Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Rockwell Logix5000 AOI for MAC and MIS –motors.

Description

The JVL AOI's will simplify the integration into the Rockwell Logix5000 environment.

The AOI's are designed to emulate the functionality as close to the existing motion commands in the Rockwell Kinetix world as possible.

For the Rockwell motion programmer the JVL AOI's will help the integration of the JVL motors into a Rockwell application.

This document along with the example project should give a good understanding on how to setup and control a JVL motor.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	. Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Downloading the packages

2 different packages are required to control the JVL motors, one MIS and one for the MAC – motors.

The packages can be downloaded following the link below:

https://www.jvl.dk/1189/addon-instructions



Please observe that an EDS –file is used for the MAC –Servo motors, however this is not the case with the MIS – motors.

The EDS file must be imported into the Studio5000 Logix Designer environment.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JV	L Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

# Content

Downloading the packages	2
AOI for the MAC servo motors	5
Importing the EDS file into the Logix Designer (OBS! Only for MAC –motors)	5
Adding the MAC-motor as a new module to the project	6
Importing the JVL AOI's into the project.	8
AOI Description	10
JVL_COM_Servo	10
JVL_Config_Servo	12
JVL_SetParameter	13
JVL_MSO ( <b>M</b> otion <b>S</b> ervo <b>O</b> n)	13
JVL_MSF ( <b>M</b> otion <b>S</b> ervo <b>O</b> ff)	14
JVL_MAJ ( <b>M</b> otion <b>A</b> xis <b>J</b> og)	14
JVL_MAS ( <b>M</b> otion <b>A</b> xis <b>S</b> top)	15
JVL_MAM ( <b>M</b> otion <b>A</b> xis <b>M</b> ove)	16
JVL_MAH ( <b>M</b> otion <b>A</b> xis <b>H</b> ome)	17
Table of homing possibilities	
Supplied MAC AOI -Example	19
AOI for the MIS stepper motors	24
Setting up the project, the first steps	24
Message paths	24
AOI Description	25
JVL_MSO ( <b>M</b> otion <b>S</b> ervo <b>O</b> n)	25
JVL_MSF ( <b>M</b> otion <b>S</b> ervo <b>O</b> ff)	26
JVL_MAJ ( <b>M</b> otion <b>A</b> xis <b>J</b> og)	26
JVL_MAS ( <b>M</b> otion <b>A</b> xis <b>S</b> top)	27
JVL_MAM ( <b>M</b> otion <b>A</b> xis <b>M</b> ove)	28
JVL_MAH ( <b>M</b> otion <b>A</b> xis <b>H</b> ome)	29
Table of homing possibilities, Active = 1	31
Torque homing	32
Sensor homing	32

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
*			Revision:	Approved:
intelligent motors	JVI	. Rockwell AOI (Add On Instructions)	1.0	KRP
Supplied MIS AOI – Exa	mple			

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
			Revision:	Approved:
intelligent motors	JAI	. KOCKWEII AOI (Add On Instructions)	1.0	ккр

## AOI for the MAC servo motors

Importing the EDS file into the Logix Designer (OBS! Only for MAC -motors)

When AOI's for the MAC-motors are used, the EDS file is required.

The .eds file is included in the package with the AOI –files and the example project.

The EDS files holds a description of the different registers in the motor allowing a far more easy integration and predefined connection properties, the only parameter required for the connection is the IP address information.

Follow these steps to import the EDS -file into the Logix Designer.

1. Locate the "EDS Hardware Installation Tool"



- 2. Follow the guidelines and find the JVL\_MAC\_ServoMotors.eds –file.
- 3. Finally the EDS file should be successfully installed.



	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Adding the MAC-motor as a new module to the project

Now that the EDS file has been installed, the MAC motor will be recognized by the Logix Designer as a "JVL MAC ServoMotor".

To add a motor to the project follow these steps.

1. Right click on the Ethernet Controller and select "New module"



2. Type in "JVL" in the search field and the "JVL MAC ServoMotor" appears.

Select the motor type and press "Create"

0	CI	ear Filters		Show Filters 🍣
Catalog Number	Description	Vendor	Category	
JVL MAC ServoMotor	JVL MAC Servomotors	JVL Industrie Elektronik A/S	Communications Adapter	

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

3. Name the motor for the project and enter the IP address.

New Module	IP Address:     192 . 168 . 0 . 110
General"       Connection       Module Info       Parametern       Internet Protocol       Post Configuration       Network         Type:       JVL MAC Servolvtor JVL MAC Servolvtors       Ethernet Address       Parametern       Ethernet Address         Description:       Image:       Image:       Image:       Image:       Image:         Module Definition       2034       Ethernet Address       Image:       Image:         Image:       Image:       Image:       Image:       Image:	The IP address will follow the motor settings. The default IP address is always "192.168.0.XXX" where XX is the decimal number of the last byte from the MAC-address. The Mac address is visible from a label placed on the motor. Exc. If an MAC address has the number:
Status: Creating OK Cancel Help	Then the last byte in the MAC address is HEX B5 (0xB5) which is equivalent to 181 in decimal. So the stock default IP address of the motor will be <b>192.168.0.181</b>

Now the motor has been added to the project.

*Please note that the motor needs configuration before the I/O connection will be ok. This configuration of the motor is done from the JVL\_Comms –AOI described later.* 

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### Importing the JVL AOI's into the project.

1. Import the JVL AOI -set into the project by right click on the "Add on instructions"



4. Select the JVL AOI from the list and press "Open".

	dvikling\Dri	opbox\Software\EtherNetIP\AOI_Servo\A0	JI_MAC	V 0	Search AUI_MAC	
Organize 👻 New folder					311	• 🗆
Favorites	^	Name	Date modified	Туре	Size	
E Desktop		JVL_COM_Servo.L5X	11/7/2018 2:03 PM	Logix Designer X	76 KB	
Downloads		IVL_Config_Servo.L5X	11/7/2018 2:03 PM	Logix Designer X	33 KB	
😹 Recent places		IVL_MAH_Servo.L5X	11/7/2018 2:03 PM	Logix Designer X	102 KB	
😻 Dropbox		JVL_MAJ_Servo.L5X	11/7/2018 2:03 PM	Logix Designer X	29 KB	
		JVL_MAM_Servo.L5X	11/7/2018 2:04 PM	Logix Designer X	33 KB	No previ
📰 Desktop		IVL_MAS_Servo.L5X	11/7/2018 2:04 PM	Logix Designer X	31 KB	availabl
OneDrive		JVL_MessageHandler.L5X	11/7/2018 2:04 PM	Logix Designer X	3 KB	
Homegroup		JVL_MSF_Servo.L5X	11/7/2018 2:04 PM	Logix Designer X	25 KB	
JVL Industri Elektronik		図 JVL_MSO_Servo.L5X	11/7/2018 2:04 PM	Logix Designer X	24 KB	
Me This PC		IVL_SetParameter.L5X	11/7/2018 2:04 PM	Logix Designer X	24 KB	
Documents	~					
File name:	JVL_COM_	Servo.L5X		~	Logix Designer XM	L Files (*.L5X
					Onen	Cancel

Unfortunately it is not possible to select more than one AOI in the dialog, so each AOI needs to be loaded one at a time.

When all AOI's are loaded the tree structure of the "Add on instructions" should appear as follows:



Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

Finally the JVL\_MessageHandler routine needs to be added to the project.

This routine handles all explicit messages that are used for configuration and special functions.

The message handler routine is also included in the package and it is imported into the project following these guidelines:

1. Right click on the Task->MainTask->MainRoutine



Select Add->Import Routine

💛 🖌 🕇 📙 « So	ftware > MAC_AOI > jvl_mac_aoi > AOI_MAC	~	Ö	Search AOI_MAC	Q
Irganize 👻 New fold	er			Bii •	. 0
This PC	Name	Date modifie	rd	Туре	Size
Fictures	JVL_COM_Servo.L5X	29-07-20191	4:34	Logix Designer X	76 K
Documents	JVL_Config_Serve.L5X	29-07-2019 1	4:34	Logix Designer X	33 K
h Music	JVL_MAH_Servo.L5X	29-07-2019 1	4:34	Logix Designer X	102 K
JI MUDIC	JVL_MAJ_Servo.L5X	29-07-2019 1	4:34	Logix Designer X	29 K
Downloads	JVL_MAM_Servo.L5X	29-07-20191	4:34	Logix Designer X	33 k
Desktop	JVL_MAS_Servo.L5X	29-07-2019 1	4:34	Logix Designer X	31 K
Videos	JVL_MessageHandler.L5X	29-07-20191	4:34	Logix Designer X	3 K
💶 OS (C:)	JVL_MSF_Servo.L5X	29-07-2019 1	4:34	Logix Designer X	25 K
👷 SYS (\\JVL01) (I:)	JVL_MSO_Servo.L5X	29-07-20191	4:34	Logix Designer X	24 K
Development (	JVL_SetParameter.L5X	29-07-2019 1	4:34	Logix Designer X	24 k
Lager (\UVL01) (					
🗙 Edb (\\JVL01) (IV 🗸	٢				
File n	ame: IVI MessageHandler I 5X		~	Logix Designer XML Fi	les (*.1.5X ~

Select the "JVL\_MessageHandler.L5X" –file from the list and press open:

	Author: HC	File: LB0061-10GB	<sup>Date:</sup> 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## **AOI** Description

### JVL\_COM\_Servo

Since the AOI's works by using a combination of cyclic data exchange and explicit messages, the JVL\_COM\_Servo instruction must be called continuously. This AOI handles scaling and basic operation of the motor and ensures that the right data is setup and copied into the cyclic data exchange handled by the I/O.

The separate routine JVL\_MessageHandler previously included in the project, is called on request from the CALL\_MessageHandler -bit in the JVL\_COM\_Servo -instance.

The basic concept is displayed below in the 2 rungs taken from the JVL example.



Whenever the "Call\_MessageHandler" –bit controlled from the JVL\_Com\_Servo instance goes high the JVL\_MessageHandler is called.

JVL Main commu	nication control module
JVL_COM_Servo	LACT400 COMMS
CyclicRead	MAC400Actuator:I
CyclicWrite	MAC400Actuator:O
Module	MAC400Actuator
Axis	LACT400_AXIS
msgBufferOut	msgO_LACT400
msgBufferIn	msgl_LACT400
MSG_Out	JVL_OutMessage .
MSG_In	JVL_InMessage .
MSG value	JVL SetValue
Save Volatile	- 0 <del>+</del>

JVL_COM_Servo:	The instance name, exc. "LACT400_COMMS"
CyclicRead:	Pointer to the Cyclic read data of the MAC-motor.
	Exc. MAC400Actuator is the name of motor in the
	project, so we point on the MAC400Actuator:I
CyclicWrite:	Pointer to the Cyclic read data of the MAC-motor.
	Similar to CyclicRead, but points to the Output –
	data, exc. MAC400Actuator:O
Module:	Motor exc. MAC400Actuator.
Axis:	Axis –data object, Create an object with a name and
	finalize with CTRL + W. This object will be used in
	other AOI's ass well.
msgBufferOut:	Buffer that holds message data. Create an instance
	with CTRL+W.
msgBufferIn:	Buffer that holds message data. Create an instance
	with CTRL+W.
MSG_Out:	MSG instance create with CTRL+W.
MSG_In:	MSG instance create with CTRL+W.
MSG_Value:	MSG value -instance create with CTRL+W.
Save Volatile:	Input bit required to configure the motor and save
-	the settings in volatile memory in the motor. This is
	only required once with a factory configured motor.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	11/1			Approved:
	JVI	L ROCKWEIT AOT (Add On Instructions)	1.0	

Tip:

Since most of the input operands are structures, for convenience they can be created using

**CTRL + W** for automatically adding the tag to the controller tag list.

Invent logical names to the tags, some of them will be used elsewhere.

For the "Module" -operand select the motor added to the ethernet controller from the list.



In case the motor holds factory default settings the PLC will flag an I/O Connection error and I/O LED will blink. This is completely normal. Simply run the JVL\_CONFIG\_Servo AOI and set the JVL\_COM\_Servo.Initialize –bit = 1 as demonstrated below. Since the "Save\_Volatile" bit is set In the JVL\_COM\_SERVO instance, settings will be saved in flash memory. After the motor resets the PLC should automatically establish cyclic communication with the motor and the I/O status should recover from the fault condition.



	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_Config\_Servo

The JVL\_Config\_Servo is used to setup the application regarding scaling, servo filter settings and motor type used.

Using this function will configure the Axis -tag with the appropriate values that is transferred to the motor and used under special circumstances like during a homing sequence etc.

When the Initialize -bit is set in the JVL\_COM\_Servo instance, it will transfer the necessary data to the motor and setup the basic EthernetIP -settings.

This should only be necessary to do once, then the EthernetIP -settings and the settings in the motor as well as the Axis setting should be in place and after the next power cycle / reset all should be configured.

	Confi actual indicate ac	gures the servodrive ed by the IP idress.	
1	JVL_Con	fig_Servo-	
	Configures the actu	al servodrive indicate	
	JVL_Config_Servo	LACT400_CONF	(DN)-
	Axis	LACT400_AXIS	and the second second
	MotorType	2	
	MaxTorque	123 🗲	
	Units_per_Rev	60.0 +	
	GearRatio	1.0 +	
	UnitSelection	2 +	
	Load_Ratio	1.0 ←	
	Servo_Response	4 +	
	Load_Coupling	4 +	
_			

JVL_Config_Servo:	The instance name, exc. "LACT400_CONF"
Axis:	Axis –data object, Create an object with a name and finalize with CTRL + W. This object will be used in other AOI's ass well
MotorType:	Select which motortype used.
	1: MAC50-141
	2: MAC050G – MAC141G (New generation)
	MAC400 - MAC4500, MAC800 w. abs. encoder (8192/rev). 3: MAC800 w. incremental encoder (8000/rev)
MaxTorque:	Max. allowed torgue 0-300 [%]
Units_per_rev:	Units per. Revolution. This operand is used for scaling
	purposes. Exc.
	An actuator has a linear movement of 60mm / rev. then 60.0 is entered.
GearRatio:	If a gearbox is mounted the ratio is entered here n:1
UnitSelection:	With this selection all motion parameters will be scaled
	according to these settings.
	1: [um] and [um/s]
	2: [mm] and [mm/s]
	3: [m] and [m/s]
<b>.</b>	
Servo interna	li filter settings
Load_Ratio:	Inertia ratio between the motor and load. N:1
Servo_Response:	Filter response 0=slow, 4=fast.
Load_Coupling:	Rigidity of the servo filter, 0=Flexible, 4=Rigid

	Author: HC	File: LB0061-10GB	<sup>Date:</sup> 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_SetParameter

This AOI is mainly supporting the AOI's when a specific register need to be configured. It is not intended for custom use.

## JVL\_MSO (**M**otion**S**ervo**O**n)

This instruction is based on the Rockwell command MSO, which basically energizes the servo drive. In JVL terms it will put the motor into Position mode. This instruction is required to be called with success prior to other motion instructions to be issued.

Executes on rising edge.

Parameters:

JVL Motion Servo On



JVL_MSO_Servo: Axis:	The instance name, exc. "LACT400_MSO" The Axis object of the motor. The same as the Axis object used in the JVL_COM_Servo instance.
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 1 = Motor has a fault
Output DN:	4 = Motor is already enabled. Instuction has completed successfully

Author: HC	File: LB0061-10GB	Date: 13-08-	2019
JVI	Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_MSF (**M**otion**S**ervo**O**ff)

This instruction is based on the Rockwell command MSF, which disables the servo drive and makes it passive. Prior of calling the instruction the JVL\_MSO must successfully have been called otherwise an error will be flagged.

Executes on rising edge.

### Parameters:



JVL_MSF_Servo: Axis:	The instance name, exc. "LACT400_MSF" The Axis object of the motor. The same as the Axis object used in the JVL_COM_Servo instance.
Output EN:	Rung enable.
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled.
Output DN:	Instuction has completed successfully

## JVL\_MAJ (MotionAxisJog)

This instruction is based on the Rockwell command MAJ [MotionAxisJog], which starts jogging the servo drive in a certain direction with a certain speed. The jogging is stopped with the command JVL\_MAS. This instruction basically puts the motor in velocity mode using JVL terms.

Executes on rising edge.

Parameters:

Motion Axis J JVL_MAJ_Serve Motion Axis Jog		
Axis LACT40 Direction	0 AXIS CERCE	
Speed	5	
Accel_Decel_Rate	1	

The instance name, exc. "LACT400_MAJ" The Axis object of the motor. The same as the Axis object used in the JVL_COM_Servo instance.
Direction of motion: 0: CW
1: CCW
Velocity in the selected unit.
Acceleration and deceleration rate in the selected unit.
Dune enchie
Rung enable.
ERP for an errorcode
5 = Motor is not enabled
13 = an extended errorcode is issued
EXERR:
2= Direction is not 0 or 1
3= Speed < 0 (negative value entered)
4= Accel Decel Rate < 0 (negative value)
,,
Instuction has completed successfully

14/36

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
Intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### JVL\_MAS (MotionAxisStop)

This instruction is based on the Rockwell command MAS [MotionAxisStop], which stops any pending motion.

In case a JVL\_MAS instruction has been issued, the motors runs until the JVL\_MAS instruction is issued.

Executes on rising edge.



Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### JVL\_MAM (MotionAxisMove)

This instruction is based on the Rockwell command MAM [MotionAxisMove], which starts a move to a certain position either absolute or incremental with a

Executes on rising edge.

#### Parameters:



JVL_MAM_Servo:	The instance name, exc. "LAC1400_MAM"
Axis:	The Axis object of the motor. The same as the Axis
	object used in the JVL_COM_Servo instance.
MoveType:	Absolute (=0) or Relative (=1)
Position:	Position in the selected unit.
Speed:	Speed used in the selected unit
Accell_Decel_rate:	Acceleration and deceleration rate in the selected unit.
Profile:	0=Trapezoid, 1=S-Curve
Jerk:	S-Curve value in case Profile = 1.
	Values: [0-256]
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EVERP.
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveTyne out of range
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit)
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit) 4= Speed < 0 (negative value)
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit) 4= Speed < 0 (negative value) 5= Arcell Decel Rate < 0 (negative value)
Output EN: Output ER: Output DN:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit) 4= Speed < 0 (negative value) 5= Accell_Decel_Rate < 0 (negative value) Instruction has completed successfully
Output EN: Output ER: Output DN: Output IP:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit) 4= Speed < 0 (negative value) 5= Accell_Decel_Rate < 0 (negative value) Instruction has completed successfully In progress

In the example above the servo will move 100 units relative to the current position. It accelerate to speed of 5 units with acceleration setting 1000 unit/s. The setting has enabled S-Curve acceleration and the jerk is set to the max. value of 256, which is the max. S-Curve value for a JVL servo motor.

High settings of accelerations or decelerations can be hard on the mechanic construction and lead to premature failure and wearing.

The S-Curve acceleration and deceleration method is often used in applications where less wear and tear is desirable due to the limitation in the dV/dT factor of the acceleration.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	L Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### JVL\_MAH (MotionAxisHome)

This instruction is based on the Rockwell command MAH [MotionAxisHoming], which starts a homing sequence.

The basic ideer is that the motor either runs a sensor, Torque based or Encoder index -based homing sequence.

A sensor based homing triggers on the sensor input (AIN, see module specs for details) when it transitions either from high to low or low to high. The position is then preset to the value indicated in the "Position" –parameter.

The Torque homing method is less accurate due to tolerances in the mechanical construction and changes due to tear and wear. However when the motor detects an applied torque setting above the indicated in the Torque\_Max – parameter it will trigger and set the position to the indicated.

The last method triggers in the internal encoder index pulse and will thus be accurate within one revolution of the motor shaft. This method is very accurate however only usable within one shaft revolution.

Both the torque and the index – method eliminates the need for an external sensor.

Executes on rising edge.

Motion JVL_MA Motion Axis Home	n Axis Home \H_Servo	
JVL_MAH_Servo		-CEN)-
Position	LAC1400_AAIS 0	-CDN)
HomingMode	1	-(ER)
Sequence	4	(PC)
Direction	2	-(IP)
Speed	10	
Torque_Max	50	

JVL_MAH_Servo:	The instance name, exc. "LACT400_MAH"
Axis:	The Axis object of the motor. The same as the Axis
	object used in the JVL_COM_Servo instance.
Position:	The reference position to set in the homing position.
HomingMode:	0= Passive, 1= Active. Only Active (=1) supported.
Sequence:	Selects the method used for the homing process
	along with the Direction parameter. See the table
	for further details on which homing method to use
	and how to setup the Sequence and Direction
	parameters.
Direction:	With the Sequence parameter, Direction selects the
	homing method used.
Speed:	Speed used during the homing process.
Torque_Max:	Trip point at which Torque homing is triggered.
Output EN:	Rung enable.
Output ER:	An error has occurred, read the instance parameter
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode.
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled.
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= Recition out of range (max/min signed 22bit)
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3= Position out of range (max/min signed 32bit) E= Sequence is bound range [1, 4]
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For converse of a 1 or 2 - local capace for the
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14].
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3= Position out of range (max/min signed 32bit) 5= Sequence is beyond range [14] 7= For sequence = 1 or 2, legal range for the direction is [14]. 8= Speed must be > 0.
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3= Position out of range (max/min signed 32bit) 5= Sequence is beyond range [14] 7= For sequence = 1 or 2, legal range for the direction is [14]. 8= Speed must be > 0. 10= For Sequence=4 (Torque) the Torque Max must
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14]. 8 = Speed must be > 0. 10 = For Sequence=4 (Torque) the Torque_Max must be > 0.
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14]. 8 = Speed must be > 0. 10 = For Sequence=4 (Torque) the Torque_Max must be > 0. 100 = Error state is present in motor.
Output ER: Output DN:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3= Position out of range (max/min signed 32bit) 5= Sequence is beyond range [14] 7= For sequence = 1 or 2, legal range for the direction is [14]. 8= Speed must be > 0. 10= For Sequence=4 (Torque) the Torque_Max must be > 0. 100= Error state is present in motor. Instuction has completed successfully
Output ER: Output DN: Output IP:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14]. 8 = Speed must be > 0. 10 = For Sequence=4 (Torque) the Torque_Max must be > 0. 100= Error state is present in motor. Instuction has completed successfully In progress.
Output ER: Output DN: Output IP: Output IP: Output PC:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14]. 8 = Speed must be > 0. 10 = For Sequence=4 (Torque) the Torque_Max must be > 0. 100 = Error state is present in motor. Instuction has completed successfully In progress. Process completed.

	Author: HC File: LB0061-10GB	Date: 13-08-	2019
Intelligent motors	JVL Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

# Table of homing possibilities

Sequence	Direction	Description
1	0	Forward homing to switch, Uni direction. Stops on switch rising edge.
1	1	Forward homing to switch, Bi direction. Detects switch on rising edge and returns to find
		falling edge. This is a more accurate method than the Uni direction –method.
1	2	Reverse homing to switch, Uni direction. Stops on switch rising edge.
1	3	Reverse homing to switch, Bi direction. Detects switch on rising edge and returns to find
		falling edge. This is a more accurate method than the Uni direction –method.
2	0	Slow Forward home to marker (internal index pulse). Very accurate.
		Only works within one revolution of the motor shaft.
2	1	Fast Forward home to marker (internal index pulse). Less accurate.
		Only works within one revolution of the motor shaft.
2	2	Slow Reverse home to marker (internal index pulse). Very accurate.
		Only works within one revolution of the motor shaft.
2	3	Fast Reverse home to marker (internal index pulse). Less accurate.
		Only works within one revolution of the motor shaft.
4	0	Forward Torque homing. Trips when the actual torque > Torque_Max [%]
4	2	Reverse Torque homing. Trips when the actual torque > Torque_Max [%]

Please observe that for MAC50-141 – servo motors the Home to marker methods are not available.

	Author: HC File: LB0061-10GB	Date: 13-08-2	2019
intelligent mators	JVL Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### Supplied MAC AOI -Example

The example supplied shows a demo of all the JVL AOI's in a typical application using a linear actuator. When the example is loaded, all JVL AOI's are loaded automatically so there is no need for importing all the AOI's prior to loading the example.

In the example we use the metric units of [mm] and [mm/s] and limits the Max. Torque allowed to 123%.

#### Observe that the motor is capable of delivering 300% of the rated torque for a short period.

The example can be used to any JVL servomotor, just change the **LACT400\_CONF.Motortype** to an appropriate value for the motor type used.



#### The Config is called with the following parameters:

MotorType = 2 (MAC400)

MaxTorque = 123% (Peak is 300%)

units\_per\_Rev = 60.0 which is this case means 60.0mm/rev of the motorshaft. This is the linear spindle ratio.

GearRatio = 1.0, means 1:1 ratio, we dont have any gearbox attached.

UnitSelection = 2 means we use [mm] and [mm/s] in this application, Then all motion instruction will use these units in the motion profile settings.

#### Internal filter settings in the motor, these will vary with the application.

Load\_Ratio = 1.0, The motor will face an intertia load ratio 1:1, this value is used internally in the motor for the motion filter settings.

Servo\_Response = 4 Means we request a fast acting servo. The value goes from 1 to 4, where 4 = fast.

Load\_Coupling = 4, we request a rigid load coupling, value goes from 1 to 4.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

Please observe that a combination of high Load ratio and high servo response setting with high load coupling setting, could result in an unstable servo motor, use with care.

In the example the JVL\_Config\_Servo will be called once at the first PLC cycle. For a factory defaulted motor, please set the Save\_Volatile (from the JVL\_Com\_Servo -object) –flag to 1 and run the process again.

This should only be necessary the first time, then the motor should have been configured and all settings saved in volatile memory.

The program is controlled through a statemachine entering different states along with the process of the different AOI has completed the motion.

When the **START** –bit is set, the servo is energized and enabled through the JVL\_MSO –instruction and starts a homing procedure JVL\_MAH in the requested\_state = 100.



The homing procedure is configured for a torque reverse homing (Sequence = 4, Direction = 2) with a trip point of 50% torque. In other words the motor rotates CCW with the speed of 10mm/s to the mechanical end stop where it will trip on an applied torque exceeding 50%. This position is defined as Position = 0.

#### Note: We assume the motor is a basic MAC400 D2/5 –CAGM, supplied with a standard incremental encoder.

When the homing is done the requested\_state changes to 140 where the JVL\_MAM instruction is called.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	L Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

The MAM instruction moves the motor to the absolute position 750mm, running 10mm/s, acceleration is 100 mm/s/s. The Profile used (Profile = 1) means the it will control the Acceleration and deceleration with S-Curve settings, using the Jerk value of 128 (Max. is 256).



When the motor reaches the position 750mm the MAM instruction flags the .PC –bit and the requested\_state = 160.

	Motion of the servo. MAM commands is called. Move to the position 50 mm, event of the server of the server of the server with 00 mm/s. use the S-Curve acceleration/ordereleration with S-Curve setting 128 When motion is completed state is changed to 180. State = 140	
		Motion Axis Move
Equi	AXIS_OneSho	[2] JVL_MAM_Serve
Source A state	[010]	JVL_MAM_Servo LACT400_MAM1(EN)
180 + Source B 180		Axis LACT400_AXIS
		Position 50 -(ER)-
		Speed 10 (IP)
		Accel_Decel_Rate 100 -(PC)
		Profile 1
		Jerk 128
	MetionAste Process Con - The astis Is end Positi LACT400 MAL	love plete the an A1 PC Move
		Source 170
		Dest requested_state 180 •

In this state the motor moves to absolute position 50mm using the same parameters as in the previous state.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

Next state, requested\_state = 180, the motor moves to the position 500mm, speed = 20 mm/s, acceleration = 1000, still using S-Curve,

	Motion of the serve, MAM commands is called. More to the possible 700 mm, using 2000 mm/s, Acc/Dec 1000 counts/s (not scaled) use the S-Curve acceleration/deceleration with S-Curve setting 128 When motion is completed state is changed to 180. State = 140		
FOU	AVIC: Oraclas(0)	Mation Axis Mave	
Equal	[ONS]	- Motion Axis Move	
Source A state 180 •		Axis LACT400_AXIS	EN)—
Source B 170		MoveType 0	DN)
		Position 500 -CE	ER)—
		Speed 20 Cl	P)
		Accel_Decel_Rate 1000 -CF	PC)-
		Profile 1	
		Jerk 128	
	Motion Acia Move Process Complete - The axis is a the end Position LACT400_MANTPC	Move	
		Source 18	U
		Dest requested_state 18	:e 10 +

Next state is similar, the position is set to 50mm



	Author: HC	File: LB0061-10GB	<sup>Date:</sup> 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

In the next state the MoveType has changed, so the motor will move relative to the current position instead of moving to an absolute position.

Motion of th Move the distance 50 mm from AccDD use the S Vhen motion	servo. MAM commands is called. the current position (relative movement) MoveType = 1, using 20 mm/s, c 1000 counts/s (not scaled) Curve acceleration/deceleration th S-Curve setting 128. completed state is changed to 191.
EQU- Equal Source A state 180 • Source B 190	Motion Axis More       [ONS]     Motion Axis Motion       [ONS]     [ONS]       [ONS]     Motion Axis Motion       [ON]     [ON]       [ON]     [ON]
	Motion Axis Move Process Complete - The axis is at the end Position LACT400_IMAN1.PC - Move Source 192 Dest requested_state 180 •

The motor will in this case move 100mm relative to the current position. In case the position was set to -100mm the motor would move 100mm in the opposite direction.

The next state, requested\_state = 200, the JVL\_MAJ AOI is demonstrated.

	Motion of We will JOG the Remember this will set the motor into a	the servo. MAJ commands is called. e motor at 5 mm/s in the positive direc jogging state, to stop the motor issue	ction. a MAS comman	id afterwards.		
EQU Equal Source A state 180 + Source B 200		M J A C S S	Motio JVL_MA Motion Axis Jog IVL_MAJ_Servo Axis Direction Speed Accel_Decel_Rat	an Axis Jag NJ_Servo	(EN) (DN) (ER) (IP)	Move Source 2 Dest requested_sta 11

The motor will JOG forward, Speed is 5mm/s, Acceleration = 1. Observe that the MAJ (MotionAxisJog) needs to be stopped by issuing a MAS instruction.

In the example a 5000ms timer controls when the MAS instruction is issued. Afterwards it will return to the first motion state again requested\_state = 140.

Equal Source A state	Wait a few seconds before we issue a MAS -command to stop the motor from jogging.		TON
Source B 210	issue the MAS command which stops the motor from localing and	tmJOG2.DN	Antice and a second and a secon
	set state = 140 which starts the motion again all over.		20. m
Equal EQU Source A state Source B 220	(RES)	Motion Avis Stan Avit, MAS, Servo Avis Decel, Rate Motion Avis Stap Done Bit Set if the Instruction Executed without Errors	MOV MOV MOV MOV Source 140 Dest requested_state 180 •

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
Intelligent motors	JVI	Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## AOI for the MIS stepper motors

The AOI's for the MIS stepper series doesn't require an .EDS file in contrary to the MAC motors.

Since the MIS AOI's isn't based on an .EDS description there are some additional configuration AOI's that needs be called for configuration of a factory defaulted motor.

The supplied example goes through all of the necessary steps to get everything working, but a small step by step procedure is also supplies later in this manual, covering configuration from MacTalk.

The AOI's are very similar to the AOI's used in the MAC motors, however due to the differences in motor technology there are deviations between the AOI's.

## Setting up the project, the first steps

#### Message paths

When a motor is added to the project the IP address needs to be added to the **AOI\_Eth\_Path\_Define** –instruction in the **\_02\_Define\_MSG\_Path**.

This is essential for the configuration of a new motor.

The JVL\_MAH homing instruction also needs a separate message object passed and the message path needs to be defined.

AOI_Eth_Path_De	fine
Generate Ethernet Path to Devi	ce 📃 🚽
AOI_Eth_Path_Define MSG_Et	h_Path_Define 🛄 🕂 (EN)
Cfg_Controller_Type	1 €
Cfg_Slot_Number	0 🗲 🗮 (DN
Cfg_Front_Port_Number	0 ←
Inp_IP_Address_Octet_1	192 + -(ER)
Inp_IP_Address_Octet_2	168 🗲
Inp IP Address Octet 3	0 ←
Inp IP Address Octet 4	41 🗲
Out MSG Path	ISG Path.Path

This configuration is for a CompactLogix 5380, Configured with A1/A2 Dual IP.

#### Examples

The I/O configuration of the controller has only the local communication module:

Go to ti	he local communication module. - Go out the EtherNet/IP port	
	Go across the backplane	
Path: LocalENB, 2, LocalENB, 2,	10.10.10.10, 1, 0 10.10.10.10, 1, 0	Browse

Author: HC File: LB0061-10GB	Date: 13-08-	2019
JVL Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: <b>KRP</b>

**AOI** Description

## JVL\_MSO (**M**otion**S**ervo**O**n)

This instruction is based on the Rockwell command MSO, which basically energizes the stepper drive. In JVL terms it will put the motor into Position mode. This instruction is required to be called with success prior to other motion instructions to be called.

Executes on rising edge.



JVL_MSO: Axis:	The instance name, exc. "JVL_MSO_Axis_01" The Axis object of the motor.
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode.
Output DN:	4 = Motor is already enabled. Instuction has completed successfully

Author: HC	File: LB0061-10GB	Date: 13-08-	2019
JVI	Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_MSF (**M**otion**S**ervo**O**ff)

This instruction is based on the Rockwell command MSF, which disables the stepper drive and makes it passive. Prior of calling this instruction the JVL\_MSO must successfully have been called otherwise an error will be flagged.

Executes on rising edge.

Parameters:



JVL_MSF: Axis:	The instance name, exc. "JVL_MSF_Axis_01" The Axis object of the motor.
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode.
Output DN:	5 = Motor is not enabled. Issue MSO first. Instuction has completed successfully

## JVL\_MAJ (MotionAxisJog)

This instruction is based on the Rockwell command MAJ [MotionAxisJog], which starts jogging the servo drive in a certain direction with a certain speed. The jogging is stopped with the command JVL\_MAS. This instruction basically puts the motor in velocity mode using JVL terms.

Executes on rising edge.

Parameters:

Motion A	xis Jog	JVL_MAJ:	Th
JVL_W	40	Axis:	Th
Motion Axis Jog		Direction:	Di
JAL_WAY_YAF	AJ_Axis_01	H(EN)-	0:
Axis M	IS_Axis_01	(DN)	1:
Direction	1	(ER)— Speed: (IP)—	Ve
Speed	1000	Accell_Decel_rate:	Ac
			in
Accel_Decel_Rate	10		
		Output EN:	Ru
		Output ER:	Ar
			ER
			5 =
			13
			ΕX
			2=
			3=
			4=
		Output DN:	Ins
		Output IP:	M

JVL_MAJ: Axis: Direction:	The instance name, exc. "JVL_MAJ_Axis01" The Axis object of the motor. Direction of motion: 0: CW (Forward) 1: CCW	
Speed:	Velocity in the selected unit.	
Accell_Decel_rate:	Acceleration and Deceleration in native motor units [Cnt's/Sample <sup>2</sup> ]	
Output EN:	Rung enable.	
Output ER:	An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled (JVL_MSO). 13 = an extended errorcode is issued EXERR: 2= Direction is not 0 or 1 3= Speed < 0 (negative value entered) 4= Accel_Decel_Rate < 0 (negative value)	
Output DN: Output IP:	Instuction has completed successfully Motion is in progress. This will be held high until a MAS instruction is issued.	

26/36

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_MAS (MotionAxisStop)

This instruction is based on the Rockwell command MAS [MotionAxisStop], which stops any pending motion.

In case a JVL\_MAS instruction has been issued, the motors runs until the JVL\_MAS instruction is issued.

Executes on rising edge.

Motion Axis Stop	
Motion Axis Stop JVL_MAS JVL_MAS_Axis_01 Axis MIS_Axis_01 Decel_Rate 1000	(EN) -(DN) -(ER) -(IP) -(PC)

n and deceleration rate in the selected
e.
s occurred, read the instance parameter errorcode. s not enabled. ended errorcode is issued ecel Rate < 0 (negative value)
has completed successfully

	Author: HC	File: LB0061-10GB	<sup>Date:</sup> 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### JVL\_MAM (MotionAxisMove)

This instruction is based on the Rockwell command MAM [MotionAxisMove], which starts a move to a certain position either absolute or incremental with a

Executes on rising edge.

Parameters:

JVL_MAM Motion Axis Move JVL_MAM JVL_MAM_Axis Axis MIS_Axis MoveType Position 409	s_01(EN s_01 -(DN 1 -(ER =(IP) 9600 -(PC	JVL_MAM_Servo: Axis: MoveType: Position: Speed: Accell_Decel_rate:	The instance name, exc. "JVL_MAM_Axis_01" The Axis object of the motor. Absolute (=0) or Relative (=1) Position in the selected unit. Speed used in the selected unit Acceleration and deceleration rate in the selected unit.
Speed	100		
Accel_Decel_Rate	10	Output EN: Output ER:	Rung enable. An error has occurred, read the instance parame ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXERR: 2= MoveType out of range 3= Position out of range (max/min signed 32bit) 4= Speed < 0 (negative value)
		Output DN: Output IP: Output PC:	S= Acceil_Decel_Kate < 0 (negative value) Instuction has completed successfully In progress. Process completed.

In the example above the servo will move 409600 Units relative to the current position. It accelerate to speed of 100 units with acceleration setting 10 unit/s.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## JVL\_MAH (**M**otion**A**xis**H**ome)

This instruction is based on the Rockwell command MAH [MotionAxisHoming], which starts a homing sequence.

The basic ideer is that the motor either runs a sensor, Torque based or Encoder index -based homing sequence.

A sensor based homing triggers on the sensor input (AIN, see module specs for details) when it transitions either from high to low or low to high. The position is then preset to the value indicated in the "Position" –parameter.

The Torque homing method is less accurate due to tolerances in the mechanical construction and changes due to tear and wear. However when the motor detects an applied torque setting above the indicated in the Torque\_Max – parameter it will trigger and set the position to the indicated.

The last method triggers in the internal encoder index pulse and will thus be accurate within one revolution of the motor shaft. This method is very accurate however only usable within one shaft revolution.

Both the torque and the index – method eliminates the need for an external sensor.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	JVL Rockwell AOI (Add On Instructions)		Approved: KRP

The JVL\_MAM instruction must be installed with the JVL\_MAH instruction.

Executes on rising edge.

Motion Axis Home	-	2
.MI MAH	IVI MAH Avis 01	
Axis	MIS Axis 01	Conty
HomingMode	1 +	(DN)
Position	50.0 🗲	
Offset	50.0 🗲	(ER)-
Sequence	2 🗧	
LimitSwitchInputType	0 ←	(PC)
LimitSwitchInput	0 🕈	
Direction	2 🗲	
Speed	20.0 🗲	100000
ReturnSpeed	5.0 🗲	
TorqueMax	50.0 🗲	
Homing_MSG_Write_Value	MSG_MAH_Value_Axis_01	
	354 🗲	
Homing_MSG	MSG_MAH_Axis_01	

JVL_MAH_Servo: Axis: Position:	The instance name, exc. "JVL_MAH_Axis_01" The Axis object of the motor The reference position to set in the homing position.
HomingMode:	0= Passive, 1= Active. Selects the method used for the homing process
Sequence.	along with the Direction parameter. See the table for further details on which homing method to use and how to setup the Sequence and Direction parameters.
LimitSwitchInputTy	/pe:
LimitSwitchInput:	Normally Open = 0, Normally closed = 1.
	BOOL tag that must be set with the homing input. Currently only a PLC input tag. Unless the motor inputs are mapped into an Cyclic write word.
Direction:	With the Sequence parameter, Direction selects the homing method used.
Speed:	Speed used during the homing process.
Torque_Max:	Trip point at which Torque homing is triggered.
Homing_MSG_Wri	te_Value: DINT tag that holds a value for the Homing_MSG to send.
Homing_MSG:	Pointer to the homing message defined in the Controller tags –section.
Output EN: Output ER:	Rung enable. An error has occurred, read the instance parameter ERR for an errorcode. 5 = Motor is not enabled. 13 = an extended errorcode is issued EXER: 3 = Position out of range (max/min signed 32bit) 5 = Sequence is beyond range [14] 7 = For sequence = 1 or 2, legal range for the direction is [14]. 8 = Speed must be > 0. 10 = For Sequence=4 (Torque) the Torque_Max must be > 0.
	100= Error state is present in motor.
Output IP:	In progress.
Output PC:	Process completed.

	Author: HC File: LB0061-10GB	Date: 13-08-	2019
intelligent motors	JVL Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Table of homing possibilities, Active = 1

Sequence	Direction	Description
0	Х	Resets encoder position, without motor shaft turning, same as Active = 0.
1	0	Forward homing to switch, Uni direction. Stops on switch rising edge.
1	1	Forward homing to switch, Bi direction. Detects switch on rising edge and returns to find falling edge. This is a more accurate method than the Uni direction –method.
1	2	Reverse homing to switch, Uni direction. Stops on switch rising edge.
1	3	Reverse homing to switch, Bi direction. Detects switch on rising edge and returns to find falling edge. This is a more accurate method than the Uni direction –method.
2	0	Forward Torque homing. Stops on actual torque exceeding the Torque_Max –setting.
2	1	Forward Torque homing. Stops on actual torque exceeding the Torque_Max –setting.
2	2	Reverse Torque homing. Stops on actual torque exceeding the Torque_Max –setting.
2	3	Reverse Torque homing. Stops on actual torque exceeding the Torque_Max –setting.

#### X: Not relevant

For Active = 0 the same applies as if Sequence = 0. The encoder position is reset and the a new reference point for the motor will be set.

Due to the nature of the Homing sequence the message paths needs to be configured for the motor. The COP – instruction needs to be executed prior to executing the MAH –instruction.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JV	L Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Torque homing

Torque homing is only possible for motors equipped with the encoder option H2 or H4. This homing method requires that the mechanics of the application is able to stall the motor at the desired homing position.



The torque threshold value is set in the *TorqueMax* –tag the unit is [%]



### Sensor homing

Sensor homing requires a free input on the PLC that needs to activate the *LimitSwitchInput* –tag in the JVL\_MAH – instruction.

The example below shows how to setup a sensortype homing using Input 0 on the 5069-1B16 –input module on the PLC.



Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
JVI	- Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

## Supplied MIS AOI – Example

The supplied example for the MIS AOI usage is a demo case on how to configure and setup a project using the MIS AOI.

Loading the project will automatically include all AOI's and no real configuration is required besides the IP settings.

The Example is divided into 4 different routines besides the MainRoutine.



### \_01\_MainRoutine

The routine handles calls to the underlying sub routines.

	VAL MISROV AMURIC SAMPLE PROGRAM 2018 - ROCKWELL AUTOMATION CEN TEAM	
	Uses Centrol: EtherHettP Module for the Communication Lowest RPI for MSI JML Integrated DriveMotor is time	
0		-[NOP]-
1	Cal Routine 0 Sufficience O Sufficience O Sufficience O Sufficience O Sufficience O O Definie JM	3_Peth
2	Cal_Routine 1 JSR 	Module
3	Cal_Routine 2 Jsr. Jump To Subroutine Routine Valee _04,MS_CO	Drive
4	Call Routine 3	eration
(End)		

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
*			Revision:	Approved:
intelligent motors	JVL Rockwell AOI (Add On Instructions)		1.0	KRP

The CALL\_Routine –tag holds a bit for each routine to call.

For a new motor not previously configured the following routines needs to be called:

- 1. \_02\_Define\_MSG\_Path
- 2. \_03\_MIS\_Cfg\_Eth\_Module
- 3. \_04\_MIS\_Cfg\_Drive

When the routines has been called the settings are stored in volatile memory in the motor and the I/O should be established automatically. The I/O size doesn't match the motor before it is configured, so an I/O fault will be active.

### \_02\_Define\_MSG\_Path

This routine converts the IP –address into a message path that can be used in the different message object used.

The motor is by factory default configured with the IP address 192.168.0.xx, where xx is the last byte from the MAC-address.

The resulting path definition is placed in the MSG\_Path –tag.

#### \_03\_MIS\_Cfg\_Eth\_Module\_

This routine configures a factory defaulted module for with the right Cyclic setup and for 8x I/O usage.

The routine is using the MSG\_Path –tag defined in the \_02\_Define\_MSG\_Path\_ -routine.

#### \_04\_MIS\_Cfg\_Drive\_

Only relevant for MIS motors with the H2 and H4 encoder options.

This routine configures a factory default MIS stepper motor.

Since the release of V4.04 firmware the Closed loop and current control are enabled by default and this routine is obsolete.

	Author: HC	File: LB0061-10GB	Date: 13-08-2	2019
intelligent motors	JVI	Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

### \_10\_MIS\_Operation

This routine holds the calls for all the AOI motion instructions.

The JVL\_Comms –instruction handles the Data conversion and needs to be called in each program cycle.

xaccax COMMS xaccax	
Cyclic Read/Write to Drive	
	Communication
	JVL_Commo JVL_Comms_Axis_01 (EN)
	Axis MIS_Axis_01 (P)
	CyclicWrite MIS_Drive_01:O.Data

The JVL\_Scaling instruction will setup the scaling parameters according to the application.

	xxxxxx SCALING xxxxx	
	Setup Scaling of the Axis	
Asis Dela Drive is Mircuit Connends : Enabled ME_Ana_01 Sinch Enabled JT		Scaling Setup           VIL Scaling           - Scaling Setup           VIL Scaling           V

In this example the scaling has been setup for 78 units per revolution, no gearbox attached so gear ratio is 1:1 and the units used are [mm] and [mm/s].

So for a linear application this means that each motor revolution = 78 mm.

The homing procedure will be processed when the Axis\_01.Home –bit is set high.

2 Rungs are executed the JVL\_MAH and the message path are configured.

Local Controls -     Hore     Add Di tele      Manual Controls -     Hore     Manual Controls -     Manual Contro	
Marui Conends - Message b Holds Marui Conends - Message b Holds Marui Conends - Core	
Marual Connexts - Marual	
More     Homoget MA and Compared and the part of the part of the part of the part of the part of the part of the part of the part of the part of the part of the part of the part of the	
Macual Contends - Materia Asia, Di tane Mate	
Manual Connexts - Manual Connexts - Maton Axis Home Maton Axis Home Ma	7-
	ath ath
xxxxx MXMMG xxxxx           Mdxxx Axis Home           Home         Mdxxx Axis Home           Axis 01 Home         Mdxxx Axis Home           1         Mdxxx Axis Home           1         Mdxx Axis Home           1         Mdx Axis	<u> </u>
Manual Connexts - Monor Acits Home         Motor Acits Home	
Matca Aste Home         Matca Aste Home	
Matural Commends - Home         Modern Acits Home           Acits (D Home	
Marual Connavis - Motion Avia Hone	
More         Motor Asite Here           Ana_01 Here         Motor Anal Here           1         Motor Anal Here           Ana_01 Here         Motor Anal Here           Motor Anal Here	
Aug_0 I time         Mc/m.Mol           Item         Mc/m.Mol time	
المركب المرك مركب المركب المركب مركب المركب المر	
Atta Mon_Anti, Un homogliota 1 e homoton 200 e Sequencia ListSvictorputype 0 e ListSvictorputype 0 e	-(EN)
Fontion         50.0 +           Offer         60.0 +           Steparot         1 +           Steparot         1 +           Little         1 +	(DN)
001tet 2004 Segundo 14 LutSiveddraptype 04 LutSiveddraptype 04	
LeefSwitzhout yes 0 + LeefSwitzhout 0 +	-(ER)
LintSystemical 0+	(PC)
Direction 2 €	-(P)
Speed 20.0 +	
ReturnSpeed 5.0 +	
TransMax Homing JKS_VHts_Value_MSG_JMHT_Value_Kas	
To any HALL going Hall going to a set of the	

	Author: HC File: LB0061-10GB		Date: 13-08-2019	
intelligent motors	JVI	. Rockwell AOI (Add On Instructions)	Revision: 1.0	Approved: KRP

This page is left blank.