KOLLMORGEN

Because Motion Matters™

# Using EtherNet/IP with RSLogix for Kollmorgen Automation Suite (KAS)

This application note has two parts. Chapter 1 is an in-depth, step-by-step "how to" of adding Ethernet/IP communications to a KAS project as well as adding KAS support to a new or existing RSLogix5000 project. Chapter 2 is a hands-on quick start guide that explains how to upload the KAS EIP sample projects (.kas & .ACD files) in order to demonstrate correct setup / working Ethernet/IP communications.



#### Chapter 1: Adding KAS Support to a New or Existing RSLogix5000 Project

PDMM 2-axis demo (p/n KAS-DEMO-100) with CompactLogix L32E controller

KOLLMORGEN

Because Motion Matters<sup>™</sup>

#### Part 1 – KAS Fieldbus Configuration

- 1. Start KAS IDE and open the project with which you want to use EtherNet/IP. Note this example uses the PDMM Controller.
- 2. Right click on the Controller in the Project Explorer and select Add Fieldbus.



3. Fieldbus will appear in the Project Explorer tree and the Fieldbus Editor screen will display.



KOLLMORGEN

Because Motion Matters™

4. Click on the Insert Network button I. Select the Ethernet/IP Adapter (server) configuration and click OK.



5. Click on the Insert Slave button to add an I/O Object. Select Type: I/O Outputs (Originator to Target). Set Identification Instance to 1 and Size to 64 (bytes). Suggested Description: "written by PLC / read by KAS." Click OK.



KOLLMORGEN

Because Motion Matters™

6. Click on the Insert Slave button to add an I/O Object. Select Type: I/O Inputs (Target to Originator). Set Identification Instance to 2 and Size to 64 (bytes). Suggested Description: "read by PLC / written by KAS." Click OK.

Image: Second system     Name       Image: Second system     Name       Image: Second system     Image: Second system		Fieldbus Editor*	
		Ethernet/IP Adapter (server)  Ether	Name
Image: Construction of the second	Mode I/Os (I	Type       OK         C I/O: Outputs (Originator to Target)       Cancel         I/O: Inputs (Target to Originator)       Cancel         Vendor specific object (Explicit messaging)       Cancel         Identification       Instance:         Class:       Instance:         Read Only       Size:         Description       read by PLC / written by KAS	Description written by PL

7. The I/O Assemblies for Implicit Messaging (Assembly Messaging) should now be configured and display in the Fieldbus Editor.

Fieldbus Editor*
Ethernet/IP Adapter (server)     Served I/Os and objects     If/o] 1 [64] - written by PLC / read by KAS     Ifin [i/o] 2 [64] - read by PLC / written by KAS

**Note:** In the KAS I/O Object definition, the Assembly size was set to 64 bytes or 64 x 8-bit. In the Rockwell I/O Object definition, notice the Assembly size is set to 16 x 32-bit. The Assemblies are the same size: 512 bits total. As is typically the case when integrating two manufacturer's products, each manufacturer has their own way of defining things. It is important to be mindful of this difference in definition.

KOLLMORGEN

Because Motion Matters™

The next step is to add user variables to be passed back and forth between the PLC and KAS. In this example, Boolean (BOOL), double integer (DINT) and long real (LREAL) type variables are shown. The user is responsible for dimensioning these.

8. Highlight "[i/o] 1 [64] – written by PLC / read by KAS" in the main window. Drag variables from the Dictionary into the lower window with the columns labeled "Symbol, Offset, Bit, Format." Notice when the variables are initially dropped into the I/O table, they are all assigned address (Offset) 0 and are Bit Format. The user must dimension the variables to the correct memory location.

Dictionary	₽×				Fieldbus Editor*	
Dictionary Controller Track Selection  Controller Type  Controller	<b>₽</b> ×	Symbol	Ethernet/IP	Adapter (server) /Os and objects 1 [64] - written by PL 0.0: JogSpeed 0.0: bEStop 0.0: MachineState 2 [64] - read by PLC	Fieldbus Editor*	Format
JogAxis2Minus BOOL Axis1Status DINT		Symbo JogSpe	l ed	Offset O	Bit O	Format Bit
Axis2Status DINT		bEStop		0	0	Bit
PipeNetwork PNCode		Machin	eState	0	0	Bit
Retain variables						
PNCode     ProfilesCode						
EtherCATCode	>					
Project Explorer Dictionary Libraries						

KOLLMORGEN

Because Motion Matters™

In KAS, each variable will have an offset of 4 bytes from the previous variable because in RSLogix all memory locations are 32-bit.

Dictionary 🗗 🗙							Fieldbus Editor*	
Controller: 🗸 📃 Track Se	election			(B)	Ethernet/IP	Adapter (server)		
1A Name	Туре	Dim.	7	Ē	Euroneun in La Served	I/Os and objects		
🗆 📷 Global variables				퓲	 	1 [64] - written by PL	.C / read by KAS	
JogSpeed MasterAbsPos MasterDeltaPos MachineSpeed	LREAL LREAL LREAL LREAL					0: JogSpeed 4.0: bEStop 8: MachineState  2 [64] - read by PLC	/ written by KAS	
MachineState bMasterAbs bMasterRel bEStop bLedStatus Profiles EtherCAT JogAxis1Plus JogAxis1Minus JogAxis2Plus	DINT BOOL BOOL BOOL ProfilesCo EtherCAT BOOL BOOL BOOL	[03]	F					
JogAxis2Minus	BOOL				Symbol	Offset	Bit	Format
Axis1Status	DINT				JogSpeed	0	0	32 bit - signed
Axis2Status	DINT				bEStop	4	0	Bit
PipeNetwork	PNCode				MachineState	8	0	32 bit - signed
Retain variables								
🕀 💼 Main								
PNCode			-					
ProfilesCode     Market Strength			-					
EthercAlCode		>						
Project Explorer Dictionary	Libraries							

KOLLMORGEN

Because Motion Matters™

9. Highlight "[i/o] 2 [64] – read by PLC / written by KAS" in the main window. Drag variables from the Dictionary into the lower window with the columns labeled "Symbol, Offset, Bit, Format." The user must dimension these variables as well. Since a Boolean variable is only 1 bit, 32 Boolean variables are dimensioned into one Offset (memory location). See the example of bLedStatus in the screen shot below. For this Boolean variable, the Offset is always 4, but the Bit value changes for each bit of the variable.

Dictionary		₽×			Fieldbus Editor*		
Controller  Track Se Controller  Global variables JogSpeed MasterAbsPos MasterDeltaPos MachineSpeed MachineState bMasterAbs bMasterAbs bMasterRel bEStop bLedStatus Profiles	LREAL LREAL LREAL LREAL LREAL DINT BOOL BOOL BOOL BOOL ProfilesCo	[03]		PAdapter (server) I/Os and objects )1 [64] - written by PI 0: JogSpeed 4.0: bEStop 8: MachineState )2 [64] - read by PLC 0: MachineSpeed 4.0: bLedStatus[0] 4.1: bLedStatus[1] 4.2: bLedStatus[2] 8: Axis1Status	Fieldbus Editor*		
EtherCAT JogAxis1Plus JogAxis1Minus JogAxis2Plus JogAxis2Minus Axis1Status Axis2Status PipeNetwork	EtherCAT BOOL BOOL BOOL DINT DINT PNCode	F	Symbol MachineSpeed bLedStatus[0] bLedStatus[1]	12: Axis2Status Offset 0 4 4	Bit 0 0 1	Format 32 bit - signed Bit Bit	
Retain variables			bLedStatus[2] bLedStatus[3]	4 4	2 3	Bit Bit	
PANCode			Axis1Status Axis2Status	8 12	0	32 bit - signed 32 bit - signed	
Project Explorer Dictionary	Libraries	/					

10. Now the KAS Ethernet/IP configuration is complete.

KOLLMORGEN

Because Motion Matters<sup>™</sup>

#### Part 2 – RSLogix5000 Configuration

- 11. Start RSLogix5000 and open the project with which you want to use KAS.
- 12. Right click on the Ethernet port in the I/O Configuration and select "New Module..."



KOLLMORGEN

Because Motion Matters™

Module		Description	Vendor
🗐 · Co	mmunications		
	- 1734-AENT	1734 Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
	- 1734-AENTR	1734 Ethernet Adapter, 2-Port, Twisted Pair Media	Allen-Bradley
	- 1738-AENT	1738 Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
	- 1738-AENTR	1738 Ethernet Adapter, 2-Port, Twisted Pair Media	Allen-Bradley
	- 1756-EN2F	1756 10/100 Mbps Ethernet Bridge, Fiber Media	Allen-Bradley
	- 1756-EN2T	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
	1756-EN2TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair	Allen-Bradley
	1756-EN3TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair	Allen-Bradley
	- 1756-ENBT	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
	- 1756-ENET/A	1756 Ethernet Communication Interface	Allen-Bradley
	1756-ENET/B	1756 Ethernet Communication Interface	Allen-Bradley
	1756-EWEB/A	1756 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv	Allen-Bradle
	1757-FFLD/A	1757 Foundation Fieldbus Linking Device	Allen-Bradle
	1768-ENBT/A	1768 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradle
	1768-EWEB/A	1768 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv	Allen-Bradle
	1769-L23E-QB1 E	10/100 Mbps Ethernet Port on CompactLogix5323E-QB1	Allen-Bradley
	- 1769-L23E-QBFC1	10/100 Mbps Ethernet Port on CompactLogix5323E-QBF	Allen-Bradle
-	1769-L32E Ethern	10/100 Mbps Ethernet Port on CompactLogix5332E	Allen-Bradle
	1769-L35E Ethern	10/100 Mbps Ethernet Port on CompactLogix5335E	Allen-Bradle
	- 1783-EMS04T	1783-EMS04T Ethernet Managed Switch	Allen-Bradle
	- 1783-EMS08T	1783-EMS08T Ethernet Managed Switch	Allen-Bradle
	- 1783-ETAP	3 Port Ethernet Tap, Twisted-Pair Media	Allen-Bradle
	- 1783-ETAP 1F	3 Port Ethernet Tap, 1 Fiber/2 Twisted-Pair Media	Allen-Bradle
	- 1783-ETAP2F	3 Port Ethernet Tap, 2 Fiber/1 Twisted-Pair Media	Allen-Bradle
	- 1788-EN2DN/A	1788 Ethernet to DeviceNet Linking Device	Allen-Bradle
	- 1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradle
	1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv	Allen-Bradle
	1794-AENT	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradle
	Drivelogix5730 Et	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradle
	ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradle
	ETHERNET-MODU	Generic Ethernet Module	Allen-Bradle
	EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradle
< 1	DECERTIN	makenesis a densities. This is a distribution of the standard	0l l l N
		Find	Add Favori
Ву С	ategory By Ve	ndor Favorites	

13. Select "ETHERNET-MODULE" under Communications and click OK.

KOLLMORGEN

Because Motion Matters<sup>™</sup>

14. Enter the settings for the new module as described below, verify the "Open Module Properties" checkbox is checked and click OK.

New Module						×
Type: Vendor: Parent:	ETHERNET-MODULE Generic Etheme Allen-Bradley LocalENB	t Module				
Name: Description:	KAS_EIPslave	Connection Para	Assembly Instance:	Size:		
b coonpain.		Input:	2	16	🛟 (32-Біt)	
		Output:	1	16	🛟 (32-Біt)	
Comm Format: Address / H	Data - DINT 💌	Configuration:	3	0	🛟 (8-bit)	
<ol> <li>IP Addre</li> </ol>	ss: 192 . 168 . 0 . 105	Status Input:				
🔿 Host Na	me:	Status Output:				
🗹 Open Modu	le Properties	ОК	Canc	el	Help	]

15. The "New Module" window displays as a "Module Properties: Local ENB" window with the Connection tab selected. Set the Requested Packet Interval (RPI) value to 10.0ms.

**Note:** Depending on the amount of data being transferred between the two controllers, the RPI may need to be reduced to 20.0ms. If option "Use Unicast Connection over EtherNet/IP" is visible, make sure it is checked. Click OK.

General Connection Module Info Requested Packet Interval (RPI): 10.0 🗢 ms (1.0 - 3200.0 ms)
Requested Packet Interval (RPI): 10.0 🗢 ms (1.0 - 3200.0 ms)
☐ Major Fault On Controller If Connection Fails While in Run Mode ✓ Use Unicast Connection over EtherNet/IP
Module Fault
Status: Office OK Cappel Apply Help

KOLLMORGEN

Because Motion Matters<sup>™</sup>

16. The KAS controller should now be configured and will show up under the Ethernet Port.



KOLLMORGEN

Because Motion Matters™

17. Make sure the Ethernet port for the Rockwell controller is setup with a compatible IP address on the same subnet as the KAS controller IP address. Right click on "Ethernet Port LocalENB" and select Properties.



Verify the Rockwell controller IP address is on the same subnet as the KAS controller. In this example, 192.168.0.xxx is correct.

The RSLogix5000 configuration is complete.

KOLLMORGEN

Because Motion Matters™

#### Part 3 – Utilizing Ethernet/IP Communications

 To monitor the Ethernet/IP communications, click on Controller Tags in the Controller Organizer tree in RSLogix5000. KAS\_EIPslave:I data is the Input Assembly "[i/o] 2 [64] – read by PLC / written by KAS." KAS\_EIPslave:O is the Output Assembly "[i/o] 1 [64] – written by PLC / read by KAS."

ntroller Organizer 🚽 🕂 🗙	Scope: 🛐 KAS_App_Note_ 🗸	Show: All Tags				🗸 🏹 Enter Na	vne Filter
Controller KAS_App_Note_Start	Name	_≘∆ Value	Force Mask 🗧 🗧	Style	Data Type	Description	Constant
Controller Fault Handler	+ KAS_EIPslave:C	{	{}		AB:ETHERNET		
Power-Up Handler	- KAS_EIPslave:I	{]	{}		AB:ETHERNET		
🔄 Tasks	KAS EIPslave:I.Data	{	{}	Decimal	DINT[16]		
😑 🤕 MainTask	+ KAS EIPslave:I.Data[0]			Decimal	DINT		
🖃 🥰 MainProgram	+ KAS_EIPslave:I.Data[1]	(	)	Decimal	DINT		
Program Tags	+ KAS_EIPslave1 Data[2]	(	1	Decimal	DINT		
	+ KAS_EIPslave1 Data[3]		1	Decimal	DINT		
Motion Groups	+ KAS_EIPdaverI Data[4]		2	Decimal	DINT		
Ungrouped Axes	KAS_EIRalayerI.Data[4]		2	Decimal	DINT		
- Add-On Instructions	KAS_ElPsiaveri Data[S]		2	Decimal	DINT		
- 🔄 Data Types	KAS_EIFsiave.I.Data[6]			Decimal	DINT		
User-Defined	+ KAS_EIPsiave:I.Data[7]		2	Decimal	DINT		
E Le Strings	+ KAS_EIPslave:I.Data[8]		)	Decimal	DINI		
	+ KAS_EIPslave:I.Data[9]		2	Decimal	DINT		
Module-Defined	+ KAS_EIPslave:I.Data[10	)] (	0	Decimal	DINT		
- Trends	+ KAS_EIPslave:I.Data[11	1] (	)	Decimal	DINT		
- 🦣 I/O Configuration	+ KAS_EIPslave:I.Data[12	2] (	)	Decimal	DINT		
🖮 🎹 Backplane, CompactLogix System	HKAS_EIPslave:I.Data[13]	3] (	0	Decimal	DINT		
1769-L32E KAS_App_Note_Start	+ KAS_EIPslave:I.Data[14	4] (	5	Decimal	DINT		
1769-L32E Ethernet Port LocalENB	+ KAS_EIPslave:I.Data[15	5] (	0	Decimal	DINT		
Ethernet	- KAS_EIPslave:0	{	{}		AB:ETHERNET		
1/69-L32E Ethernet Port LocalENB	KAS_EIPslave:0.Data	{]	{}	Decimal	DINT[16]		
CompactBus Local	+ KAS_EIPslave:0.Data[0	)] (	0	Decimal	DINT		
	+ KAS EIPslave:0.Data[1	1 (	0	Decimal	DINT		
	+ KAS EIPslave:0.Data[2	1 (	0	Decimal	DINT		
	+ KAS_EIPslave:0.Data[3	3] (	)	Decimal	DINT		
	+ KAS_EIPslaver0_Data[4	1 (	)	Decimal	DINT		
	+-KAS_EIPdaver0_Data[4		2	Decimal	DINT		
	+-KAS_EIPslaver0_Data[6	3	2	Decimal	DINT		
	E KAS_ElPalaver0. Data[0	n (	2	Decimal	DINT		
	KAS_EIFsiave.0.Data[7	J		Decimal	DINT		_
		)j		Decimal	DINT		_
	+ KAS_EIPslave:U.Data[9	ı] (		Decimal	DINT		
	+ KAS_EIPslave:U.Data[1	U] (	)	Decimal	DINT		
	Monitor Tags Edit Tag	s/			<		

KOLLMORGEN

Because Motion Matters™

19. To edit the Output Assembly data, simply click in the Value field of the memory location to edit. For example, the variable MachineState is Memory Location [2] in the Output Assembly (KAS\_EIPslave:O.Data[2]). Keep in mind the note regarding how the two manufacturers define the Assemblies. In KAS IDE, MachineState is Offset 8 [bytes]. Divide 8 by 4 (since Rockwell defines the memory locations as 4 bytes or 32-bits each) to get memory location 2, KAS\_EIPslave:O.Data[2].

_						
	E-KAS_EIPslave:0	{}	{}		AB:ETHERNET	
	- KAS_EIPslave:0.Data	{}	{}	Decimal	DINT[16]	
		0		Decimal	DINT	
		0		Decimal	DINT	
		1		Decimal	DINT	
		0		Decimal	DINT	
		0		Decimal	DINT	
	ELKAS, EIRoloue:0 Disto[5]	n		Decimal	DINT	

Fieldbus Editor									
Ethernet/IP Adapter (server)									
⊟ – ∰a Served I	/Us and objects	C Loss d by KAC							
$\square \square $									
□ III (i/o) 2 [64] - read by PLC / written by KAS									
C: MachineSpeed = 0.0									
🔜 4.0: bLedStatus[0] = TRUE									
🛄 4.1: bLedStatus[1] = TRUE									
🛄 4.2: bLedStatus[2] = TRUE									
🖬 4.3: bLedStatus[3] = FALSE									
📖 🛄 12: Axis2Status = 4415									
Symbol	Offset	Bit	Format						
JogSpeed=0.0	0	0	32 bit - signed						
bEStop=FALSE	4	0	Bit						
MachineState=1	8	0	32 bit - signed						
	Symbol Symbol Bestop=FALSE MachineState=1	Symbol       Offset         Symbol       Offset         Symbol       Offset         Symbol       Offset         Symbol       Served	Symbol       Offset       Bit         Symbol       Offset       Bit         Symbol       Offset       0         Symbol       0       0         Bit       0       0         Symbol       0       0         Symbol       0       0         Bit       0 </th <th>Served I/Os and objects         Image: Served I/Os and objects         Image:</th>	Served I/Os and objects         Image:					

As shown – when the value is changed to a 1 in RSLogix5000, it automatically updates the Fieldbus Editor in KAS IDE.

KOLLMORGEN

Because Motion Matters<sup>™</sup>

Alternatively, add code to the RSLogix5000 ladder to change the value of the KAS\_EIPslave:O.Data[2] from the ladder.

	SetState	MOV
0		Move
	50	Source MachineState
		1€
		Dest KAS_ElPslave:O.Data[2]
		1€

KOLLMORGEN

Because Motion Matters™

#### Chapter 2: Quick Start with the KAS EIP Sample Projects



The primary purpose of the sample projects is to demonstrate the correct setup of the EtherNet/IP communication channel between a PDMM and a CompactLogix L32E controller. RSLogix\_KAS\_SampleProject.ACD for RSLogix5000 paired with KAS\_EIP\_SampleProject.kas for KAS IDE demonstrates the correct setup of each controller.

The sample projects can help you learn:

- How to setup cyclic Ethernet/IP communication in KAS IDE and RSLogix5000
- How to start motion from the CompactLogix controller
- How to change motion variables (such as machine acceleration and deceleration)
- How to read/write individual AKD axis drive parameters (such as current limits)
- How to read drive status (Faulted, Enabled, Disabled, etc.)

KOLLMORGEN

Because Motion Matters™

#### Part 1 — Setup Controllers for Cyclic EtherNet/IP Communications

**Note:** These sample projects were designed for the KAS PDMM 2-axis demo kit, p/n KAS-DEMO-100.

- 1. Start KAS IDE and open the file KAS\_EIP\_SampleProject.kas.
- 2. You will most likely need to update the Controller Properties to match your specific installation. Right click on Controller at the top of the tree and select "Properties."



3. Update the IP Address in order to match your specific hardware setup and then click OK to close the controller properties window.

Controller Properties					
IP Address	192.168.0.105				
Version					
Controller type					
O PAC					
Download Project Source to the controller					
OK Cancel					

#### KOLLMORGEN

Because Motion Matters™

- 4. Next, open the EtherCAT Devices editor by double-clicking on EtherCAT in the tree.
- 5. Click the Scan Devices button on the far right. Verify that AKD\_1 is mapped to AXIS1 and AKD\_2 is mapped to AXIS2.
- Once you have updated all of the configuration settings to match your specific hardware setup, you can compile and download the program to the PDMM controller and then Run the project.
- 7. Start RSLogix5000 and open the file RSLogix\_KAS\_SampleProject.ACD.
- 8. You will most likely need to update the controller properties to match your specific installation. Right click on the controller ("KAS\_EIP\_Project") at the top of the tree and select "Properties."



KOLLMORGEN

Because Motion Matters™

9. Update any controller properties in order for the controller to match your specific hardware setup, most notably any communication settings and/or the controller type, and then click Apply and close the controller properties window.

🛿 Controller Properties - KAS_EIP_Project				
Date/Time General g	Advanced SFC Execution File Nonvolatile Memory Memory Serial Port System Protocol User Protocol Major Faults Minor Faults			
Vendor:	Allen-Bradley			
Туре:	1769-L32E CompactLogix5332E Controller Change Controller			
Revision:	19.11			
Name:	KAS_EIP_Project			
Description:				
Chassis Tune:				
Class				
5100				
	OK Cancel Annly Help			

KOLLMORGEN

Because Motion Matters™

10. Next, open the Ethernet-Module setup for the axis communications by right clicking on "ETHERNET-MODULE KAS\_eipSlave" under the Ethernet port in the I/O Configuration tree.



KOLLMORGEN

Because Motion Matters™

11. Update the IP Address in order to match your specific hardware setup and then click OK to close Module Properties.

🗖 Module Pro	operties: LocalENB (ETHERNET	-MODULE 1.1)			×	
General Connection Module Info						
Type: Vendor: Parent:	ETHERNET-MODULE Generic Ethern Allen-Bradley LocalENB	net Module				
Name: Description:	KAS_eipSlave	Connection Para	Assembly Instance:	Size:		
	V	Input: Output:	2	16 🛟 (32-bit) 16 🛟 (32-bit)		
Comm Format: Address / H	Data - DINT  ost Name	Configuration:	3	0 🛟 (8-bit)		
IP Addre     Host Nar	ss: 192 . 168 . 0 . 105	Status Input: Status Output:				
Status: Offline OK Cancel Apply Help					)	

12. Once you had updated all of the configuration settings to match your specific hardware setup, you can download the program to the L32E controller and put the project in Run Mode. You should see the "I/O OK" LED show solid Green as pictured below.

🔀 RSLogix 5000 - KAS_EIP_Project in KAS_EIP_Sampl	e_Pr			
🛱 File Edit View Search Logic Communications Tools Wi	ndow			
Rem Run     Run Mode       No Forces     Controller OK       Battery OK       I/O OK	Path H			
Controller Organizer				

KOLLMORGEN

Because Motion Matters™

**Note:** If a blinking green LED with the message "I/O Not Responding" displays, it is likely the KAS project is not running. Go to KAS IDE and click Run button

13. After both projects are running, you should see the Control Panel in the KAS IDE updating.



**WARNING!** If you try to run the KAS project while the RSLogix5000 project is not running, you will encounter issues. All the variables that are mapped to be written by the PLC (listed under " $O \rightarrow T$  [i/o] 1 [64] – written by PLC"), cannot be changed / forced to a new value within the KAS IDE. They must be changed by PLC. If PLC is not running, you will not be able to change these variables.

KOLLMORGEN

Because Motion Matters™

#### **Explanation of the Sample Projects**

The KAS sample project is a 2-axis structured text, Pipe Network program. It is based on the standard application template available in the KAS IDE. The program execution is dependent on the machine state [program variable MachineState – written by AB controller]. There are three possible machine states; OFF is 0 (default). Setting MachineState to 1 enables individual axis motion (jog plus and minus). Setting MachineState to 2 enables Pipe Network / Master axis motion (jogging or indexing moves).

For individual axis motion, the jog speed is a program variable [JogSpeed] written by the AB controller. To start clockwise (plus) or counter-clockwise (negative) motion for each axis a bit must be set from the AB controller [Boolean program variables JogAxis1Plus, JogAxis1Minus, JogAxis2Plus, JogAxis2Minus]. The Homing Slider bar is used to select the type of homing routine to execute. If the KAS Demo Unit hardware (p/n KAS-DEMO-100) is present, make this selection. When the Axis 1 button is clicked, the program physically homes the axis, moving the motor shaft to the home switch. If the KAS Demo Unit hardware is not present / the KAS Simulator is being used, select Other. If the Simulator is being used, when the Axis 1 button is clicked, the virtual motor is homed to "top-dead-center" position. If a real motor is being used, when the Axis 1 button is clicked the motor is homed to absolute zero.

For Pipe Network Motion, the machine speed, acceleration, and deceleration as well as relative move distance or absolute target position are written by the AB controller [program variables MachineSpeed, MachineAccel, MachineDecel, MasterAbsPos, MasterDeltaPos respectively].

For all modes of operation, velocity and position feedback data is read by the AB controller. Finally, the axis 1 current limits are an example of an AKD drive parameters (IL.LIMITP and IL.LIMITN) that are written by the AB controller.

The RSLogix\_KAS\_SampleProject.ACD is a very simplistic ladder that shows reading and writing values in the objects defined by the Input and Output Assemblies (Ethernet/IP cyclic communications channel). In this case, the program flow and control is defined by the structured text program in the PDMM controller. The RSLogix project is simply reading and setting variables in the PDMM program.

KOLLMORGEN

Because Motion Matters™

#### Appendix

Data Types KAS vs. Rockwell

BOOL	Boolean (Bit)	FALSE or TRUE - stored in 1 byte
SINT	Small signed integer in 8 bits	-128 to +127
INT	Signed integer in 16 bits	-32768 to +32767
DINT	Signed double precision integer in 32 bits	-2147483648 to +2147483647
REAL	Single precision floating point stored in 32 bits	-3.4E38 to 3.4E38 and -3.4E-38 to 3.4E-38 (6 to 7 significant digits of accuracy)
LREAL**	Double precision floating point stored in 32 bits	-1.7E308 to 1.7E308 and -1.7E-308 to 1.7E-308 (14 to 15 significant digits of accuracy)

KAS IDE supports all data types listed. RSLogix5000 does not have the LREAL data type.

**Note:** In the examples above, the Generic Ethernet Module (KAS I/O Module in RSLogix5000) Communications Format is defined as DINT (32 bit signed integers) so only integer values are passed.