

Setup for Decimal Data Entry for AKI (KVB2.0) to AKD-T (BASIC)

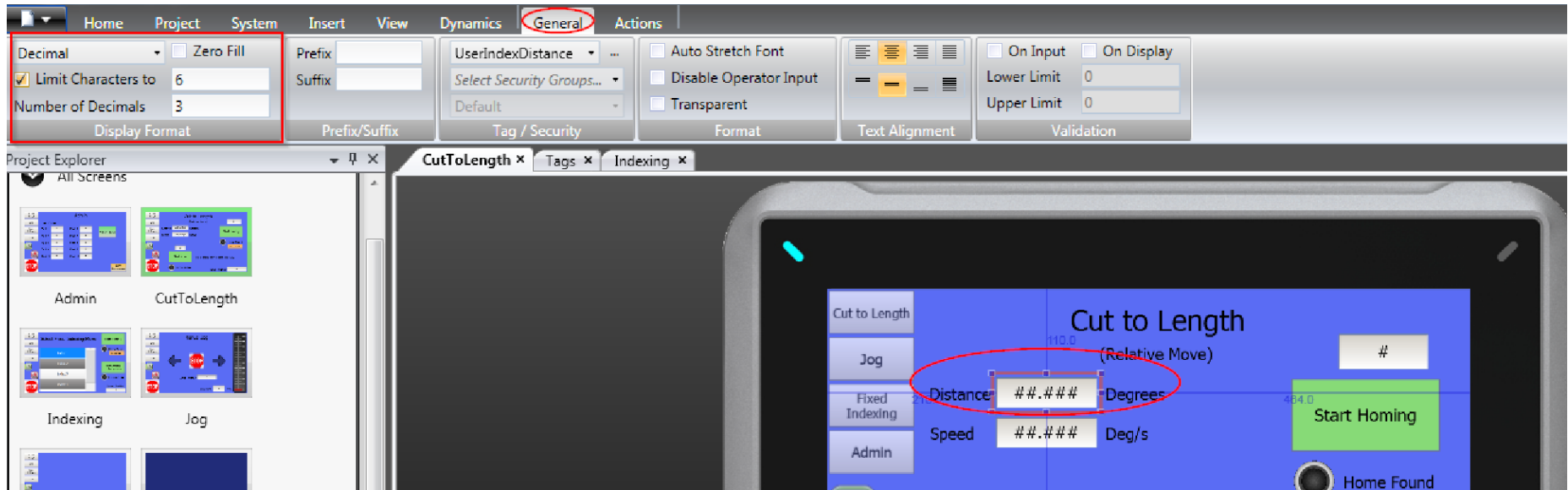
Highlighted below are the 2 tags created for Distance and Speed.

In order for the data entry/display on the AKI HMI to be set to “Decimal” the tags are declared as “FLOAT” but the Controller Data Type remains “INT32”.

For each tag Scaling was used with a gain set to “0.001” for 3 decimal places. For higher resolution, “0.0001” would be required for example.

Tag		Controllers			Others							Scaling	
Name	Data Type	Access Right	Data Type	AKD 1	Description	Poll Group	Always Active	Non-volatile	Initial Value	Index Register	Action	Offset	Gain
MODBUS_FAULT8	INT32	Read	INT32	40968		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	1
MODBUS_FAULT9	INT32	Read	INT32	40970		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	1
PL_FB32	INT32	ReadWrite	INT32	42072		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	1
UserIndexDistance	FLOAT	ReadWrite	INT32	45000		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	0.001
UserIndexSpeed	FLOAT	ReadWrite	INT32	45002		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	0.001
UserJogSpeed	INT32	ReadWrite	INT32	45004		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	1
UserStartHome	INT32	ReadWrite	INT32	45006		PollGroup1	<input type="checkbox"/>	<input type="checkbox"/>		None		0	1

When the data entry field on the screen development layout is selected and the “General” tab is also selected, the Display Format can be set for the number of decimal places, etc.



The variables to be used in the program of the AKD BASIC drive are declared as integers.

Also note the code that associates a given Modbus User Register with the related variable. The AKD BASIC Programming Manual section 3.11.3 defines the address range for user variables (5000 to 5999).

```
7  '----- Define (dim) Global Variables -----
8  Dim indexdistance as integer
9  Dim indexspeed as integer
10 Dim jogspeed as integer
11 Dim starthome as integer
12 Dim startmove as integer
13 Dim moveselection as integer
14 Dim startmoveselected as integer
15 Dim homefound as integer
16 Dim startjogpos as integer
17 Dim startjogneg as integer
18
19 MBInfo
20 $MBMap32(5000, indexdistance) 'Modbus registers 5000 and 5001 for indexdistance from HMI
21 $MBMap32(5002, indexspeed) 'Modbus registers 5002 and 5003 for indexspeed from HMI
22 $MBMap32(5004, jogspeed) 'Modbus registers 5004 and 5005 for jogspeed from HMI
23 $MBMap32(5006, starthome) 'Modbus register 5006 for starthome button from HMI
24 $MBMap32(5008, startmove) 'Modbus register 5007 for startmove button from HMI
25 $MBMap32(5010, moveselection) 'Modbus register 5008 for startmove button from HMI
26 $MBMap32(5012, startmoveselected) 'Modbus register 5009 for startmove button from HMI
27 $MBMap32(5014, homefound) 'Modbus register 5010 for homestatus indicator in HMI
28 $MBMap32(5016, startjogpos) 'Modbus register 5011 for startjogpos button from HMI
29 $MBMap32(5018, startjogneg) 'Modbus register 5012 for startjogneg button from HMI
30 End
```

3.11.3 User Created Variables with Assigned Modbus Address Numbers

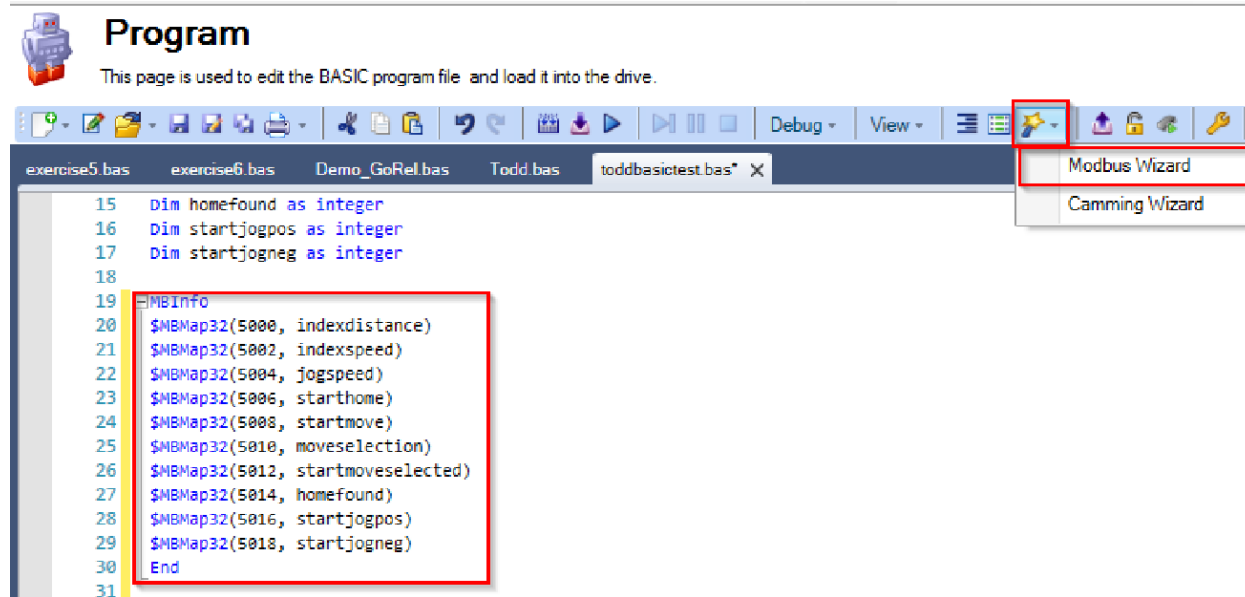
User variables can be assigned an Modbus address number. The range of available numbers is from 5000 to 5999. An example program:

```
Dim int2 as integer
Dim flt1 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 variables 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.

The AKD BASIC Programming Editor has a Modbus Mapping Wizard which inserts/configures the list in the example program.




From the sample project, the current values of the speed and distance variables are printed to the Console and used for the current move.

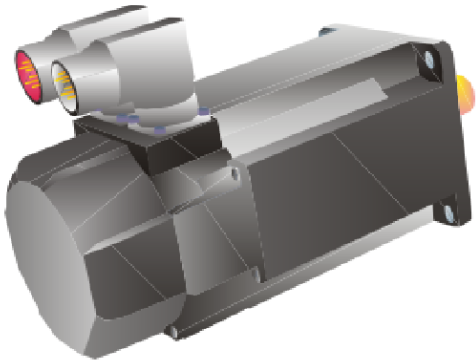
```
31 '----- Main Program -----
32 Main
33 DRV.SMENABLE = 1
34
35 Move.Acc = 10000000 ' Acceleration (drive units)
36 Move.Dec = 10000000 ' Deceleration (drive units)
37
38
39 MOVE.POSCOMMAND=0 'reset position to zero
40
41 While 1 = 1 'start continuous loop
42
43
44 'Indexing Move
45 While DIN1.STATE = 0: Wend 'Wait for Din1 to turn on
46 Print "speed= ",indexspeed
47 Print "distance= ",indexdistance
48 Move.RunSpeed = indexspeed 'index speed in integer position units per sec (from HMI)
49 MOVE.RELATIVEDIST = indexdistance 'index distance in integer position units (from HMI)
50 Move.GoRel
51 While DIN1.STATE=1: Wend 'Wait for Din1 to turn off
52
53 Wend 'return to "While 1 = 1" loop
54 End Main
```

In order to use the HMI's set data in the scaled format (i.e. 1.000 on the HMI display will equal 1000 in the AKD BASIC variable/program), the AKD's Units were set to custom (mechanics dependent) and the UNIT.PIN was set to 1000, the label set to the meaningful application units, and UNIT.POUT was set to 1 rev in this example. These settings are application dependent so it is up to the user to insure the setup is correct for their case. Keep in mind that this rescales the Limits screen items as well as the displayed position and velocity values within Workbench.

Units

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

Select Type of Mechanics:



Position Unit:

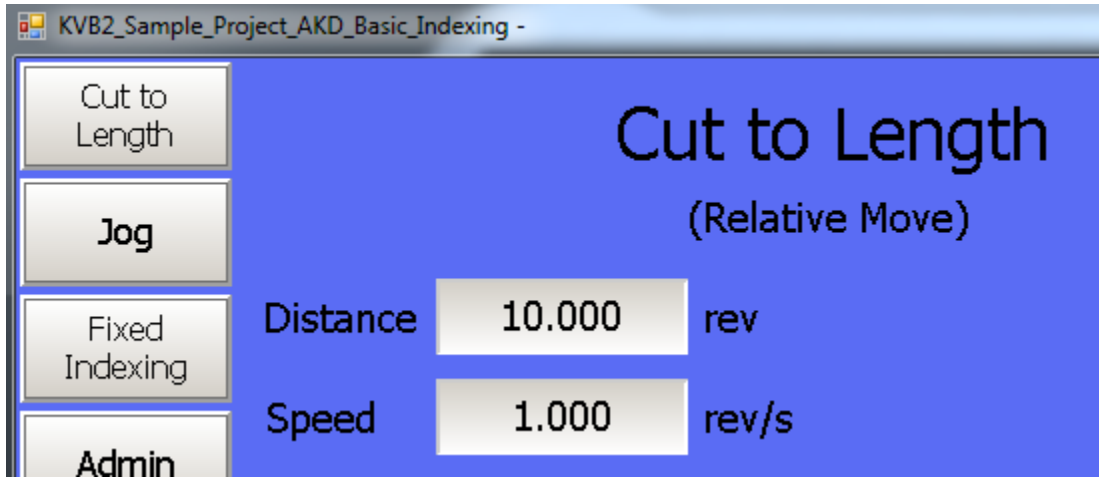
Velocity Unit:

Acceleration Unit:

Modbus Unit: [Goto Modbus](#)

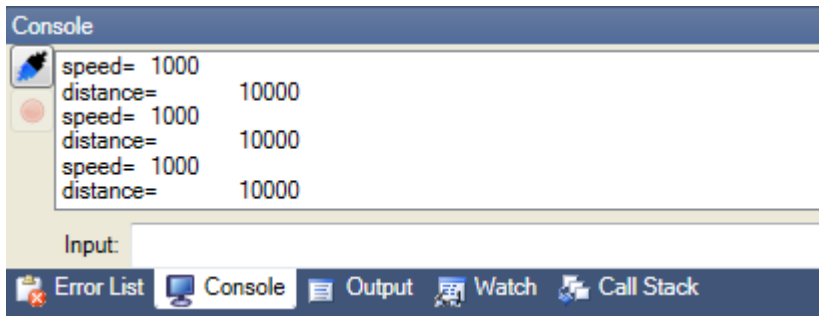
Custom
Label:
 in rev.

During runtime, the HMI was set for 10.000 (revs) and 1 (rev/sec).



The printout in the AKD BASIC Console indicated the value of 10000 for the distance and 1000 for the speed.

Based on the UNIT scaling in the drive, this implies 10.000 rev and 1.000 rev/s.

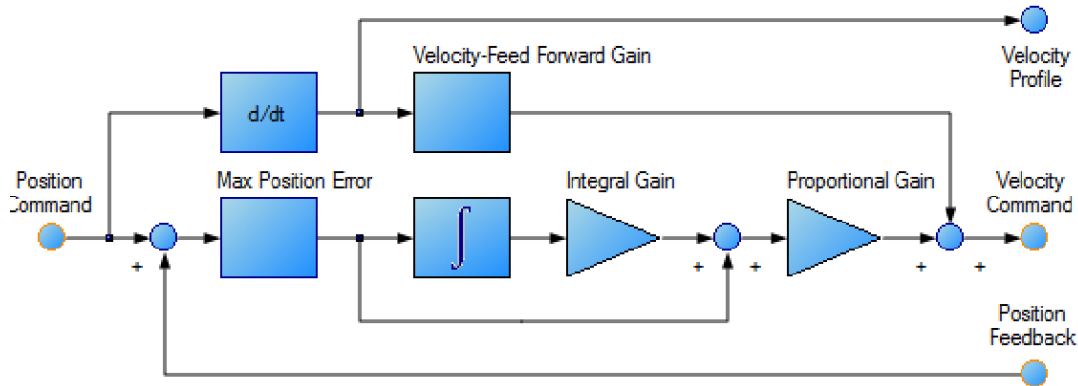


After execute of the 10 revolution move, the Position Command and Position feedback display “10,000” inches which again implies 10.000 revs based on the scaling.



Position Loop

The parameters for controlling the position of the motor.



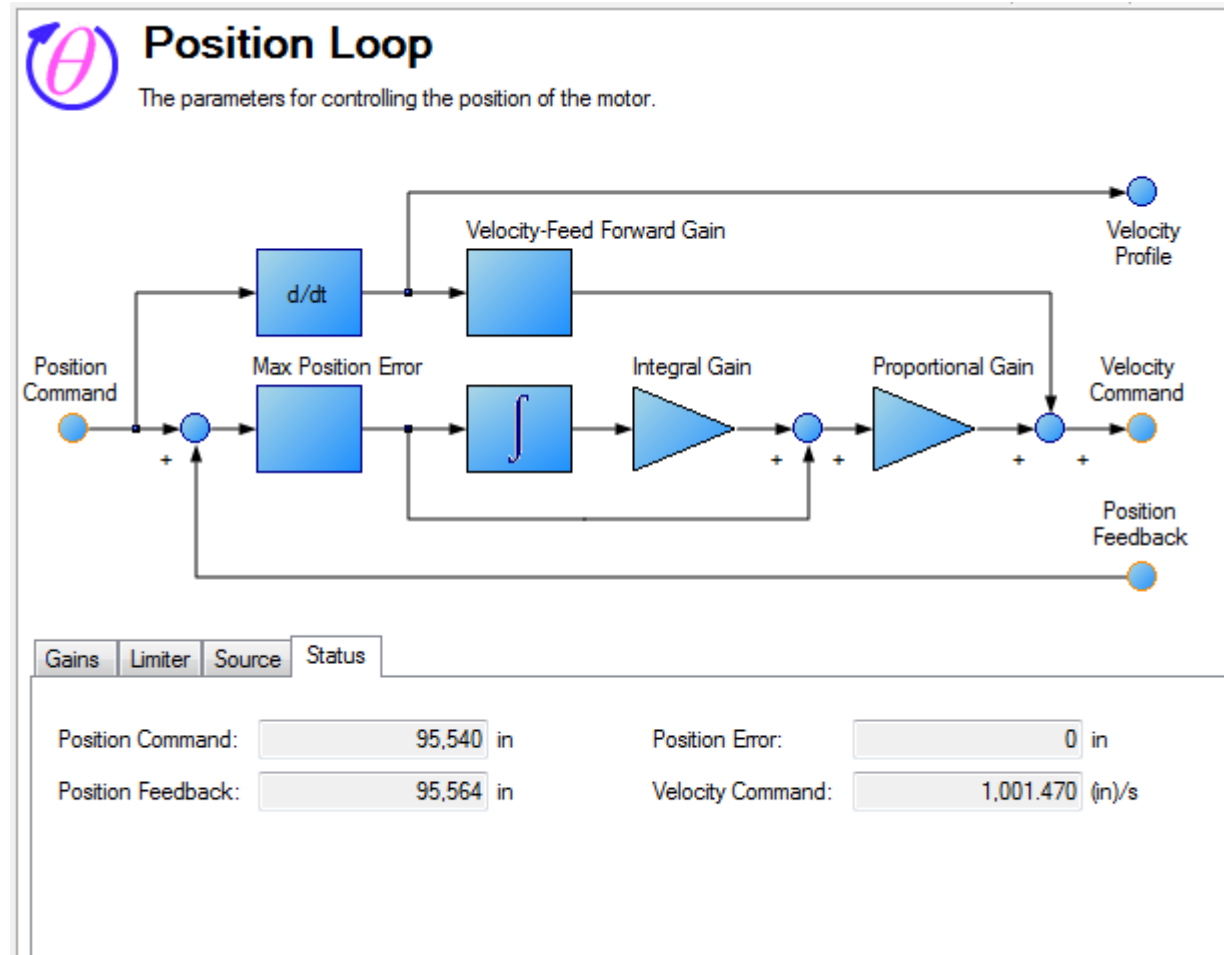
Gains Limiter Source Status

Position Command: in
Position Feedback: in

Position Error: in

Velocity Command: (in)/s

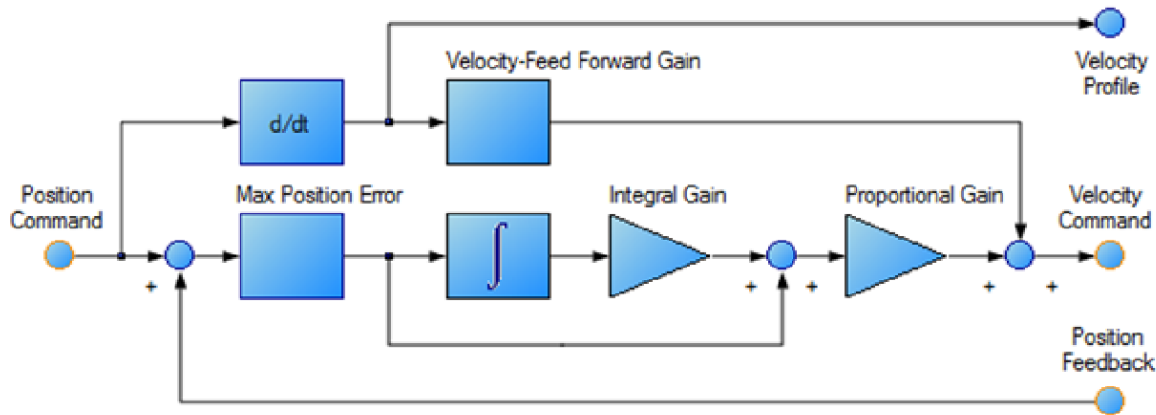
Likewise, during the move, the Velocity Command displays the 1,000 in/s which again implies 1.000 in/s based on the scaling.





Position Loop

The parameters for controlling the position of the motor.



Gains Limiter Source Status

Position Command: in

Position Error: in

Position Feedback: in

Velocity Command: (in)/s