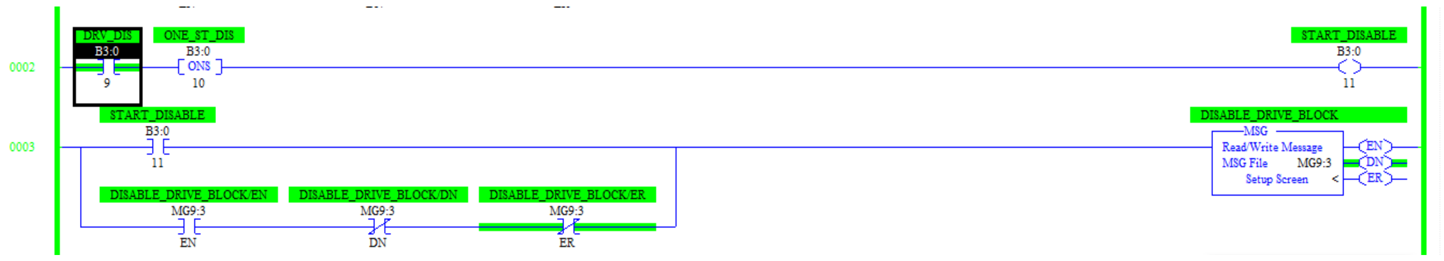
Name: no_name
Device Model: AKD-T00606-ICAN-0000
Device Type: 6 - BASIC
Device State: 1 - Drive active
Device Display:

Next the sample program shows how to command to software disable the drive.

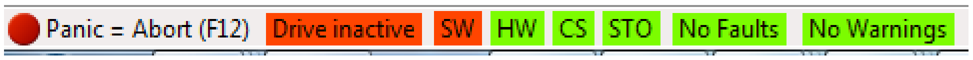
Using the same convention as before, on rung 2 right click on the DRV_DIS contact and select “Toggle Bit” to trigger the disable MSG block.



The DN (done) bit on the output of the DISABLE_DRIVE_BLOCK MSG in rung 3 should turn on indicating the MSG block successfully executed (the drive should be disabled).

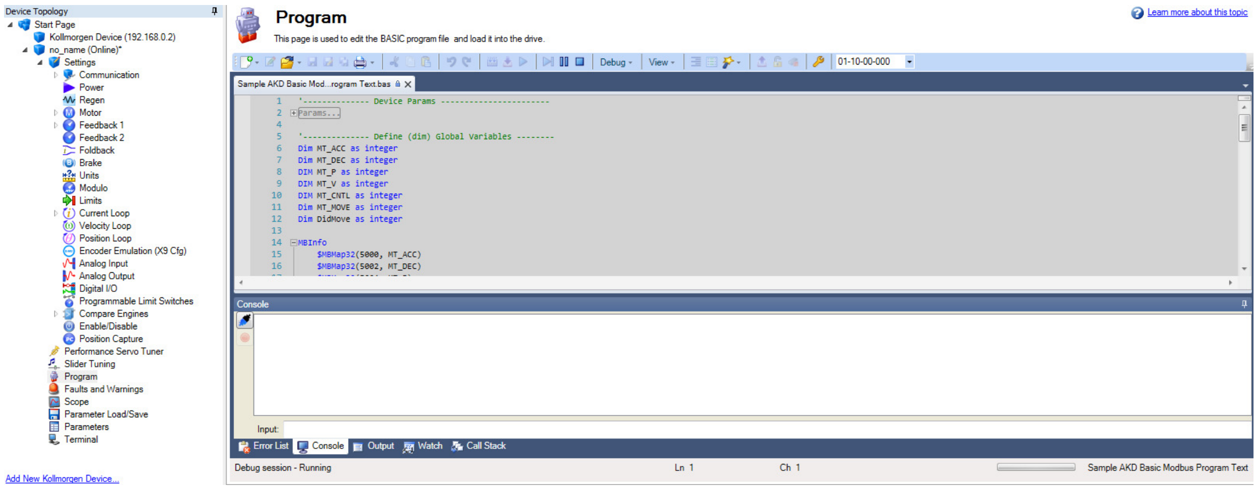


From Workbench bottom status bar, the Drive is Inactive and the SW is OFF.



Next the Sample Program will demonstrate writing data to Modbus Registers.
At this point before proceeding, load the AKD BASIC sample program into the AKD BASIC drive and also perform the dynamic mapping.

After opening the sample program, compiling, downloading, and running you can see the status bar at the bottom of the Program screen in Workbench that shows the name of the loaded program and the status of “Running”.



Program

This page is used to edit the BASIC program file and load it into the drive.

Sample AKD Basic Mod...rogram Text.bas

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

'------------- Define (dim) Global Variables --------
Dim MT_ACC as integer
Dim MT_DEC as integer
DIM MT_P as integer
DIM MT_V as integer
DIM MT_CNTL as integer
Dim MT_MOVE as integer
Dim DidMove as integer

MBInfo
    $MBMap32(5000, MT_ACC)
    $MBMap32(5002, MT_DEC)
```

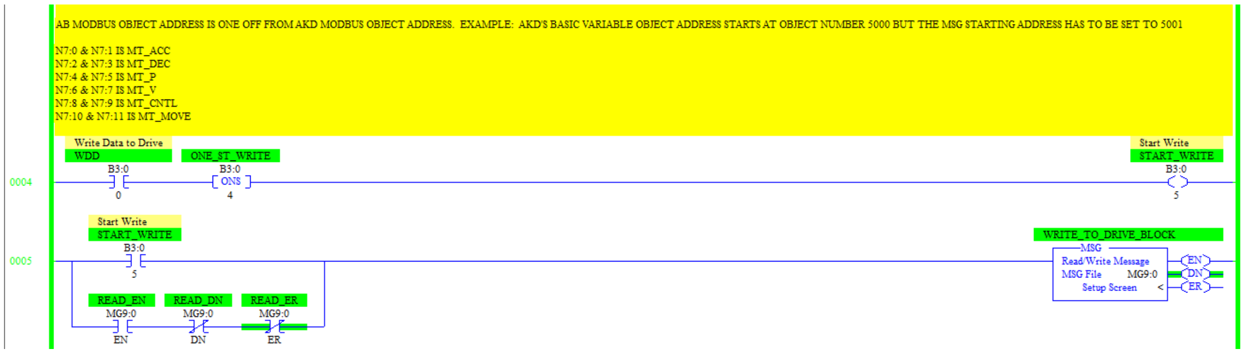
Debug session - Running   Ln 1   Ch 1   Sample AKD Basic Modbus Program Text

The sample project makes use of Modbus Dynamic Mapping (more detail on this later).

Either via Workbench Terminal by typing each line one by one or by putting the entire mapping into a Macro and running the macro (also in the Workbench Terminal screen), drive parameters used for this sample project are dynamically mapped.

```
MODBUS.DYNMAP 1

MODBUS.CLRDYNMAP

MODBUS.ADDR8192 2072

MODBUS.ADDR8193 2073

MODBUS.ADDR8194 856

MODBUS.ADDR8195 857

MODBUS.ADDR8196 220

MODBUS.ADDR8197 221

MODBUS.ADDR8198 240

MODBUS.ADDR8199 241

MODBUS.ADDR8200 954

MODBUS.ADDR8201 955

MODBUS.ADDR8202 956

MODBUS.ADDR8203 957

MODBUS.ADDR8204 958

MODBUS.ADDR8205 959

MODBUS.DYNMAP 0
```

Modbus Writes



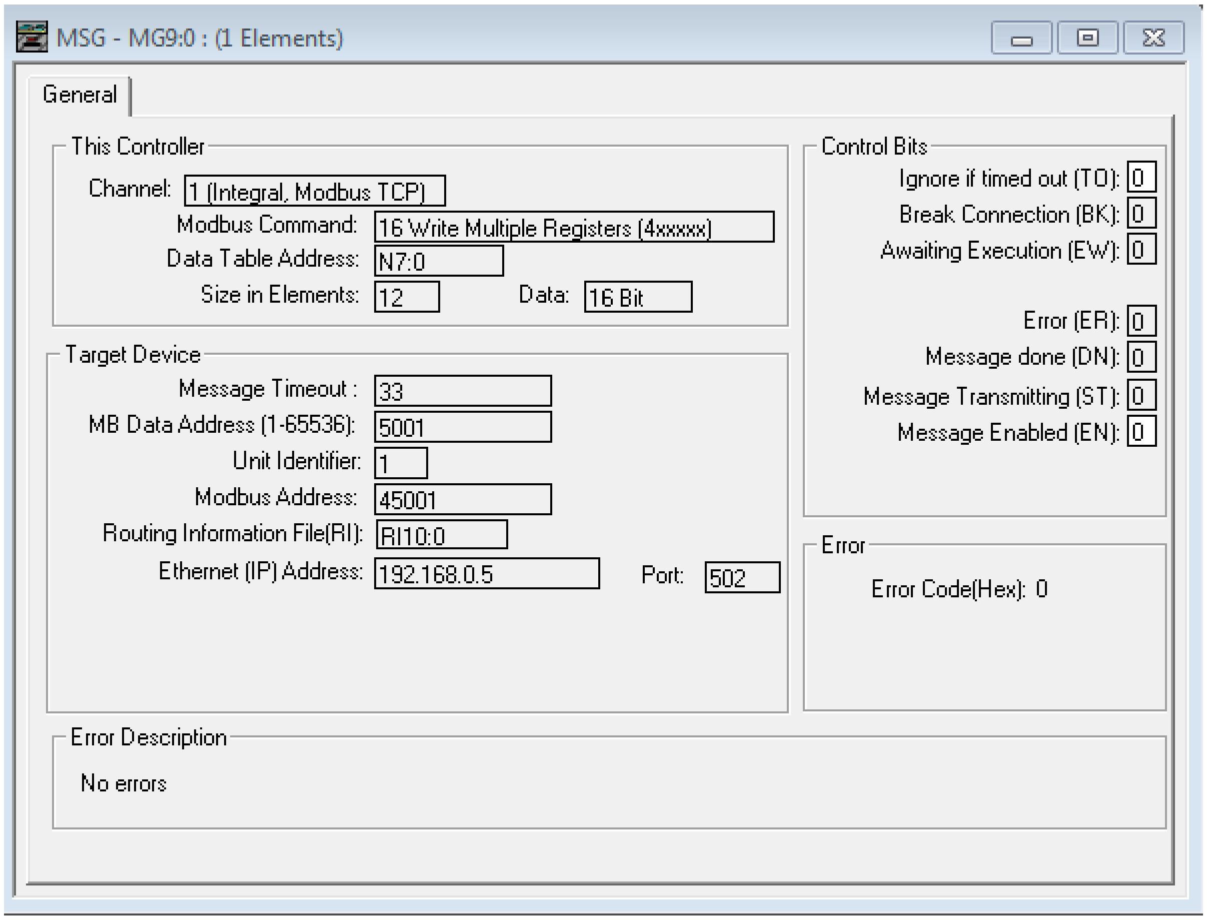
AB MODBUS OBJECT ADDRESS IS ONE OFF FROM AKD MODBUS OBJECT ADDRESS. EXAMPLE: AKD'S BASIC VARIABLE OBJECT ADDRESS STARTS AT OBJECT NUMBER 5000 BUT THE MSG STARTING ADDRESS HAS TO BE SET TO 5001

N7:0 & N7:1 IS MT_ACC
N7:2 & N7:3 IS MT_DEC
N7:4 & N7:5 IS MT_P
N7:6 & N7:7 IS MT_V
N7:8 & N7:9 IS MT_CNTL
N7:10 & N7:11 IS MT_MOVE

As indicated before, the sample BASIC program (included in the zip file) will need to be loaded and running in the drive which includes the following Modbus declarations which allows these registers to be valid (exist) otherwise there will be errors and timeouts from the Modbus master. The zip file has the AKD BASIC program names as a *.txt file extension because often the required *.bas file is blocked by email servers, etc. Simply rename the file with the *.bas extension and open, compile,download, and run in the AKD BASIC test drive. Note AKD BASIC variables are mapped with Modbus registers starting at register address 5000 up to 5999.

```
MBInfo

        $MBMap32(5000, MT_ACC)

        $MBMap32(5002, MT_DEC)

        $MBMap32(5004, MT_P)

        $MBMap32(5006, MT_V)

        $MBMap32(5008, MT_CNTL)

        $MBMap32(5010, MT_MOVE)
```

It is important to note that the PLC's addressing of Modbus registers is offset by 1 to the addresing of the AKD drive (i.e. drive address 5000 is PLC's address 5001, drive's 5001 is the PLC's 5002, and so forth). From the MSG block for the writes, note the starting address (MB Data Address) for the Target Device uses the offset and starts at 5001.


MSG - MG9:0 : (1 Elements)

General

This Controller

Channel: 1 (Integral, Modbus TCP)
Modbus Command: 16 Write Multiple Registers (4xxxxx)
Data Table Address: N7:0
Size in Elements: 12   Data: 16 Bit

Target Device

Message Timeout : 33
MB Data Address (1-65536): 5001
Unit Identifier: 1
Modbus Address: 45001
Routing Information File(RI): RI10:0
Ethernet (IP) Address: 192.168.0.5   Port: 502

Control Bits

Ignore if timed out (TO): 0
Break Connection (BK): 0
Awaiting Execution (EW): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error

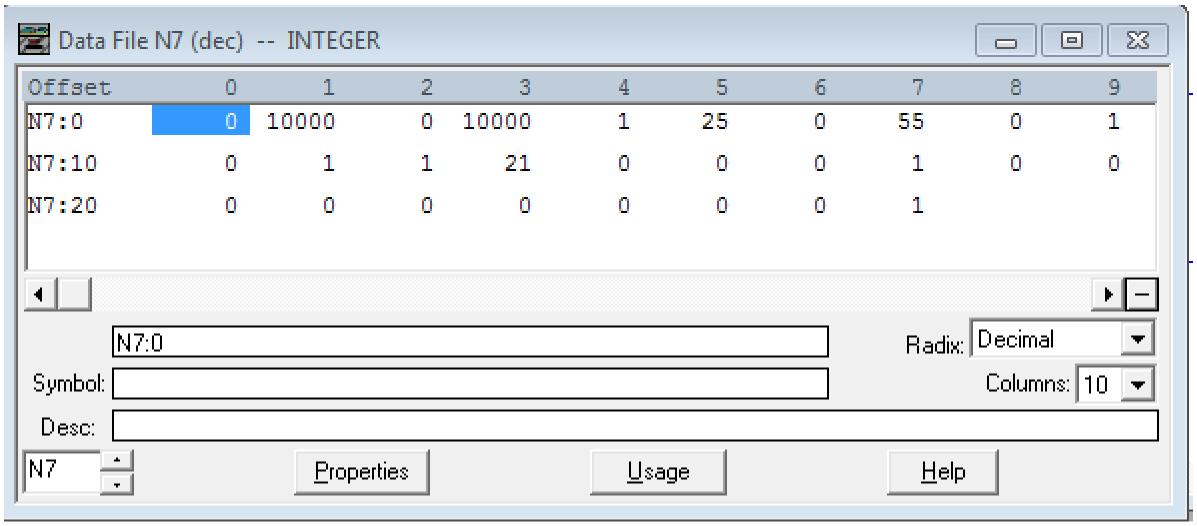
Error Code(Hex): 0

Error Description

No errors

In summary:

AKD BASIC Modbus Address	Variable	Variable Description	PLC Modbus address	PLC Memory Address	PLC value in Sample Program
5000-5001	MT_ACC	Move Accel	5001-5002	N7:0-N7:1	10000
5002-5003	MT_DEC	Move Decel	5003-5004	N7:2-N7:3	10000
5004-5005	MT_P	Move Target Position or Relative Distance	5005-5006	N7:4-N7:5	1 and 25 (translates to 65561 in the 32 bit word)
5006-5007	MT_V	Move Runspeed	5007-5008	N7:6-N7:7	55
5008-5009	MT_CNTL	Selects Absolute (MT_CNTL=0) or Relative Move (MT_CNTL=1)	5009-5010	N7:8-N7:9	1 (Relative Move)
5010-5011	MT_MOVE	Start Move	5011-5012	N7:10-N7:11	1(Start The Move)


Data File N7 (dec) -- INTEGER

Offset	0	1	2	3	4	5	6	7	8	9
N7:0	0	10000	0	10000	1	25	0	55	0	1
N7:10	0	1	1	21	0	0	0	1	0	0
N7:20	0	0	0	0	0	0	0	1		

N7:0

Radix: Decimal

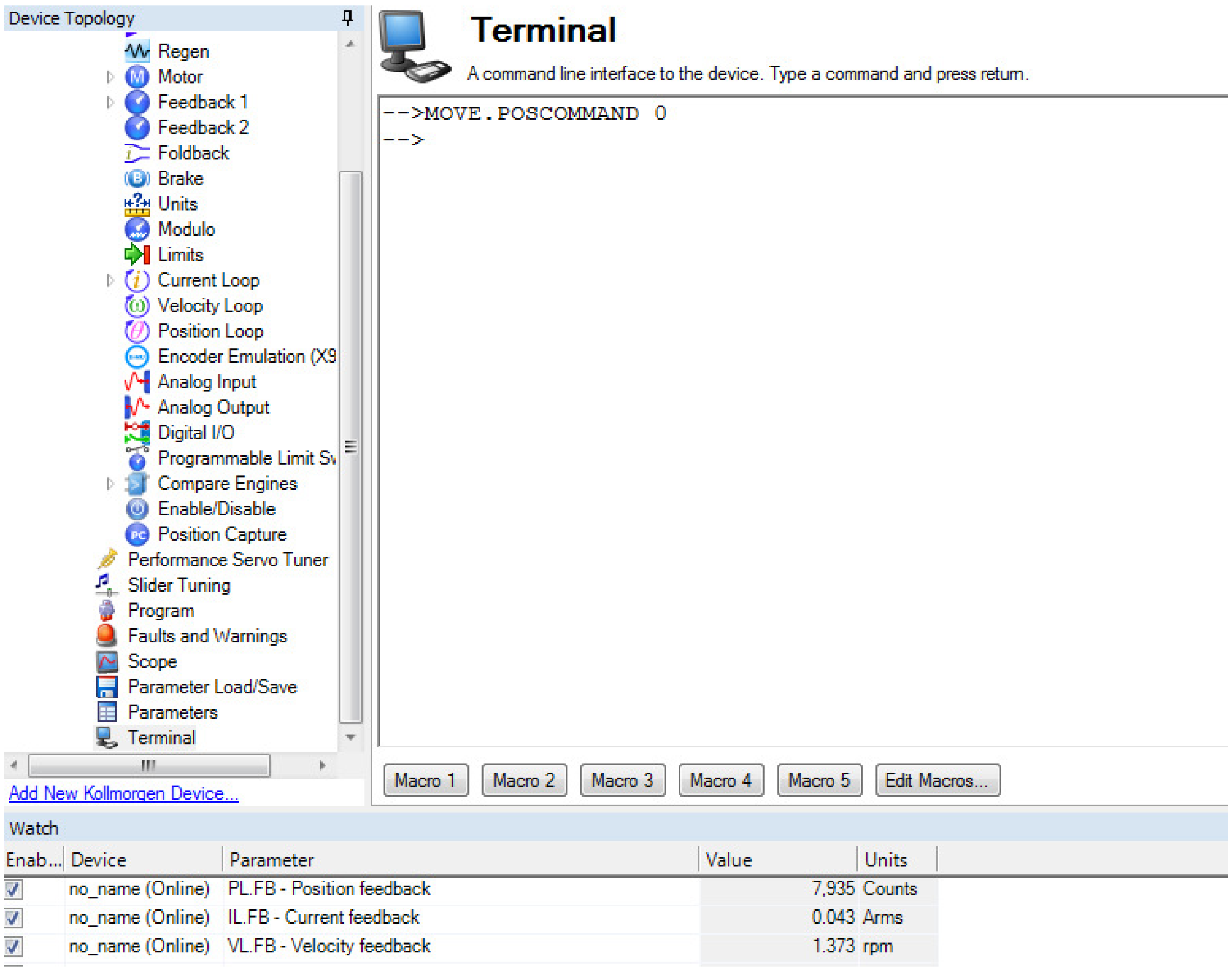
Symbol:

Columns: 10

Desc:

N7

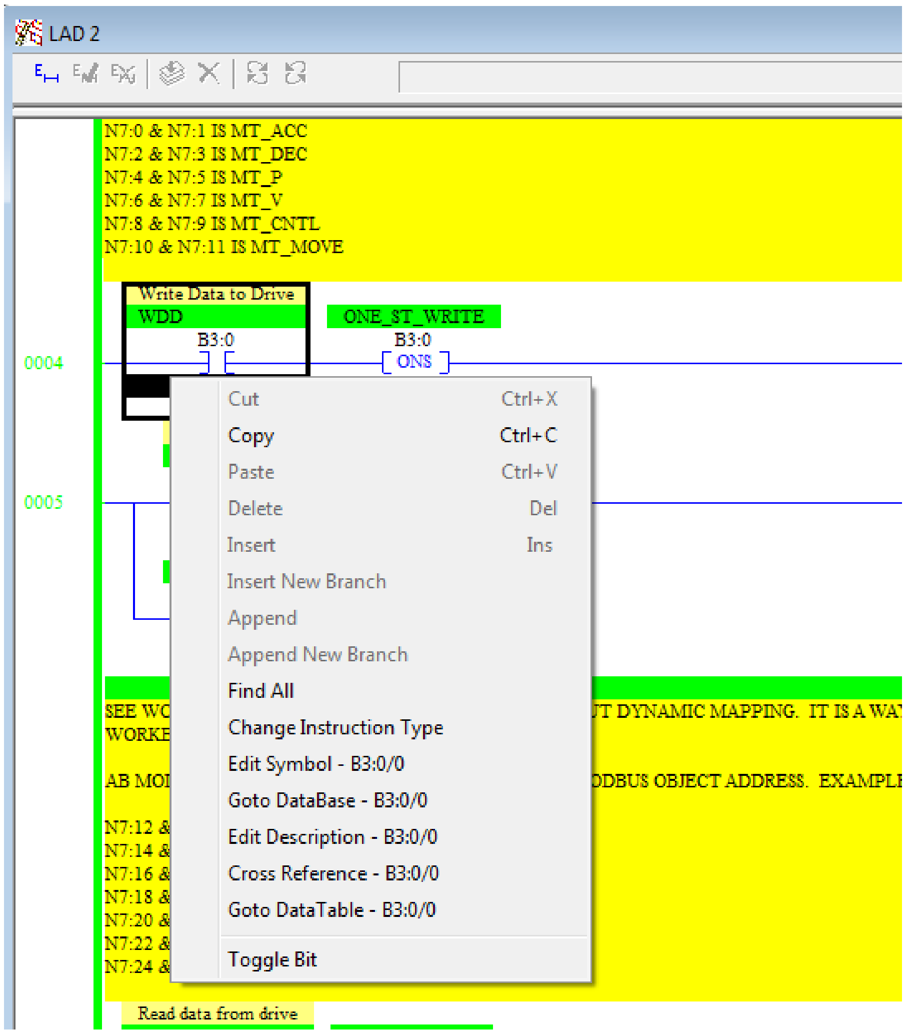
For simplicity, I used the MOVE.POSCOMMAND in the Workbench Terminal to reset the current feedback position to 0.



```
-->MOVE.POSCOMMAND 0
-->
```

Enab...	Device	Parameter	Value	Units
✓	no_name (Online)	PL.FB - Position feedback	7,935	Counts
✓	no_name (Online)	IL.FB - Current feedback	0.043	Arms
✓	no_name (Online)	VL.FB - Velocity feedback	1.373	rpm

To trigger the Writes (and the move), right click on the WDD contact in rung 4 and select “Toggle” bit:


LAD 2

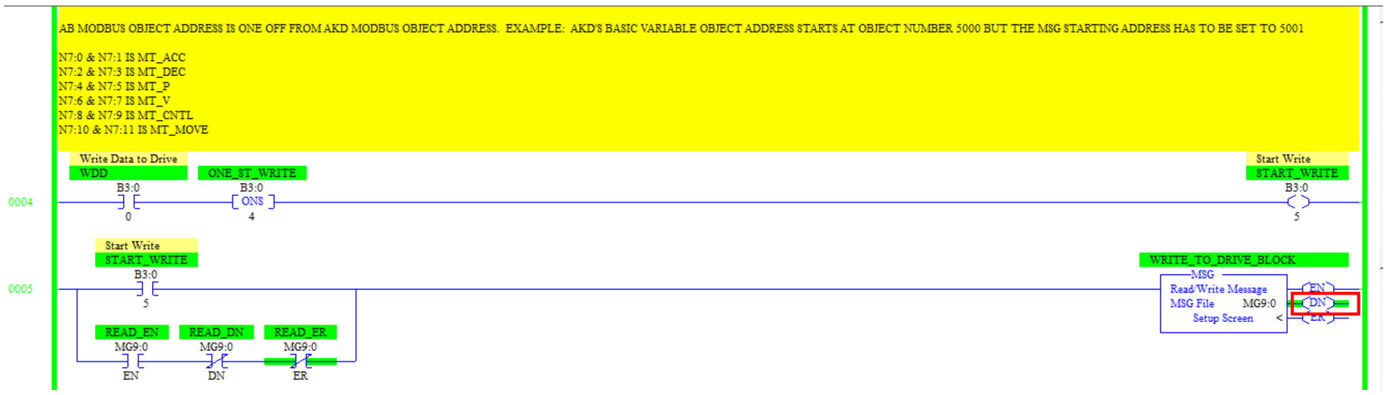
N7:0 & N7:1 IS MT_ACC
N7:2 & N7:3 IS MT_DEC
N7:4 & N7:5 IS MT_P
N7:6 & N7:7 IS MT_V
N7:8 & N7:9 IS MT_CNTL
N7:10 & N7:11 IS MT_MOVE

SEE WO… WORKE… …T DYNAMIC MAPPING. IT IS A WA…

AB MOI… …DBUS OBJECT ADDRESS. EXAMPL…

N7:12 &
N7:14 &
N7:16 &
N7:18 &
N7:20 &
N7:22 &
N7:24 &

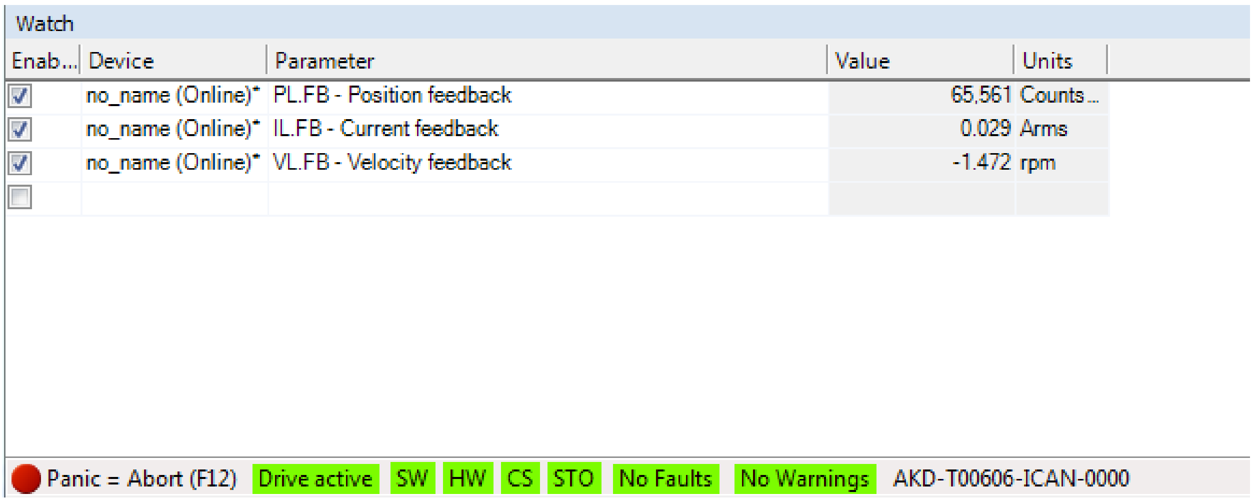
The DN (Done) bit should turn on indicating the write was successful.



AB MODBUS OBJECT ADDRESS IS ONE OFF FROM AKD MODBUS OBJECT ADDRESS. EXAMPLE: AKD'S BASIC VARIABLE OBJECT ADDRESS STARTS AT OBJECT NUMBER 5000 BUT THE MSG STARTING ADDRESS HAS TO BE SET TO 5001

N7:0 & N7:1 IS MT_ACC
N7:2 & N7:3 IS MT_DEC
N7:4 & N7:5 IS MT_P
N7:6 & N7:7 IS MT_V
N7:8 & N7:9 IS MT_CNTL
N7:10 & N7:11 IS MT_MOVE

From the Workbench Watch window, the relative distance of 65561 was reached.



Watch

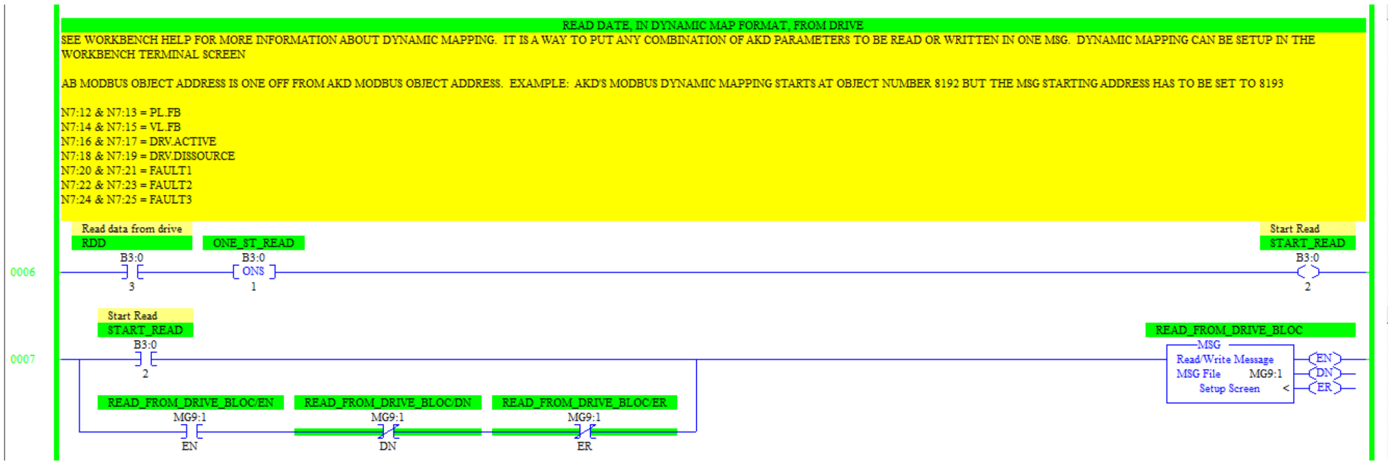
Enab...	Device	Parameter	Value	Units
☑	no_name (Online)*	PL.FB - Position feedback	65,561	Counts...
☑	no_name (Online)*	IL.FB - Current feedback	0.029	Arms
☑	no_name (Online)*	VL.FB - Velocity feedback	-1.472	rpm
☐				

You can keep triggering the write by toggling the contact on and off to index multiples of 65561 counts.

To perform absolute positioning set N7:9=0 and then set the desired absolute position in registers N7:4 and N7:5. For example I set N7:4=10 and N7:5=0. This is equivalent to 65560 (32 bit word). On retrigger the drive should move to the new position.

Modbus Reads

Next the Sample Program demonstrates reading data from Modbus Registers.



READ DATE, IN DYNAMIC MAP FORMAT, FROM DRIVE

SEE WORKBENCH HELP FOR MORE INFORMATION ABOUT DYNAMIC MAPPING. IT IS A WAY TO PUT ANY COMBINATION OF AKD PARAMETERS TO BE READ OR WRITTEN IN ONE MSG. DYNAMIC MAPPING CAN BE SETUP IN THE WORKBENCH TERMINAL SCREEN

AB MODBUS OBJECT ADDRESS IS ONE OFF FROM AKD MODBUS OBJECT ADDRESS. EXAMPLE: AKD'S MODBUS DYNAMIC MAPPING STARTS AT OBJECT NUMBER 8192 BUT THE MSG STARTING ADDRESS HAS TO BE SET TO 8193

N7:12 & N7:13 = PL.FB
N7:14 & N7:15 = VL.FB
N7:16 & N7:17 = DRV.ACTIVE
N7:18 & N7:19 = DRV.DISSOURCE
N7:20 & N7:21 = FAULT1
N7:22 & N7:23 = FAULT2
N7:24 & N7:25 = FAULT3

The Sample Program makes use of dynamic mapping to put the desired AKD Parameters' Modbus Addresses in sequential order to accomplish a block read. Recall the dynamic mapping previously covered but repeated here.

The following must be mapped using Workbench terminal. Note all of the following can be copied and pasted into a Macro in Workbench Terminal. The Macro can be run in order to execute the mapping with one button.

MODBUS.DYNMAP 1

MODBUS.CLRDYNMAP

MODBUS.ADDR8192 2072

MODBUS.ADDR8193 2073

MODBUS.ADDR8194 856

MODBUS.ADDR8195 857

MODBUS.ADDR8196 220

MODBUS.ADDR8197 221

MODBUS.ADDR8198 240

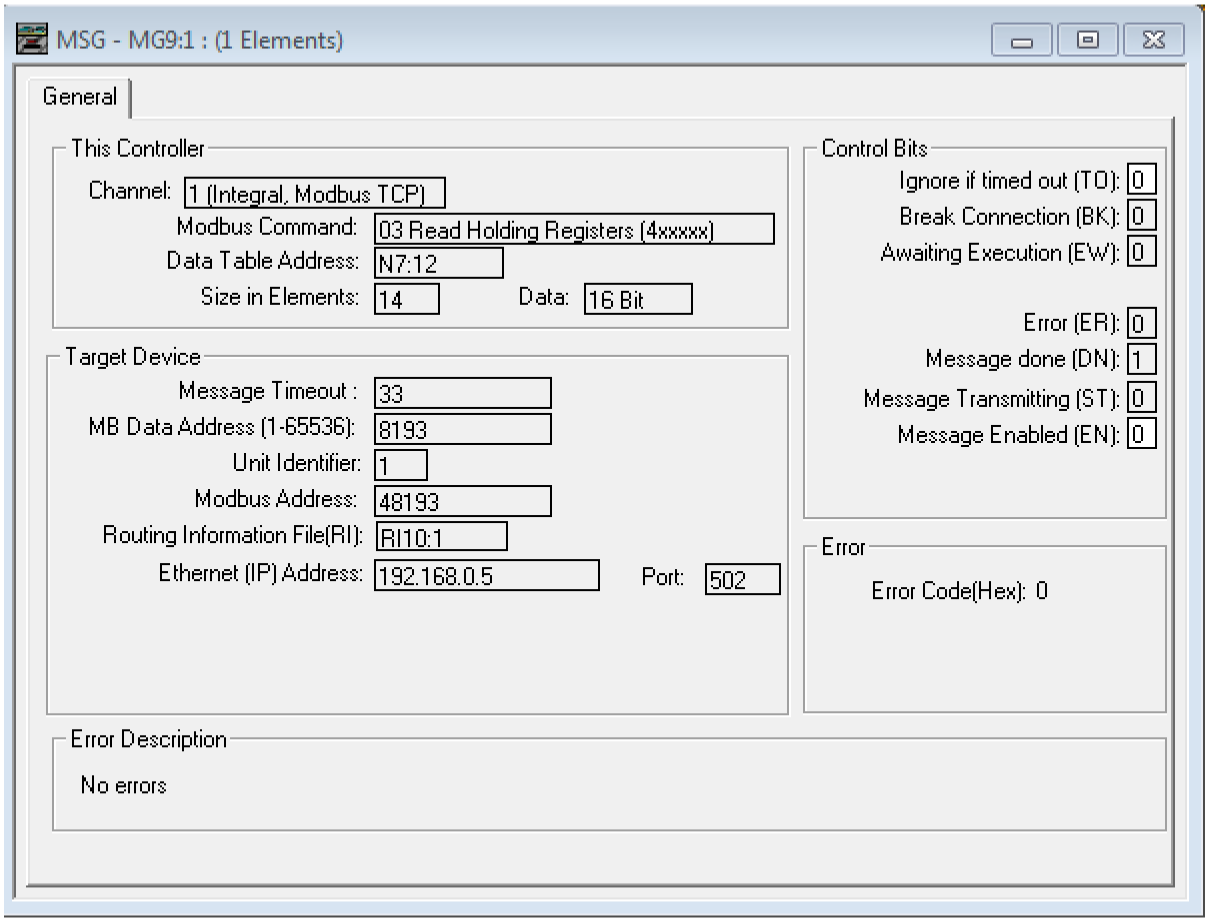
MODBUS.ADDR8199 241

MODBUS.ADDR8200 954

```
MODBUS.ADDR8201 955

MODBUS.ADDR8202 956

MODBUS.ADDR8203 957

MODBUS.ADDR8204 958

MODBUS.ADDR8205 959

MODBUS.DYNMAP 0
```

In summary:

Dynamic Address	Standard Address	Parameter	Parameter Description	PLC Modbus address	PLC Memory Address
8192-8193	2072-2073	PL.FB_32	Position Feedback	8193-8194	N7:12-N7:13
8194-8195	856-857	VL.FB	Velocity Feedback	8195-8196	N7:14-N7:15
8196-8197	220-221	DRV.ACTIVE	Drive Active	8197-8198	N7:16-N7:17
8198-8199	240-241	DRV.DISSOURCES	Possible Reasons For A Drive Disable	8199-8200	N7:18-N7:19
8200-8201	954-955	DRV.FAULT1	Current Drive Fault 1	8201-8202	N7:20-N7:21
8202-8203	956-957	DRV.FAULT2	Current Drive Fault 2	8203-8305	N7:22-N7:23
8204-8305	958-959	DRV.FAULT3	Current Drive Fault 3	8205-8206	N7:24-N7:25



MSG - MG9:1 : (1 Elements)

General

This Controller

Channel:	1 (Integral, Modbus TCP)
Modbus Command:	03 Read Holding Registers (4xxxxx)
Data Table Address:	N7:12
Size in Elements:	14
Data:	16 Bit

Target Device

Message Timeout :	33
MB Data Address (1-65536):	8193
Unit Identifier:	1
Modbus Address:	48193
Routing Information File(RI):	RI10:1
Ethernet (IP) Address:	192.168.0.5
Port:	502

Control Bits

Ignore if timed out (TO):	0
Break Connection (BK):	0
Awaiting Execution (EW):	0
Error (ER):	0
Message done (DN):	1
Message Transmitting (ST):	0
Message Enabled (EN):	0

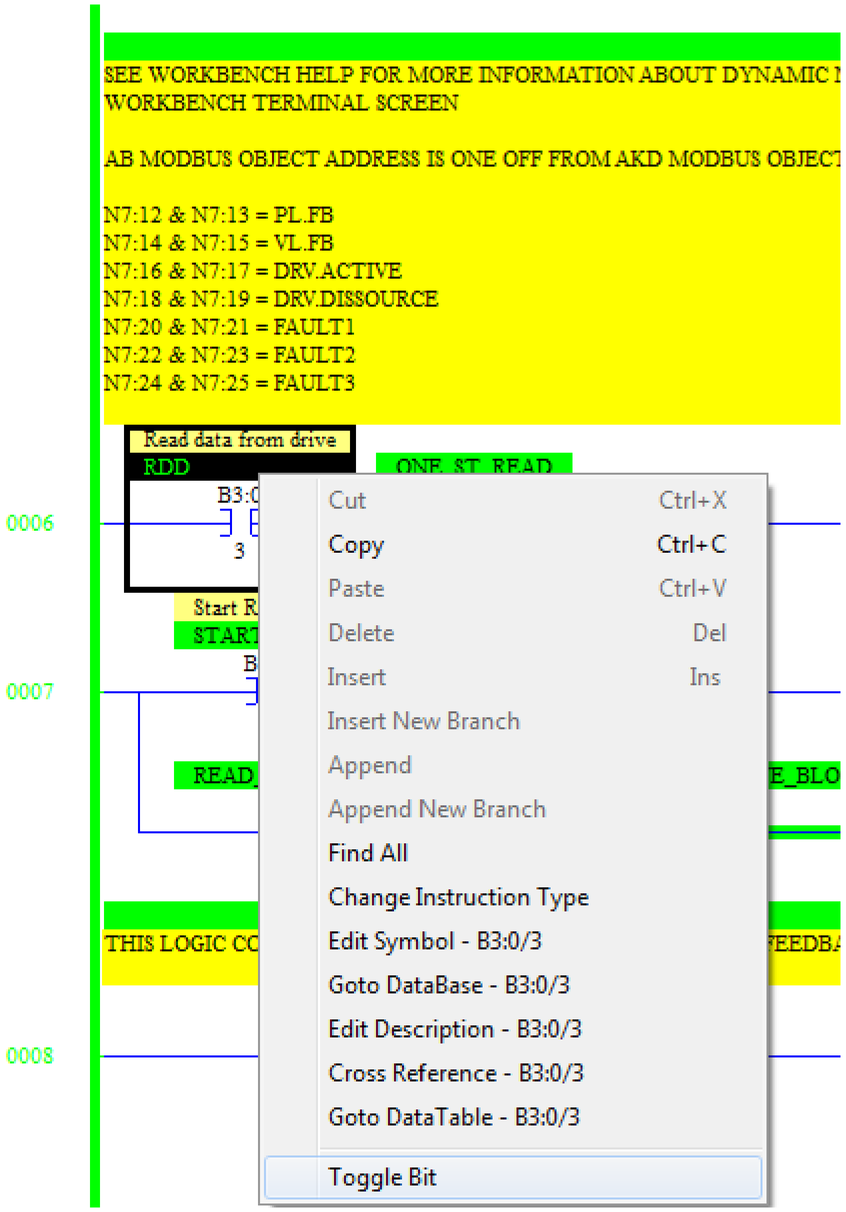
Error

Error Code(Hex): 0

Error Description

No errors

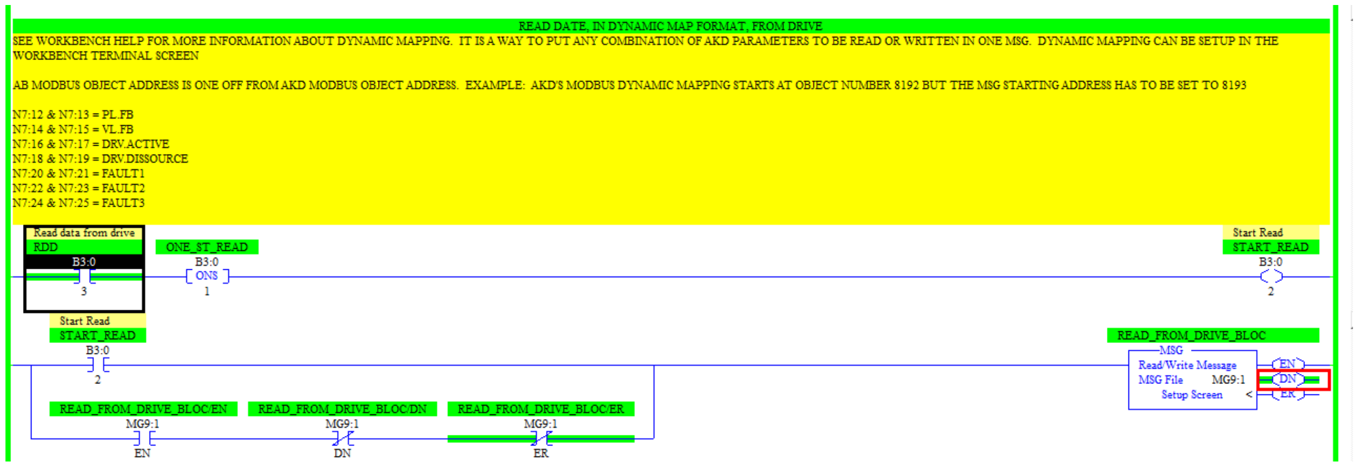
To perform the block read, right click on the contact "RDD" in rung 6 and select "Toggle Bit".


SEE WORKBENCH HELP FOR MORE INFORMATION ABOUT DYNAMIC I
WORKBENCH TERMINAL SCREEN

AB MODBUS OBJECT ADDRESS IS ONE OFF FROM AKD MODBUS OBJECT

N7:12 & N7:13 = PL.FB
N7:14 & N7:15 = VL.FB
N7:16 & N7:17 = DRV.ACTIVE
N7:18 & N7:19 = DRV.DISSOURCE
N7:20 & N7:21 = FAULT1
N7:22 & N7:23 = FAULT2
N7:24 & N7:25 = FAULT3

The DN (Done) bit on the output of the MSG block should turn on indicating it executed successfullly without error.



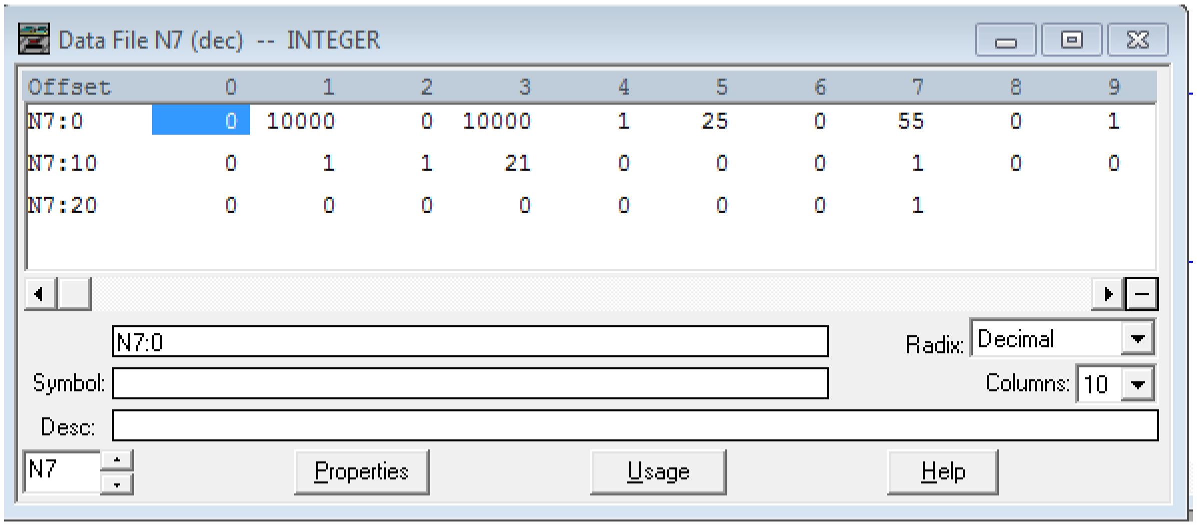
READ DATE, IN DYNAMIC MAP FORMAT, FROM DRIVE

SEE WORKBENCH HELP FOR MORE INFORMATION ABOUT DYNAMIC MAPPING. IT IS A WAY TO PUT ANY COMBINATION OF AKD PARAMETERS TO BE READ OR WRITTEN IN ONE MSG. DYNAMIC MAPPING CAN BE SETUP IN THE WORKBENCH TERMINAL SCREEN

AB MODBUS OBJECT ADDRESS IS ONE OFF FROM AKD MODBUS OBJECT ADDRESS. EXAMPLE: AKD'S MODBUS DYNAMIC MAPPING STARTS AT OBJECT NUMBER 8192 BUT THE MSG STARTING ADDRESS HAS TO BE SET TO 8193

N7:12 & N7:13 = PL.FB
N7:14 & N7:15 = VL.FB
N7:16 & N7:17 = DRV.ACTIVE
N7:18 & N7:19 = DRV.DISSOURCE
N7:20 & N7:21 = FAULT1
N7:22 & N7:23 = FAULT2
N7:24 & N7:25 = FAULT3

The values read can be verified online:



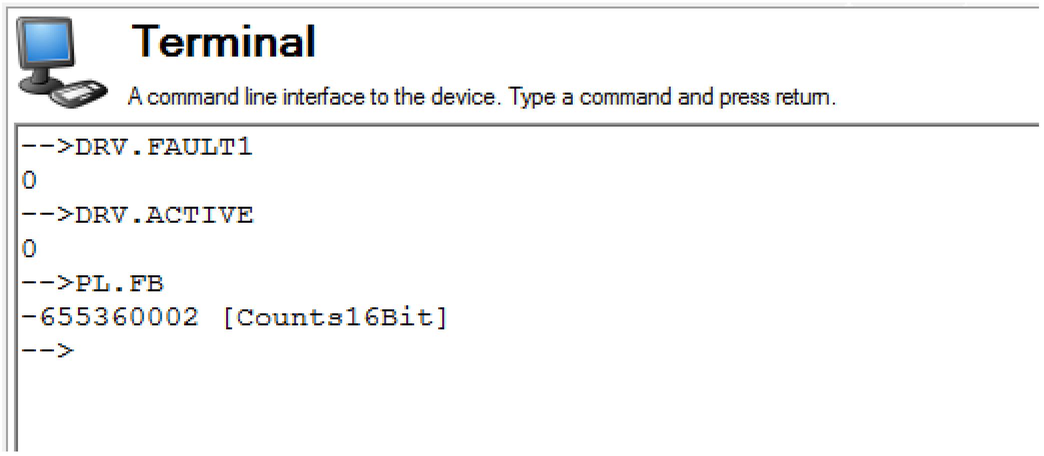
Data File N7 (dec) -- INTEGER

Offset	0	1	2	3	4	5	6	7	8	9
N7:0	0	10000	0	10000	1	25	0	55	0	1
N7:10	0	1	1	21	0	0	0	1	0	0
N7:20	0	0	0	0	0	0	0	1		

N7:0
Radix: Decimal
Symbol:
Columns: 10
Desc:
N7

You can compare the value in the window above with values queried in Workbench Terminal.

For example:



Terminal

A command line interface to the device. Type a command and press return.

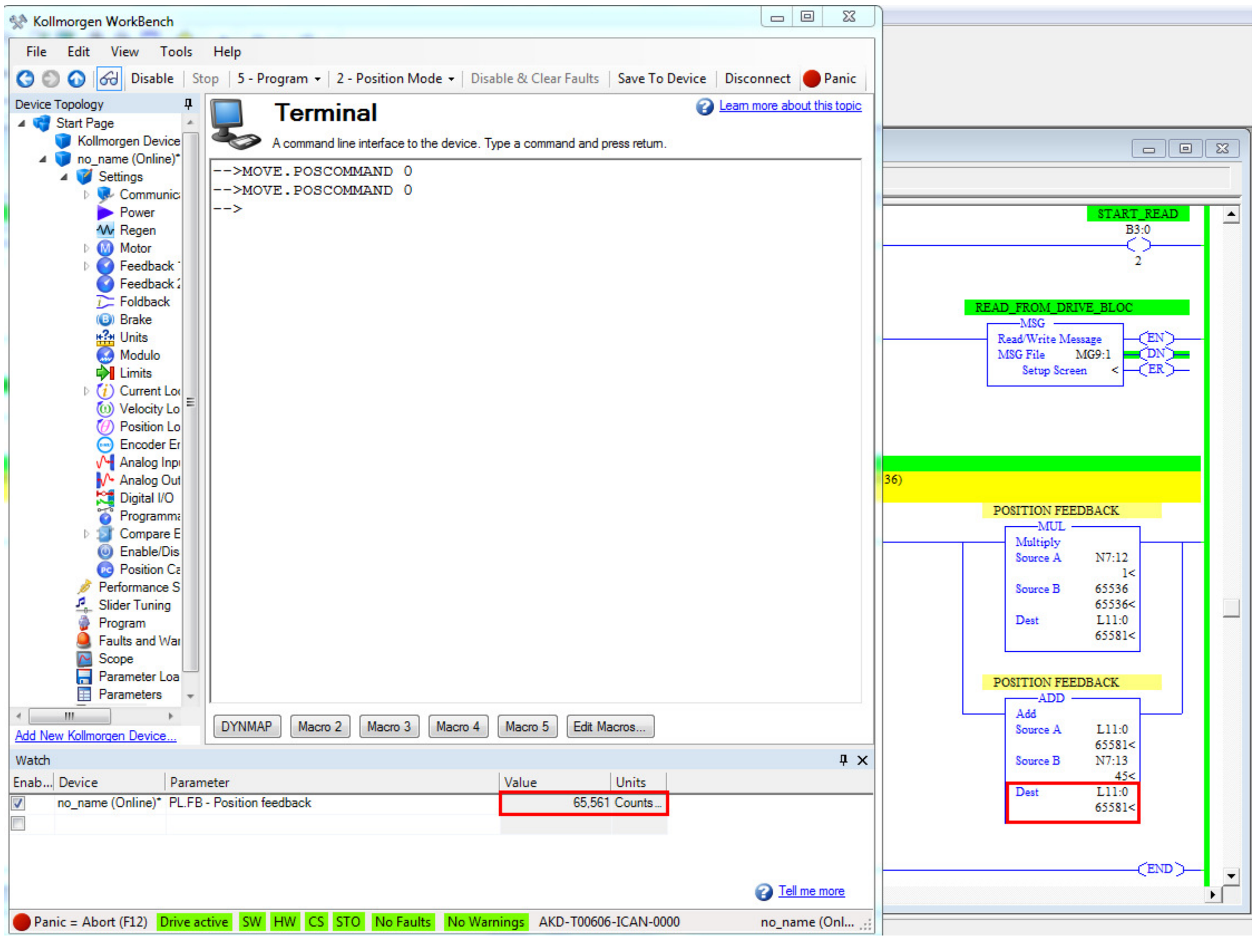
```
-->DRV.FAULT1
0
-->DRV.ACTIVE
0
-->PL.FB
-655360002 [Counts16Bit]
-->
```

Reading Actual Feedback Position using AB Micrologix 1400 Modbus TCP

From the previous the dynamically mapped parameters should have been updated in the PLC.

Since N7 registers in the PLC are signed 16 bit registers and can only hold -32768 to +32767 the position feedback value that can be contained is limited. One method which is shown in the sample project is to multiply the high word by 65536 and add the low word to the result and store the value in a L11 register which can hold a signed 32 bit value (-2, 147, 483, 648 to + 2, 147, 483, 647).

After triggering the read as described previously, I compared the value in the Workbench watch with the L11:0 value in rung 8 of the sample project and the 2 values matched.


Kollmorgen WorkBench
File Edit View Tools Help
Disable Stop 5 - Program 2 - Position Mode Disable & Clear Faults Save To Device Disconnect Panic
Device Topology
Start Page
Kollmorgen Device
no_name (Online)*
Settings
Communic
Power
Regen
Motor
Feedback
Feedback
Foldback
Brake
Units
Modulo
Limits
Current Lo
Velocity Lo
Position Lo
Encoder Er
Analog Inp
Analog Out
Digital I/O
Programm
Compare E
Enable/Dis
Position Ca
Performance S
Slider Tuning
Program
Faults and Wa
Scope
Parameter Loa
Parameters
Add New Kollmorgen Device...
Terminal
Learn more about this topic
A command line interface to the device. Type a command and press return.
DYNMAP Macro 2 Macro 3 Macro 4 Macro 5 Edit Macros...
Tell me more
Panic = Abort (F12) Drive active SW HW CS STO No Faults No Warnings AKD-T00606-ICAN-0000 no_name (Onl...

```
-->MOVE.POSCOMMAND 0
-->MOVE.POSCOMMAND 0
-->
```

Watch

Enab...	Device	Parameter	Value	Units
✓	no_name (Online)*	PL.FB - Position feedback	65,561	Counts...

START_READ B3:0 2

READ_FROM_DRIVE_BLOC
MSG Read/Write Message MSG File MG9:1 Setup Screen (EN) (DN) (ER)

36)

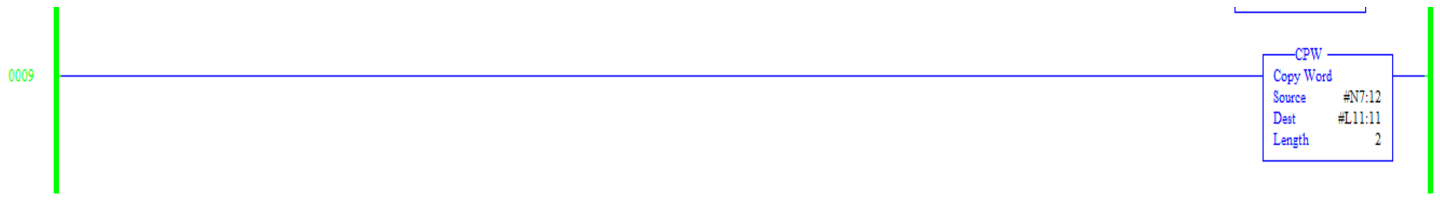
POSITION FEEDBACK
MUL Multiply
Source A N7:12 1<
Source B 65536 65536<
Dest L11:0 65581<

POSITION FEEDBACK
ADD Add
Source A L11:0 65581<
Source B N7:13 45<
Dest L11:0 65581<

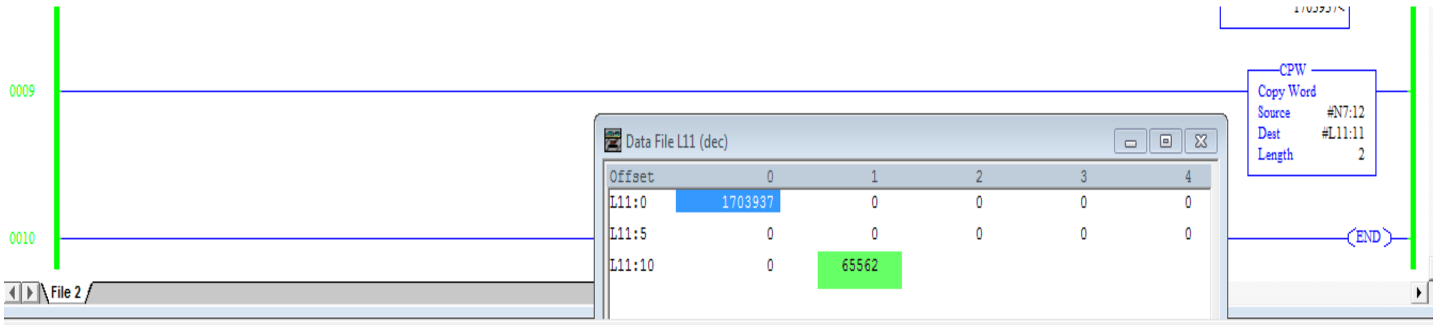
(END)

An alternate method is to use the CPW (Copy Word) instruction. The Copy Word Instruction can be used to copy the 2 consecutive 16 bit integers (N7) into a long 32 bit integer (L11). The CPW instruction copies the 2 16 bit integers in ascending order (i.e. N7:12 is copied to the low word and N7:13 is copied high word of the L11 register), therefore unlike the method above, it is required the dynamic mapping swap the word order in order for the CPW data to appear correctly in the PLC. Note this code is not included in the Sample Project but for demonstrational purposes.

```
MODBUS.DYNMAP 1
MODBUS.ADDR8192 2073
MODBUS.ADDR8193 2072
MODBUS.DYNMAP 0
```



On retrigger of the read, the value in L11:11 matched what was in the AKD's PL.FB.



Data File L11 (dec)

Offset	0	1	2	3	4
L11:0	1703937	0	0	0	0
L11:5	0	0	0	0	0
L11:10	0	65562			