# Basic Control of SMVector over Ethernet IP using RSLogix<sup>™</sup>5000

This application example illustrates the basic control of an SMVector Drive over Ethernet IP using an RSLogix<sup>™</sup>5000 programmed PLC. A CompactLogix<sup>®</sup> controller was used for the development of this application and its export file is attached to this application note. The application cycles the drive between two moves: First a 60 second forward move at 40Hz with a 1 second acceleration and a 5 second deceleration, and then a second move for 30 seconds at 60Hz in the reverse direction with a 5 second acceleration and a 1 second deceleration.

This logic example uses Assemblies 100 and 101 to control the SMVector drive over Ethernet IP.

The controller tags in this project have aliasing to break out the relevant SMV data from the assemblies.

	Bit 0	0 = NOT Run Forward 1 = Run Forward						
	Bit 1	0 = NOT Run Reverse 1 = Run Reverse						
	Bit 2	Fault reset on transition	rom 0 to 1					
	Bit 3	Reserved						
	Bit 4	Reserved						
	Bit 5	0 = Local Control 1 = Network Control						
-	Bit 6	0 = Local Speed reference 1 = Network Speed reference						
rd C	Bit 7	Reserved						
Mo	Bit 8	Network Speed reference (valid when bit 6 set)						
	Bit 9	0 – Network	3 – 4-20mA	6 – Preset #3	9 – Preset #6			
	Bit 10	1 – keypad	4 – Preset #1	7 – Preset #4	10 – Preset #7			
	Bit 11	2 – 0-10VDC	5 – Preset #2	8 – Preset #5	11 – MOP			
	Bit 12	0 = No Action 1 = Inhibit (Coast to STOP)						
	Bit 13	0 = No Action 1 = Activate Quick STOP						
	Bit 14	0 = No Action 1 = Force Manual Mode (active only in Network Control, in PID mode will force open loop)						
	Bit 15	0 = DC brake active 1 = DC brake NOT active						
Word 1	Unsigned speed 0.1Hz resolution • received value = 0x01F0 = 49.6Hz							
Word 2	Digital Output + Relay – Active when parameter P140, P142 = 25 Network Control Bit 9 – Open Collector Bit 10 - Relay Others – reserved for future use							
Word 3	Analog Output [0 • received value	0.01VDC] – Active when p = 0x024B = 5.87[VDC]	parameter P150 = 9 Netv	work Control				

### Output Assembly 100 - Speed (Hz) & Digital and Analog Output

	Rit 0	1 – Foultod					
	DILU						
	Bit 1						
	Bit 2	1 = Running For	vard				
	Bit 3	1= Running Reverse					
	Bit 4	1 = Ready					
	Bit 5	0 = Local Control 1 = Control from Network					
0 p.	Bit 6	δ 0 = Local reference 1 = Reference from Network					
Moi	Bit 7	1 = At reference					
	Bit 8	Actual set point source:					
	Bit 9	0 – keypad	3 – Preset #1	6 – Preset #4	9 – Preset #7		
	Bit 10	1 – 0-10VDC	4 – Preset #2	7 – Preset #5	10 – MOP		
	Bit 11	2 – 4-20mA	5 – Preset #3	8 – Preset #6	11 – Network		
	Bit 12	1 = PID Active (c	losed loop)				
	Bit 13	1 = Torque mode	e active				
	Bit 14	1 = Current limit					
	Bit 15	1 = DC Braking					
Word 1	Unsigned actual frequency 0.1Hz resolution.						
Word 2	Digital Input/Output states ( See Note 1 for details)						
Word 3	Analog Input 0-10V TB • received value = 0x0	[0.01VDC] 24B = 5.87[VDC]					

## Input Assembly 101 - Speed (Hz) & Digital and Analog Input

#### The following parameters must be set first in the SMV:

- P100=3
- P101=6
- P111=2 to enable the deceleration logic to function
- P112=1
- P121=9
- P140=14
- P142=14
- P400=5
- P410-413 = 192.168.124.16
- P414-417 = 255.255.255.0
- Both TB1 and TB13A need to be asserted prior to running this logic.

Ensure the SMVector drive is power cycled after programming prior to running this PLC application to ensure proper operation.

#### To map the drive to an Ethernet IP scanner in RSLogix<sup>™</sup> 5000 for implicit messaging:

Click the [I/O Configuration] folder in the left-hand navigation window

Click the appropriate Ethernet Port folder, [1769-L32E Ethernet Port] in this example.

Right click on the [Ethernet] network icon and select [New Module].



Expand "Communications" and select [ETHERNET-MODULE Generic Ethernet Module].

Select Module		
Module     Des       … 1788-EN2DN/A     178       … 1788-ENBT/A     178       … 1788-ENBT/A     178       … 1788-EWEB/A     178       … 1794-AENT/A     179       … 1794-AENT/B     179       … Drivelogix5730 Eth     10/3       … ETHERNET-BRIDGE     Gen       … EtherNet/IP     Soft       … PH-PSSCENA/A     Ether       ●. Drives     ●. HMI	pription 8 Ethernet to DeviceNet Linking Device 9 10/100 Mbps Ethernet Bridge, Twisted-Pa 9 10/100 Mbps Ethernet Bridge w/Enhance 4 10/100 Mbps Ethernet Adapter, Twisted- 100 Mbps Ethernet Port on DriveLogix5730 eric EtherNet/IP CIP Bridge eric Ethernet Module 1 Logix5800 EtherNet/IP ernet Adapter, Twisted-Pair Media	Vendor Allen-Bradley Allen-Bradley d Web Serv Allen-Bradley Pair Media Allen-Bradley Pair Media Allen-Bradley Allen-Bradley Allen-Bradley Allen-Bradley Parker Hannif
By Category By Vendor	Favorites	Find Add Favorite
Type:       ETHERNET-MODULE         Vendor:       Allen-Bradley         Parent:       LocalENB         Name:       My_SMVector_Drive         Description:	Generic Ethernet Module Connection Parameters Assem Instan Input: 101 Uutput: 100 Configuration: 1 124 . 16 Status Input:	bly ce: Size: 4 • (16-bit) 4 • (16-bit) 0 • (8-bit)
Open Module Properties	ОК	Cancel Help

Enter a name for the drive -usually relating to the process (i.e. booster\_pump\_4, or an equipment tag number such as PP105).

Enter the IP address of the SMV drive. Ensure that it is on the same subnet as the PLC (the first 3 octets of the IP address match).

For this basic application, enter "Data – INT" for the Comm format.

Enter the desired Input and Output Assembly numbers and their corresponding lengths. Remember the size must be set to the number of words that actually make up the assembly you want to use.

For this example application we will use Assemblies 101 (Input) and 100 (Output).

For Configuration enter assembly instance 1 and a size of 0. This value is required.

Under the connection tag enter the desired RPI rate. This is how frequently the drive will be polled by the PLC. The minimum recommended value is 5.0 milliseconds.

From this screen you can also optionally set the controller to fault if the Ethernet IP connection is lost to the drive while the controller is running. This selection is the [Major Fault On Controller If Connection Fails While in Run Mode].

Module Properties: LocalENB (ETHERNET-MODULE 1.1)
General Connection* Module Info
Requested Pasket Interval (RPI): 50 th ma (1.0, 2200.0 ms)
Major Fault On Controller If Connection Fails While in Bun Mode
Module Fault
Status: Offline OK Cancel Apply Help

The corresponding tags will then be created in the controller tags of the project as shown herein.

Controller Tags - CompactLogix	c1(controller	)			
Scope: 🚺 CompactLogix1 💌 Sh	ow Show	All			
Name	Δ	Value 🔸	Force Mask 💦 🔦	Style	Data Type 📃 🔺
+-Assembly_20_data		{}	{}	Decimal	INT[1,2]
		{}	{}	Decimal	INT[1,2]
CMD_GetValue		1		Decimal	BOOL
CMD_SetValue		1		Decimal	BOOL
──		{}	{}		MESSAGE
		{}	{}		AB:ETHERNET_MODULE:C:0
		{}	{}		AB:ETHERNET_MODULE_INT
		{}	{}		AB:ETHERNET_MODULE_INT
		{}	{}		MESSAGE
⊞-SImpleServo:C		{}	{}		AB:ETHERNET_MODULE:C:0
⊞-SImpleServo:I		{}	{}		AB:ETHERNET_MODULE_REA
		{}	{}		AB:ETHERNET_MODULE_REA
		{}	{}		AB:ETHERNET_MODULE_DIN
		3		Decimal	INT
Value_Get		4.20389539		Float	REAL
──		3		Decimal	INT 🚽
Monitor Tags (Edit Tags /			•		▶ /

To understand this from the above configuration we named the drive "My\_SMVector\_Drive". There are three sets of tags labeled "My\_SMVector\_Drive":

[:C] for the Configuration assembly (1)

[:I] for the Input Assembly (101 in this example)

[:0] for the Output assembly (100 in this example)

Click on the [+] and expand the [My\_SMVector\_Drive:0] data to reveal all four words that make up the Output assembly.

Scope: CompactLogix1 - Show Sh	now All			
Name	🛆 Value 🛛 🗲	Force Mask 💦 🔶	Style	Data Type
	{}	{}	Decimal	INT[1,2]
	{}	{}	Decimal	INT[1,2]
CMD_GetValue	1		Decimal	BOOL
CMD_SetValue	1		Decimal	BOOL
	{}	{}		MESSAGE
⊞-My_SMVector_Drive:C	{}	{}		AB:ETHERNET_MODULE:C:0
⊞-My_SMVector_Drive:I	{}	{}		AB:ETHERNET_MODULE_INT_8
⊟-My_SMVector_Drive:0	{}	{}		AB:ETHERNET_MODULE_INT_8
Hy_SMVector_Drive:0.Data	{}	{}	Decimal	INT[4]
	0		Decimal	INT
	0		Decimal	INT
	0		Decimal	INT
Hy_SMVector_Drive:0.Data[3]	0		Decimal	INT
	{}	{}		MESSAGE
⊞-SImpleServo:C	{}	{}		AB:ETHERNET_MODULE:C:0
+-SImpleServo:I	{}	{}		AB:ETHERNET_MODULE_REAL
	{}	{}		AB:ETHERNET_MODULE_REAL
- SImpleServo:S	{}	{}		AB:ETHERNET_MODULE_DINT
Monitor Tags / Edit Tags /		•		•

6

Rung 0 illustrates the basic MSG instruction to write to the drive's Acceleration time (Parameter 104)

Configuration	Communication Tag		50		
Message Ty	vpe: CIP Generic		•		
Service [ Type:	Parameter Write	<u>*</u>	Source Element: Source Length:	SMV_accell_time	.▼
Service F Code: F	0 (Hex) Class: [f	(Hex)	Destination		•
Instance:	U4 Attribute:[]	(Hex)		New Tag	
Fnable	Enable Wating	) Start	Done	Done Length: 0	
<ul> <li>Error Code</li> </ul>	Extended E	Error Code:	1	Timed Out +	

Here we use the Examine if Off (Normally Closed Contact) expression evaluating the MSG's own enable bit so that it will continuously read from the SMV the Motor Current (Parameter 508)

	read motor current MSG.EN		MSG	
2	]/E	Message Message Control	read_motor_current_MSG	



Aessage Confi Configuration ( Message Type	guration - read Communication   : [CIP Gene	d_motor_curre Tag   mic	nt_MSG	
Service Par Type: Par Service e Code: e Instance: 508	ameter Read (Hex) Class: Attribu	▼  f (Hex) te: 1 (Hex)	Source Element: Source Length: Destination	0 (Bytes) SMV_motor_current ↓ New Tag
<ul> <li>Enable</li> <li>Error Code:</li> <li>Error Path:</li> <li>Error Text:</li> </ul>	Enable Waiting Exten	Start ded Error Code:	Done	Done Length: 0 IT Timed Out ←
		ОК	Cancel	Apply Help

This MSG changes the value of the drive's DECEL time (parameter 105)

Configurati	on Communication Ta	ig			
Message Service Type: Service Code	Type: CIP Generic Parameter Write	•	Source Element: Source Length: Destination	SMV_Decel_time	tes)
Instance:	105 Attribute:	1 (Hex)		New Tag	



2

SMV_Change_Decel_MSG.DN	Write_decel_trigger
	(U)

As the output frequency command in the drive assembly 100 is an integer value in tenths of a Hz (i.e. 412 =41.2 Hz), this MUL instruction converts the REAL format value (Source A) to the scaled integer value for use in the drive assembly by multiplying by 10 (Source B).

MUL
Source A SMV_Speed_Hz_command
Source B 10
Dest SMV_drive:O.Data[1] 400

As the drive's actual frequency in assembly 101 is an integer value in tenths of a Hz (i.e. 412 = 41.2 Hz), This DIV instruction converts the scaled integer value from the drive assembly (Source A) to a REAL value for use in the PLC by dividing by 10 (Source B).

6	Divide Source A SMV_drive:I.Data[1] 400
	Source B 10
	Dest SMV_Actual_Output_Frequency 0.0

This rung forces bits 5 and 6 of output assembly 100 to be on. This is required for network based start/stop and speed reference for the SMV.

7	SMV_Enable_Network_SpeedReference <smv_drive:o.data[0].6></smv_drive:o.data[0].6>	SMV_Enable_Network_StartandStop <smv_drive:o.data[0].5></smv_drive:o.data[0].5>
		0

This rung loads target velocity (to SMV\_Speed\_Hz\_command), then accel (to SMV\_accel\_time), and then decel (to SMV\_Decel\_time) values to create the first move profile of the program.

Note that the SMV\_Speed\_Hz\_command is an alias in the controller tag for Assembly 100 word 1



This rung loads target velocity (to SMV\_Speed\_Hz\_command), then accel (to SMV\_accel\_time), and then decel (to SMV\_Decel\_time) values to create the second move profile of the program.

Note that the SMV\_Speed\_Hz\_command is an alias in the controller tag for Assembly 100 word 1



Note here that we first check to ensure that the drive reports a READY status and ensures the drive is not faulted prior to starting a 5 second timer and initiating the process. The timer allows for more than ample time for the I/O to become active at the start of the process.

10	SMV_Ready_Status SMV_Faulted <smv_drive:i.data[0].4> <smv_drive:i.data[0].0></smv_drive:i.data[0].0></smv_drive:i.data[0].4>	Timor On Delay	
10		Timer Start_of_Process_Timer Preset 5000 Accum 0	-(DN)

Note that this logic checks to ensure that TB1 and TB13A are both asserted prior to starting the process. Both are required using the SMV Parameter settings listed at the top of this file.



Latch and Unlatch coils are used to control the program execution.



The .DN outputs of the MSG instructions are tested to ensure the accel and decel values have been loaded prior to starting this first move (forward).



This rung starts the 60 second timer governing the first move.

Note SMV\_at\_set\_speed status is aliased from Assembly 101 word 0 bit 7. This is checked to ensure the timer does not begin accumulating time until the drive has reached the commanded speed.



The .TT tag of the Running\_Fwd\_Timer is also used in this example to trigger the SMV drive's relay output. This output will be on while the drive is running forward at the set speed. It will not turn on until the acceleration is complete and will turn off prior to decelerating



Once the Running\_Fwd\_Timer times out, the drive is stopped and the second move profile is sent to the drive (reference rung 9).



The .DN outputs of the MSG instructions are tested to ensure the accel and decel values have been loaded prior to starting this second move (reverse).

17	Set_Second_Profile	SMV_set_accell_MSG.DN	SMV_Change_Decel_MSG.DN	SMV_Run_Rev <smv_drive:o.data[0].1></smv_drive:o.data[0].1>
17			1	

This rung starts the 30 second timer governing the second move.

Note SMV\_at\_set\_speed status is aliased from Assembly 101 word 0 bit 7. This is checked to ensure the timer does not begin accumulating time until the drive has reached the commanded speed.



The .TT tag of the Running\_Rev\_Timer is also used in this example to trigger the SMV drive's open collector output. This output will be on while the drive is running reverse at the set speed. It will not turn on until the acceleration is complete and will turn off prior to decelerating

19	Running_Rev_Timer.TT	<smv_oc_output <smv_drive:o.data[2].9></smv_drive:o.data[2].9></smv_oc_output 
	2 B	~ 2

Once the Running\_Rev\_Timer times out, the drive is stopped and the process is repeated (reference rung 12).



In the event of a fault we turn off all process flow logic and enable TB13C to be read to reset the drive fault. The SMV\_Fault\_reset tag is aliased from Assembly 0 word 0 bit 2.



Rockwell Automation®, CompactLogix®, RSLogix<sup>TM</sup> and RSLogix<sup>TM</sup> 5000 are trademarks owned and used by Rockwell Automation and its various subsidiary entities.