

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación



**LABORATORIO DE COMUNICACIONES INDUSTRIALES
Y SISTEMAS SCADA**

Tema de Guía: Mensajería Explícita PowerFlex 525

Autor:

Arturo Freire Veliz

GUAYAQUIL – ECUADOR

I PAO 2025

Table of Contents

1. Objetivos	3
1.1. Objetivo general	3
1.2. Objetivos específicos	3
2. Equipos y herramientas	3
3. Marco teórico	4
3.1. Mensajería Explícita.....	4
3.2. Proceso detrás de la Mensajería Explícita.....	4
3.3. Parámetros del Adaptador.....	5
3.4. Objetos Ethernet/IP.....	7
3.4.1. Objeto Identidad	7
3.4.2. Objeto Parámetro DPI.....	8
4. Arquitectura de Comunicación.....	12
5. Procedimiento	12
5.1. Lectura de un solo parámetro	17
5.1.1. Lectura del código de fallo	18
5.1.2. Lectura del Product Name	19
5.2. Lectura de varios parámetros	22
5.3. Escritura de un solo parámetro.....	24
5.4. Escritura de varios parámetros.....	25

Guía Teórica-Práctica

Tema: Mensajería Explícita PowerFlex 525

1. Objetivos

1.1. Objetivo general

Implementar una práctica de comunicación industrial orientada al uso de mensajería explícita sobre el protocolo CIP, con el fin de acceder a parámetros del variador de frecuencia PowerFlex 525, utilizando controladores Allen-Bradley y comprendiendo el uso de los objetos estándar definidos por la arquitectura EtherNet/IP.

1.2. Objetivos específicos

- Analizar los EtherNet/IP Objects aplicables al PowerFlex 525.
- Configurar mensajes explícitos en Studio 5000 Logix Designer para realizar operaciones de lectura y escritura sobre parámetros seleccionados del variador PowerFlex 525.

2. Equipos y herramientas

- Studio 5000 Logix Designer.
- RSLinx Classic o FactoryTalk Linx.
- Cables Ethernet.
- Stratix 5700 / 8300.
- PCs.
- CompactLogix 1769-L33ERM.
- ControlLogix L73 y Módulo 1756-EN2TR/C.
- Manuales de Programación: PowerFlex 525 Embedded EtherNet/IP Adapter.



3. Marco teórico

3.1. Mensajería Explícita

Los mensajes explícitos contienen la *información de dirección* y de *servicio* que lleva al dispositivo receptor a realizar cierto servicio (una **acción**) en una parte específica de un dispositivo (por ejemplo, un **atributo**).

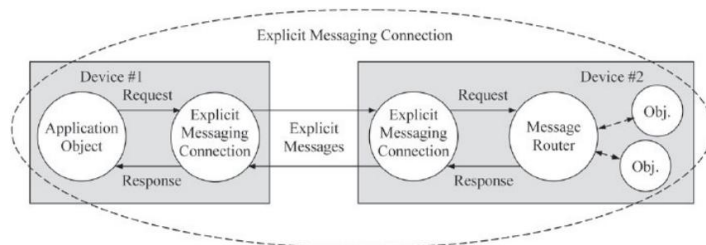


Figura 1. Conexión CIP para Mensajería Explícita punto a punto.

La mensajería explícita se utiliza para transferir información *que no necesita actualizaciones recurrentes*. Con esto se puede configurar y monitorear los parámetros de un dispositivo en la red.

Nota: El drive PowerFlex 525 presenta *limitaciones* de mensajería explícita. A continuación, la compatibilidad de códigos de clase para el drive:

Explicit Messaging Class Code Compatibility with PowerFlex 525 Drives

EtherNet/IP Object Class Code	Compatibility	Explicit Messaging Function
Parameter Object 0x0F	Yes	Single parameter reads/write
DPI Parameter Object 0x93	Yes	Single and scattered parameter reads/write

Figura 2. Compatibilidad de Códigos de Clase para Mensajería Explícita PF525.

3.2. Proceso detrás de la Mensajería Explícita

Existen cinco eventos básicos en el proceso de mensajería explícita. Recordar que debe **siempre existir un mensaje de solicitud y uno de respuesta** para cualquier mensaje genérico, ya sea que se realice una operación de lectura o escritura de información.

Explicit Messaging Process

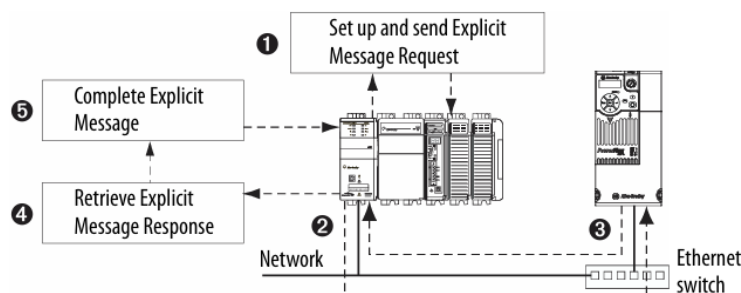


Figura 3. Proceso detrás de la Mensajería Explícita.

Paso 1: Se formatea la información requerida y se establece una lógica de programación para enviar una solicitud de mensaje explícito al escáner (puerto ethernet embebido) o módulo de puente (tarjeta de red). Esto sería el proceso de **Download**.

Paso 2: El escáner o módulo de puente transmite la solicitud de mensaje explícito al dispositivo “esclavo” en la red.

Paso 3: El dispositivo esclavo transmite la respuesta de mensaje explícito de retorno al escáner o módulo de puente. La información recibida se almacena en el *buffer del escáner*.

Paso 4: El controlador recupera la respuesta de mensaje explícito del buffer del escáner. Esto sería el proceso de **Upload**.

Paso 5: El mensaje explícito se ha completado.

Nota: Es recomendable revisar *la máxima cantidad de mensajes explícitos que pueden ejecutarse en un mismo tiempo*. Para esto se examina la documentación del escáner o módulo de puente respectivo del controlador utilizado.

3.3. Parámetros del Adaptador

Dado que se pretende realizar una comunicación explícita a través de Ethernet/IP, se deben tener presentes los parámetros del Grupo de Comunicación (Prefijo C).

Parameter		
No.	Name and Description	Details
C121	[Comm Write Mode] Saves parameter values in active drive memory (RAM) or in drive non-volatile memory (EEPROM). Important: Parameter values set prior to setting 1 “RAM only” are saved in RAM.	Default: 0 = Save Values: 0 = Save 1 = RAM only Type: Read/Write Reset Required:
C122	[Cmd Stat Select] Selects velocity-specific or position/fibers-specific Command and Status Word bit definitions for use over a communication network. This parameter cannot be changed when an I/O connection is established through the communication adapter or the drive’s embedded EtherNet/IP port.	Default: 0 = Velocity Values: 0 = Velocity 1 = Position Type: Read/Write Reset Required: Yes

Figura 4. Parámetros de Comunicación PF525.

C128	[EN Addr Sel] Enables BOOTP to set the IP address, subnet mask and gateway address with a BOOTP server. Identifies the connections that would be attempted on a reset or power cycle. A reset or power cycle is required after selection is made.	Default: 2 = BOOTP Values: 1 = Parameters 2 = BOOTP Type: Read/Write Reset Required: Yes
C129	[EN IP Addr Cfg 1]	Default: 0
C130	[EN IP Addr Cfg 2]	Default: 0
C131	[EN IP Addr Cfg 3]	Default: 0
C132	[EN IP Addr Cfg 4] Sets the bytes in the IP address. A reset or power cycle is required after selection is made. 192.168.1.62 [EN IP Addr Cfg 1] [EN IP Addr Cfg 2] [EN IP Addr Cfg 3] [EN IP Addr Cfg 4] Important: To set the IP address using these parameters, C128 [EN Addr Sel] must be set to 1 "Parameters".	Default: 0 Minimum: 0 Maximum: 255 Type: Read/Write Reset Required: Yes
C133	[EN Subnet Cfg 1]	Default: 0
C134	[EN Subnet Cfg 2]	Default: 0
C135	[EN Subnet Cfg 3]	Default: 0
C136	[EN Subnet Cfg 4] Sets the bytes of the subnet mask. A reset or power cycle is required after selection is made. 255.255.255.0 [EN Subnet Cfg 1] [EN Subnet Cfg 2] [EN Subnet Cfg 3] [EN Subnet Cfg 4] Important: To set the subnet mask using these parameters, C128 [EN Addr Sel] must be set to 1 "Parameters".	Default: 0 Minimum: 0 Maximum: 255 Type: Read/Write Reset Required: Yes
C137	[EN Gateway Cfg 1]	Default: 0
C138	[EN Gateway Cfg 2]	Default: 0
C139	[EN Gateway Cfg 3]	Default: 0
C140	[EN Gateway Cfg 4] Sets the bytes of the gateway address. A reset or power cycle is required after selection is made. 192.168.1.1 [EN Gateway Cfg 1] [EN Gateway Cfg 2] [EN Gateway Cfg 3] [EN Gateway Cfg 4] Important: To set the gateway address using these parameters, C128 [EN Addr Sel] must be set to 1 "Parameters".	Default: 0 Minimum: 0 Maximum: 255 Type: Read/Write Reset Required: Yes

Figura 5. Parámetros de Comunicación PF525.

3.4. Objetos Ethernet/IP

En este apartado se presentará información de algunos de los Objetos Ethernet/IP que pueden ser accedidos mediante mensajería explícita (revisar el manual para poder ver la lista completa).

3.4.1. Objeto Identidad

Class Code

Hexadecimal	Decimal
0x01	1

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x01	Yes	Yes	Get_Attribute_All

Figura 6. Código de Clase y Servicios del Objeto Identidad.

Instances (Single-Drive)

Instance, Ins	DPI Inst#	Port#	Device
0x0000...0x3FFF	Ins	0	Drive
0x4000...0x43FF	Ins & 0x3FF	Interface	Interface ⁽¹⁾
0x4400...0x47FF	Ins & 0x3FF	1	DSI
0x4800...0x4BFF	Ins & 0x3FF	2	Option
0x4C00...0x4FFF	—	—	—
0x5000...0x53FF	—	—	—
0x5400...0x57FF	—	—	—

(1) Interface is the current interface being used to access the information.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Max Instance	UINT	Total number of instances

Figura 7. Instancias y Atributos de Clase del Objeto Identidad.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Vendor ID	UINT	1 = Allen-Bradley
2	Get	Device Type	UINT	150
3	Get	Product Code	UINT	Number identifying product name and rating
4	Get	Revision: Major Minor	STRUCT of: USINT USINT	Value varies Value varies
5	Get	Status	UINT	Bit 0 = Owned Bit 8 = Minor recoverable fault Bit 10 = Major recoverable fault
6	Get	Serial Number	UDINT	Unique 32-bit number
7	Get	Product Name	SHORT_STRING	Product name and rating
9	Get	Configuration Consistency Value	UNIT	Current Parameter NVS CRS value

Figura 8. Atributos de Instancia del Objeto Identidad.

3.4.2. Objeto Parámetro DPI

Class Code

Hexadecimal	Decimal
0x93	147

Instances

The number of instances depends on the number of components in the device.
The total number of components can be read in Instance 0, Attribute 0.

Instances (Hex.)	Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
0x0000...0x3FFF	0...16383	Instances 0...1023 in the drive	Instances 0...1023 in Drive 0
0x4000...0x43FF	16384...17407	Instances 0...1023 in the interface ⁽¹⁾	Instances 0...1023 in the interface ⁽¹⁾
0x4400...0x47FF	17408...18431	DSI	Instances 0...1023 in Drive 1
0x4800...0x4BFF	18432...19455	Option	Instances 0...1023 in Drive 2
0x4C00...0x4FFF	19456...20479	Not supported	Instances 0...1023 in Drive 3
0x5000...0x53FF	20480...21503	Not supported	Instances 0...1023 in Drive 4
0x5400...0x57FF	21504...22527	Not supported	Instances 0...1023 in the Option

(1) Interface is the current interface being used to access the information. Note: If using the embedded EtherNet/IP adapter, this instance range contains the same data as Drive 0.

Figura 9. Código de Clase e Instancias del Objeto Parámetro DPI.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	UINT	Number of parameters in the device
1	Set	Write Protect Password	UINT	0 = Password disabled n = Password value
2	Set	NVS Command Write	USINT	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory 4 = Partial defaults 5 = System defaults
3	Get	NVS Parameter Value Checksum	UINT	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	UINT	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	UINT	First parameter available if parameters are protected by passwords. A "0" indicates all parameters are protected.
7	Get	Class Revision	UINT	2 = DSI
8	Get	First Parameter Processing Error	UINT	The first parameter that has been written with a value outside of its range. A "0" indicates no errors.
9	Set	Link Command	USINT	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

Figura 10. Atributos de Clase del Objeto Parámetro DPI.

Instance Attribute

Attribute ID	Access Rule	Name	Data Type	Description
6	Get	DPI Offline Read Full	STRUCT of: BOOL[32] CONTAINER CONTAINER CONTAINER STRING[16] STRING[4] UINT UINT UINT UINT UINT UINT USINT USINT UINT CONTAINER UINT UNIT UNIT INT	Descriptor Offline Minimum value Offline Maximum value Offline Default value Parameter name Offline parameter units Online minimum parameter instance Online maximum parameter instance Online default parameter instance Multiplier parameter instance Divisor parameter instance Base parameter instance Offset parameter instance Formula number Pad byte (always zero) Help instance Pad word (always a value of zero) Parameter value Multiplier Divisor Base Offset

Figura 11. Atributos de Instancia del Objeto Parámetro DPI.

7	Get	DPI Online Read Full	STRUCT of: BOOL[32] CONTAINER ⁽¹⁾ CONTAINER CONTAINER CONTAINER UINT UINT STRING[4] UINT UINT UINT INT USINT[3] USINT STRING[16]	Descriptor (see page 144) Parameter value Minimum value Maximum value Default value Next parameter Previous parameter Units (for example, Amps, Hz) Multiplier ⁽²⁾ Divisor ⁽²⁾ Base ⁽²⁾ Offset ⁽²⁾ Link (source of the value) (0 = no link) Always zero (0) Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (see page 144)
9	Get/Set	DPI Parameter Value	Various	Name of the component with support for Unicode.
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in NVS. ⁽³⁾
11	Get/Set	DPI Link	USINT[3]	Parameter value in temporary memory. Valid only for DSI drives.
12	Get	Help Object Instance	UINT	Link (parameter or function block that is the source of the value) (0 = no link)
13	Get	DPI Read Basic	STRUCT of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER STRING[16] STRING[4]	Descriptor (see page 144) Parameter value Minimum value Maximum value Default value Parameter name Units (for example, Amps, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer-supplied parameter name.

Attribute ID	Access Rule	Name	Data Type	Description
16	Get	Parameter Processing Error	USINT	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum
18	Get	International DPI Offline Parameter Text	Struct of: STRINGN STRINGN	International parameter name International offline units
19	Get	International DPI Online Parameter Text	Struct of: STRINGN STRINGN	International parameter name International online units
20	Get	International DPI Online Read Full	Struct of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT INT USINT[3] USINT BOOL[32] STRINGN STRINGN	Descriptor Parameter value Online minimum value Online maximum value Online default value Next Previous Multiplier Divisor Base Offset Link Pad word (always zero) Extended descriptor International parameter name International online parameter unit
21	Get	DPI Extended Descriptor	UDINT	Extended Descriptor (see page 145)

Figura 12. Atributos de Instancia del Objeto Parámetro DPI.

22	Get	International DPI Offline Read Full	Struct of: BOOL CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT UINT USINT USINT UINT INT CONTAINER UINT UINT INT BOOL[32] STRINGN STRINGN	Descriptor Offline minimum value Offline maximum value Offline default value Online minimum parameter instance Online maximum parameter instance Online default parameter instance Multiplier parameter instance Divisor parameter instance Base parameter instance Offset parameter instance Formula number Pad word (always zero) Help instance Pad word (always a value of zero) Parameter value Multiplier Divisor Base Offset Extended DSI descriptor International DSI parameter name International DSI offline parameter units
----	-----	-------------------------------------	--	---

Figura 13. Atributos de Instancia del Objeto Parámetro DPI.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Implemented for:		Service Name	Allocation Size (in bytes)	
	Class	Instance		Par. Number	Par. Value
0x32	Yes	No	Get_Attributes_Scattered	4	4
0x34	Yes	Yes	Set_Attributes_Scattered	4	4

The table below lists the parameters for the `Get_Attributes_Scattered` and `Set_Attributes_Scattered` object-specific service:

Name	Data Type	Description
Parameter Number	UDINT	Parameter to read or write
Parameter Value	UDINT	Parameter value write (zero when reading)

The response data appears in the following format:

Name	Data Type	Description
Parameter Number	UDINT	Parameter read or write ⁽¹⁾
Parameter Value	UDINT	Parameter value read (zero when writing) ⁽²⁾

(1) If an error occurred, bit 15 will be turned on in the response.

- (1) If an error occurred, bit 15 will be turned on in the response.
- (2) If an error occurred, the error code will appear instead of the value.

Figura 14. Servicios Comunes y de Objetos Específicos del Objeto Parámetro DPI.

4. Arquitectura de Comunicación

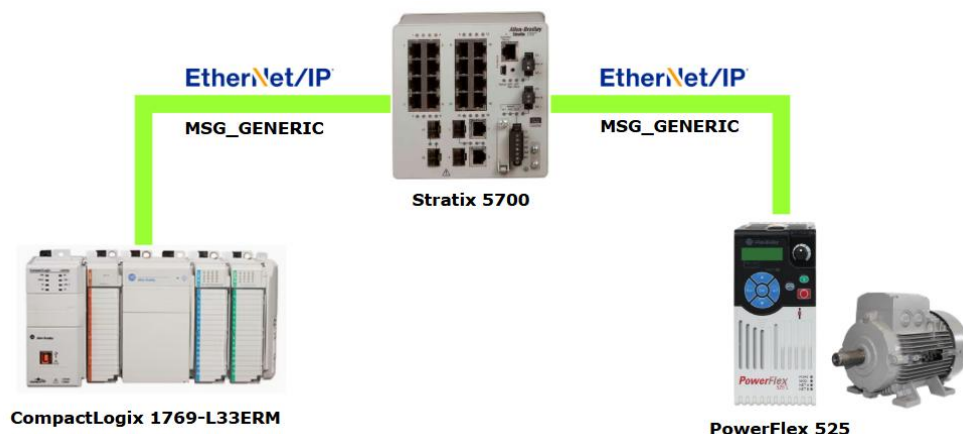


Figura 15. Arquitectura Protocolo CIP para Mensajería Explícita.

5. Procedimiento

En este apartado se explicará la configuración de la instrucción de mensajería para realizar operaciones de lectura y escritura de forma explícita entre un PLC CompactLogix y el Drive PowerFlex525 basada en la teoría de modelado de objetos.

Previo a explicar las distintas configuraciones de esta práctica, debería de tener ya agregado en el árbol de dispositivos del proyecto al Drive PowerFlex525, el cual debe encontrarse correctamente configurado y parametrizado según la aplicación. Si todavía no ha realizado dicho procedimiento, puede revisar la documentación de la Práctica 3.

A continuación, se mostrarán ciertas capturas de cómo debería encontrarse configurado y parametrizado el Drive para un Motor Siemens 1LA7 070-4YA60.

Tipo	Frame IEC	Potencia	Factor de Servicio	Corriente Nominal [A]	Eficiencia η	Factor de Potencia	Velocidad Nominal	Torque Nominal	Torque de Arranque	Corriente de Arranque	Peso Neto IMB3	Rodamientos		
		HP	Kw	440V	220V	(%)	Cos ϕ	(rpm)	[Nm]	(Ta/Tn)	(Ia/In)	[kg]	AS/BS	
MOTORES DE 4 POLOS, 1800 RPM														
1LA7 070-4YC60	71	0.4	0.3	1.05	1.6	0.8	63.3	0.77	1640	1.7	1.8	2.8	4.7	6202 2Z C3 / 6202 2Z C3
1LA7 070-4YA60	71	0.5	0.4	1.15	1.9	0.95	63.6	0.81	1590	2.24	1.3	2.7	4.7	6202 2Z C3 / 6202 2Z C3
1LA7 071-4YA60	71	0.6	0.4	1.15	2.2	1.1	70.2	0.76	1680	2.54	1.8	3.4	6.0	6202 2Z C3 / 6202 2Z C3

Figura 16. Datos de Placa Motor Siemens 1LA7 070-4YA60.

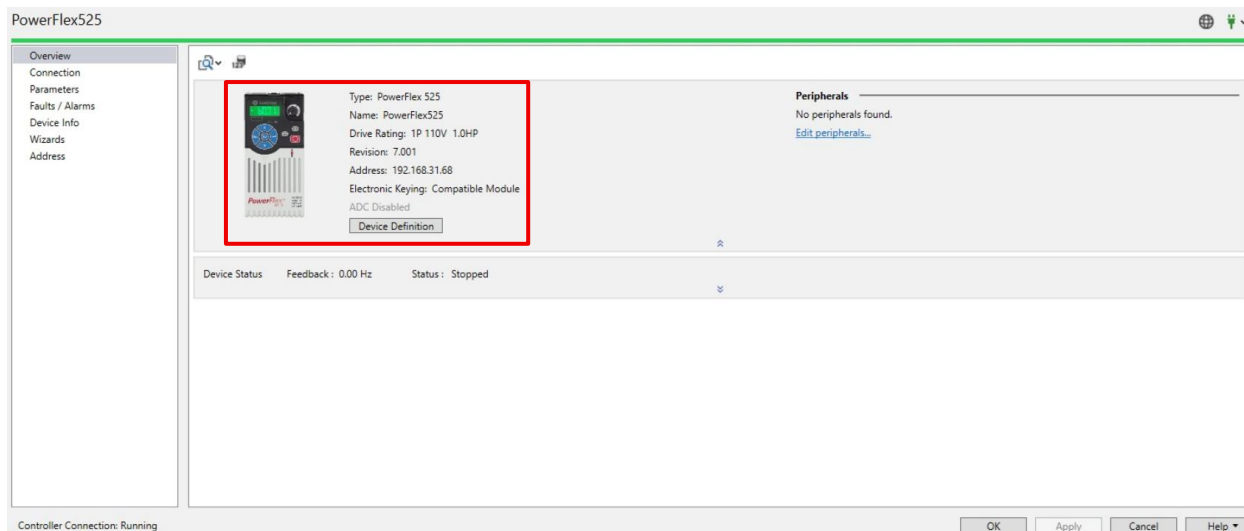


Figura 17. PowerFlex525 Overview.

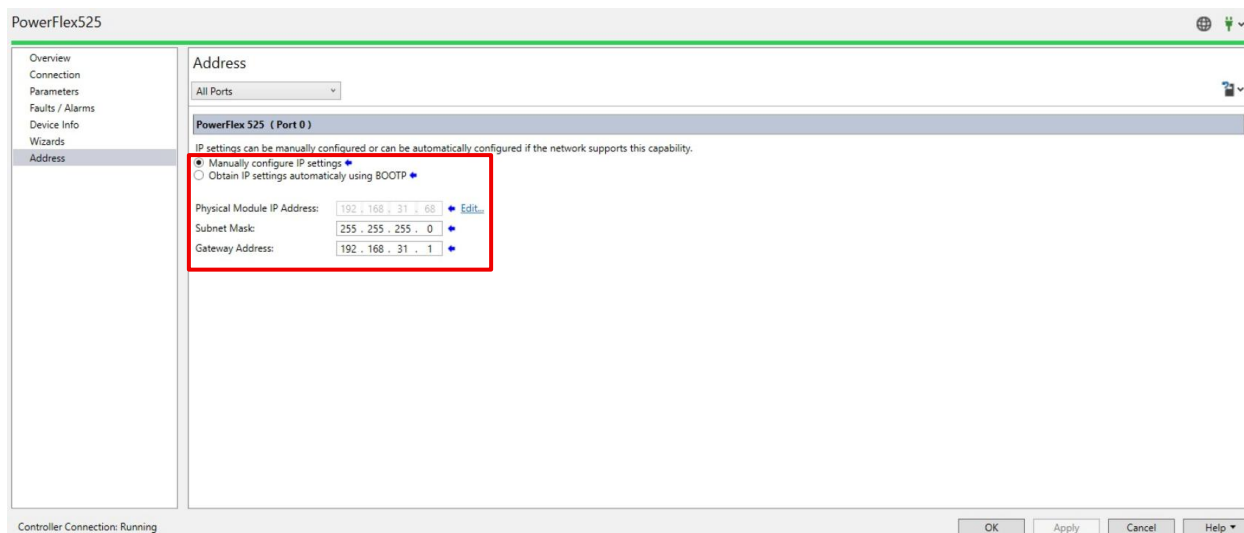


Figura 18. PowerFlex525 Address.

Device Definition

Upload Import Export

Identity
Peripherals
Connection Format
Automatic Device Configuration

Connection Format

Mode: Velocity

Input Output

Data Type Name: AB:PowerFlex525V_E_8A02D441:I:0 Data Type Size: 16 Bytes

Members:

Name	Data Type	Parameter
DriveStatus	INT	
OutputFreq	INT	
OutputCurrent	INT	Port 0: Output Current
OutputVoltage	INT	Port 0: Output Voltage
CommandedFreq	INT	Port 0: Commanded Freq

OK Cancel Help

Figura 19. PowerFlex525 Input Format.

Device Definition

Upload Import Export

Identity
Peripherals
Connection Format
Automatic Device Configuration

Connection Format

Mode: Velocity

Input Output

Data Type Name: AB:PowerFlex525V_E_54838AFD:O:0 Data Type Size: 8 Bytes

Members:

Name	Data Type	Parameter
LogicCommand	INT	
FreqCommand	INT	
AccelTime1	INT	Port 0: Accel Time 1
DecelTime1	INT	Port 0: Decel Time 1

OK Cancel Help

Figura 20. PowerFlex525 Output Format.

Nota: El “Connection Format” permite una lectura y escritura directa desde las variables globales del proyecto (Controller Tags). Sin embargo, la cantidad de parámetros por formato *es limitada*.

PowerFlex525

Overview	Parameters									
Connection										
Parameters										
Faults / Alarms										
Device Info										
Wizards										
Address										

Parameters										
0 - PowerFlex 525		All Parameters		Show Non-Defaults		Filter Value				
Port	#	Name	Value	Units	Internal Value	Default	Min	Max		
0	25	Accum Cost Sav	0.0		0	0.0	0.0	6553.5	...	^
0	26	Accum CO2 Sav	2.0	kg	20	0.0	0.0	6553.5	...	
0	27	Drive Temp	30	C	30	0	0	120	...	
0	28	Control Temp	50	C	50	0	0	120	...	
0	29	Control SW Ver	7.001		7001	0.000	0.000	65.535	...	
0	30	Language	English		1	English	1	15	...	
0	31	Motor NP Volts	220	V	220	230	20	230	...	
0	32	Motor NP Hertz	60	Hz	60	60	15	500	...	
0	33	Motor OL Current	2.3	A	23	2.5	0.0	5.0	...	
0	34	Motor NP FLA	1.9	A	19	1.7	0.1	5.0	...	
0	35	Motor NP Poles	4		4	4	2	40	...	
0	36	Motor NP RPM	1590	RPM	1590	1750	0	24000	...	
0	37	Motor NP Power	0.37	kW	37	0.40	0.00	655.35	...	
0	38	Reserved	0		0	0	0	0	...	
0	39	Torque Perf Mode	SVC		1	SVC	0	5	...	
0	40	Autotune	Ready/Idle		0	Ready/Idle	0	2	...	
0	41	Accel Time 1	10.00	Sec	1000	10.00	0.00	600.00	...	
0	42	Decel Time 1	10.00	Sec	1000	10.00	0.00	600.00	...	
0	43	Minimum Freq	0.00	Hz	0	0.00	0.00	500.00	...	

Figura 21. PowerFlex525 Motor Parameters.

PowerFlex525

Overview	Parameters									
Connection										
Parameters										
Faults / Alarms										
Device Info										
Wizards										
Address										

Parameters										
0 - PowerFlex 525		All Parameters		Show Non-Defaults		Filter Value				
Port	#	Name	Value	Units	Internal Value	Default	Min	Max		
0	37	Motor NP Power	0.37	kW	37	0.40	0.00	655.35	...	^
0	38	Reserved	0		0	0	0	0	...	
0	39	Torque Perf Mode	SVC		1	SVC	0	5	...	
0	40	Autotune	Ready/Idle		0	Ready/Idle	0	2	...	
0	41	Accel Time 1	10.00	Sec	1000	10.00	0.00	600.00	...	
0	42	Decel Time 1	10.00	Sec	1000	10.00	0.00	600.00	...	
0	43	Minimum Freq	0.00	Hz	0	0.00	0.00	500.00	...	
0	44	Maximum Freq	60.00	Hz	6000	60.00	0.00	500.00	...	
0	45	Stop Mode	Ramp, CF		0	Ramp, CF	0	11	...	
0	46	Start Source 1	EtherNet/IP		5	Keypad	1	5	...	
0	47	Speed Reference1	EtherNet/IP		15	Drive Pot	1	16	...	
0	48	Start Source 2	DigIn TrmBlk		2	DigIn TrmBlk	1	5	...	
0	49	Speed Reference2	0-10V input		5	0-10V input	1	16	...	
0	50	Start Source 3	EtherNet/IP		5	EtherNet/IP	1	5	...	
0	51	Speed Reference3	EtherNet/IP		15	EtherNet/IP	1	16	...	
0	52	Average kWh Cost	0.00		0	0.00	0.00	655.35	...	
0	53	Reset To Defaults	Ready/Idle		0	Ready/Idle	0	4	...	
0	54	Display Param	Keypad Disp		0	Keypad Disp	0	54	...	
0	55	Reserved	0		0	0	0	0	...	

Figura 22. PowerFlex525 Source Parameters.

PowerFlex525

Port	#	Name	Value	Units	Internal Value	Default	Min	Max
0	127	RS485 Format	RTU 8-N-1		0	RTU 8-N-1	0	5
0	128	EN Addr Sel	Parameters		1	BOOTP	1	2
0	129	EN IP Addr Cfg 1	192		192	0	0	255
0	130	EN IP Addr Cfg 2	168		168	0	0	255
0	131	EN IP Addr Cfg 3	31		31	0	0	255
0	132	EN IP Addr Cfg 4	68		68	0	0	255
0	133	EN Subnet Cfg 1	255		255	0	0	255
0	134	EN Subnet Cfg 2	255		255	0	0	255
0	135	EN Subnet Cfg 3	255		255	0	0	255
0	136	EN Subnet Cfg 4	0		0	0	0	255
0	137	EN Gateway Cfg 1	192		192	0	0	255
0	138	EN Gateway Cfg 2	168		168	0	0	255
0	139	EN Gateway Cfg 3	31		31	0	0	255
0	140	EN Gateway Cfg 4	1		1	0	0	255
0	141	EN Rate Cfg	Auto detect		0	Auto detect	0	4
0	142	Reserved	0		0	0	0	0
0	143	EN Comm Flt Actn	Fault		0	Fault	0	4
0	144	EN Idle Flt Actn	Fault		0	Fault	0	4
0	145	EN Flt Cfg Loaic	00000000.00000000		0	00000000.00000000		

Figura 25. PowerFlex525 Communication Parameters.

Luego de haber configurado y parametrizado correctamente el Drive, implementar la siguiente lógica de mensajería periódica en la rutina principal:

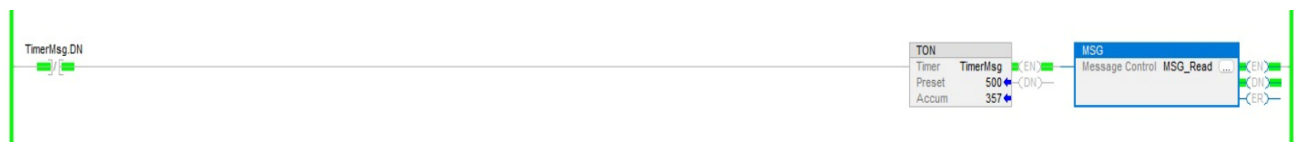


Figura 26. Lógica de Mensajería Periódica.

Otra lógica que se puede implementar es la siguiente, en la cual la mensajería se ejecuta con cada scan del controlador:

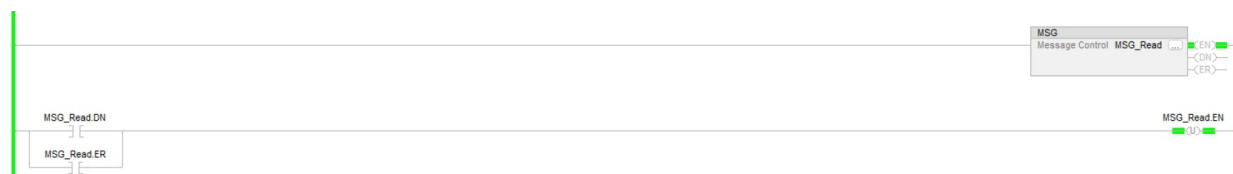


Figura 27. Lógica de Mensajería por Scantime.

Nota: También se podría implementar una lógica de mensajería mediante un contacto NA (Enable_MSG), en el caso de que no sea necesario una conexión continua.

5.1. Lectura de un Solo Parámetro

- Para realizar la lectura de un solo parámetro se debe seguir la siguiente configuración de la instrucción de mensajería. En *Service Type* seleccionar la opción *Get Attribute Single* (automáticamente el Service Code se le asigna el valor de 0x0E). En *Class* colocar el valor de 0x93 (Class ID del Objeto DPI Parameter). En *Instance* colocar el valor de 3 (la Corriente de Salida es el parámetro b003). En *Attribute* colocar el valor de 0x09 (Atributo de Instancia DPI Parameter Value). En *Destination Element* buscar o crear una tag de tipo DINT llamada “Output_Current”. Dar click en Apply, luego OK.

Figura 28. Lectura de un solo parámetro (Output_Current).

- Revisar que la bandera DN de la instrucción se haya activado. En el caso de que la bandera ER se haya activado revisar la sección de troubleshooting de la instrucción MSG.

Figura 29. MSG Troubleshooting.

- Ir a Controller Tags, luego a Monitor Tags para revisar el valor de la variable Output_Current.

▶ Output_Current	102	Decimal	DINT
------------------	-----	---------	------

Figura 30. Parámetro b003.

Nota: La razón por la cual el valor obtenido de la corriente sea 102 **no significa que físicamente se leyeron 102 [A]**, sino que así es el formato interno del Drive para el parámetro b003. A este valor se le debe realizar un escalamiento de 0.01 (con una instrucción DIV por ejemplo), por lo que el valor real de la corriente de salida es de 1.02 [A].

5.1.1. Lectura del Código de Fallo

- Un parámetro de interés para cualquier Sistema HMI es el código de fallo. Mediante lo anteriormente explicado se puede obtener fácilmente este parámetro. Para ello modificar la anterior configuración con el valor 7 en *Instance* (el Código de Fallo 1 es el parámetro b007). Luego, crear una variable “Fault_Code” de tipo DINT y asignarla en *Destination Element*.

Figura 31. Lectura del Código de Fallo 1.

- Ir a Controller Tags, luego a Monitor Tags para revisar el valor de la variable Fault_Code.

► Fault_Code	73
--------------	----

Figura 32. Parámetro b007.

5.1.2. Lectura del Product Name

- Mediante el Objeto de Identidad se pueden obtener las propiedades del Drive. En este ejemplo se leerá el atributo *Product Name*. En *Service Type* seleccionar la opción *Get Attribute Single*. En *Class* colocar el valor de 0x01 (Class ID del Objeto Identidad). En *Instance* colocar el valor de 1 (puede ser también 0, ya que hasta el valor 0x3FFF la instancia se refiere al propio Drive). En *Attribute* colocar el valor de 0x07 (Atributo de Instancia Product Name). En *Destination Element* crear un array de 33 posiciones de tipo SINT llamado “Product_Name_SINTs”. Dar click en Apply, luego OK.

Message Configuration - MSG_Read

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 1 (Hex) Instance: 1 Attribute: 7 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: Product_Name_SINT

New Tag...

☒ Enable
 ☐ Enable Waiting
 ☐ Start
 ☒ Done
 Done Length: 30

☐ Error Code:
 Extended Error Code:
 ☐ Timed Out

Error Path: PowerFlex525

Error Text:

OK Cancel Apply Help

Figura 33. Lectura del Product Name.

Nota: La razón por la cual se crea un array de 33 posiciones de SINT es porque se pretende luego copiar el contenido a una tag de tipo STRING_32. También recordar que cada carácter de un STRING ocupa 1 byte, lo cual es equivalente a 1 SINT.

- Ir a Controller Tags, luego a Monitor Tags para revisar el valor de la variable Product_Name_SINTs.

Product_Name_SINTs	{...}
Product_Name_SINTs[0]	29
Product_Name_SINTs[1]	80
Product_Name_SINTs[2]	111
Product_Name_SINTs[3]	119
Product_Name_SINTs[4]	101
Product_Name_SINTs[5]	114
Product_Name_SINTs[6]	70
Product_Name_SINTs[7]	108
Product_Name_SINTs[8]	101
Product_Name_SINTs[9]	120
Product_Name_SINTs[10]	32
Product_Name_SINTs[11]	53
Product_Name_SINTs[12]	50
Product_Name_SINTs[13]	53
Product_Name_SINTs[14]	32
Product_Name_SINTs[15]	49
Product_Name_SINTs[16]	80
Product_Name_SINTs[17]	32
Product_Name_SINTs[18]	49
Product_Name_SINTs[19]	49
Product_Name_SINTs[20]	48
Product_Name_SINTs[21]	86
Product_Name_SINTs[22]	32
Product_Name_SINTs[23]	32
Product_Name_SINTs[24]	32
Product_Name_SINTs[25]	46
Product_Name_SINTs[26]	53
Product_Name_SINTs[27]	48
Product_Name_SINTs[28]	72
Product_Name_SINTs[29]	80

Figura 34. Contenido de la variable Product_Name_SINTs.

Nota: Para poder entender el contenido recibido de la mensajería explícita se debe tener en cuenta que el Data Type del atributo *Product Name* es un SHORT_STRING. Todo STRING es una estructura de dos datos: LEN de tipo DINT (Longitud de la cadena), DATA de tipo SINT[32] (Caracteres de un STRING_32). Por lo que, lo primero que se obtiene en Product_Name_SINTs[0] es el LEN o longitud del STRING, el cual tendría **29 caracteres**. Luego, los siguientes valores del array serían la DATA o los caracteres en **formato ASCII** (por ejemplo 80 es 'P', 111 es 'o', 119 es 'w', etcétera). Por lo que, para obtener el STRING, se debe realizar una lógica de conversión.

- Una solución sencilla a este problema es el uso conjunto de las instrucciones MOVE y COP. La operación MOVE se realiza no solo para copiar el contenido de Product_Name_SINTs[0] a Product_Name.LEN, sino también convertirlo de SINT a DINT (lo cual no es posible con la instrucción COP). Luego, mediante la instrucción COP se copia el contenido restante de Product_Name_SINTs en Product_Name.DATA.

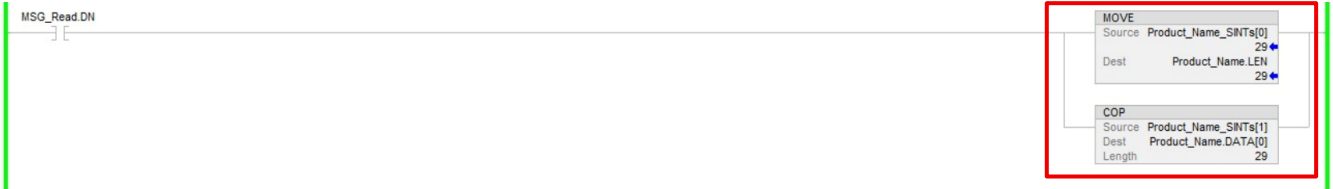


Figura 35. Lógica de Conversión a STRING.

- Ir a Controller Tags, luego a Monitor Tags para revisar el contenido de la variable Product_Name.

Product_Name	'PowerFlex 525 1P 110V .50HP'
Product_Name.LEN	29
Product_Name.DATA	{...}
Product_Name.DATA[0]	'p'
Product_Name.DATA[1]	'o'
Product_Name.DATA[2]	'w'
Product_Name.DATA[3]	'e'
Product_Name.DATA[4]	'r'
Product_Name.DATA[5]	'F'
Product_Name.DATA[6]	'I'
Product_Name.DATA[7]	'e'
Product_Name.DATA[8]	'x'
Product_Name.DATA[9]	' '
Product_Name.DATA[10]	'5'
Product_Name.DATA[11]	'2'
Product_Name.DATA[12]	'5'
Product_Name.DATA[13]	' '
Product_Name.DATA[14]	'1'
Product_Name.DATA[15]	'p'
Product_Name.DATA[16]	' '
Product_Name.DATA[17]	'1'
Product_Name.DATA[18]	'1'
Product_Name.DATA[19]	'0'
Product_Name.DATA[20]	'V'

Figura 36. Contenido de la variable Product_Name.

5.2. Lectura de Varios Parámetros

- Para realizar la lectura de más de un solo parámetro se utilizará el servicio específico de objeto *Get_Attributes_Scattered* con valor 0x32. Se pretenderá en este ejemplo leer 10 parámetros del Drive.
- En *Service Type* seleccionar la opción *Custom*. En *Service Code* colocar el valor de 0x32 (ID del servicio *Get_Attributes_Scattered*). En *Class* colocar el valor de 0x93 (Class ID del Objeto DPI Parameter). En *Instance* colocar el valor de 0 (la instancia se refiere al propio Drive). En *Attribute* colocar el valor de 0x00 (Atributo de Clase Number of Instances). En *Source Element* crear un array de 20 posiciones de tipo INT llamado “Scattered_Read_Request”. En *Destination Element* crear un array de 20 posiciones de tipo INT llamado “Scattered_Read_Response”. En *Source Length* colocar el valor de 40 Bytes (la suma de las dos longitudes). Dar click en Apply, luego OK.

Figura 37. Lectura de varios parámetros.

- Ir a Controller Tags, luego a Monitor Tags para modificar el contenido de la variable *Scattered_Read_Request*.

- Según el formato que dicta el servicio, se debe escribir *en grupos de 2*, primero el número del parámetro a ser leído (por ejemplo b015 es RMP Salida) y luego el valor del Pad que siempre es 0 (para lectura).

Scattered_Read_Request	{...}	{...} Decimal	INT[20]	
Scattered_Read_Request[0]	15	Decimal	INT	RMP Salida
Scattered_Read_Request[1]	0	Decimal	INT	Pad
Scattered_Read_Request[2]	17	Decimal	INT	Potencia Salida
Scattered_Read_Request[3]	0	Decimal	INT	Pad
Scattered_Read_Request[4]	19	Decimal	INT	Tiempo de Marcha
Scattered_Read_Request[5]	0	Decimal	INT	Pad
Scattered_Read_Request[6]	21	Decimal	INT	Contador kWh
Scattered_Read_Request[7]	0	Decimal	INT	Pad
Scattered_Read_Request[8]	22	Decimal	INT	MWh acumulado
Scattered_Read_Request[9]	0	Decimal	INT	Pad
Scattered_Read_Request[10]	24	Decimal	INT	kWh acum guard
Scattered_Read_Request[11]	0	Decimal	INT	Pad
Scattered_Read_Request[12]	25	Decimal	INT	Costo acum guard
Scattered_Read_Request[13]	0	Decimal	INT	Pad
Scattered_Read_Request[14]	26	Decimal	INT	CO2 acum guard
Scattered_Read_Request[15]	0	Decimal	INT	Pad
Scattered_Read_Request[16]	27	Decimal	INT	Temp Variador
Scattered_Read_Request[17]	0	Decimal	INT	Pad
Scattered_Read_Request[18]	28	Decimal	INT	Temp Control
Scattered_Read_Request[19]	0	Decimal	INT	Pad

Figura 38. Contenido de la variable *Scattered_Read_Request*.

Nota: Se recomienda comentar cada elemento del arreglo para evitar errores.

- Verificar que no se haya activado la bandera ER de la instrucción MSG. Ir a Controller Tags, luego a Monitor Tags para visualizar el contenido de la variable *Scattered_Read_Response*.

Scattered_Read_Response	{...}	{...} Decimal	INT[20]	
Scattered_Read_Response[0]	15	Decimal	INT	RMP Salida
Scattered_Read_Response[1]	900	Decimal	INT	Parameter Value
Scattered_Read_Response[2]	17	Decimal	INT	Potencia Salida
Scattered_Read_Response[3]	0	Decimal	INT	Parameter Value
Scattered_Read_Response[4]	19	Decimal	INT	Tiempo de Marcha
Scattered_Read_Response[5]	1	Decimal	INT	Parameter Value
Scattered_Read_Response[6]	21	Decimal	INT	Contador kWh
Scattered_Read_Response[7]	0	Decimal	INT	Parameter Value
Scattered_Read_Response[8]	22	Decimal	INT	MWh acumulado
Scattered_Read_Response[9]	0	Decimal	INT	Parameter Value
Scattered_Read_Response[10]	24	Decimal	INT	kWh acum guard
Scattered_Read_Response[11]	4	Decimal	INT	Parameter Value
Scattered_Read_Response[12]	25	Decimal	INT	Costo acum guard
Scattered_Read_Response[13]	0	Decimal	INT	Parameter Value
Scattered_Read_Response[14]	26	Decimal	INT	CO2 acum guard
Scattered_Read_Response[15]	20	Decimal	INT	Parameter Value
Scattered_Read_Response[16]	27	Decimal	INT	Temp Variador
Scattered_Read_Response[17]	29	Decimal	INT	Parameter Value
Scattered_Read_Response[18]	28	Decimal	INT	Temp Control
Scattered_Read_Response[19]	51	Decimal	INT	Parameter Value

Figura 39. Contenido de la variable *Scattered_Read_Response*.

5.3. Escritura de un Solo Parámetro

- Para realizar la escritura de un solo parámetro se debe seguir la siguiente configuración de la instrucción de mensajería. En *Service Type* seleccionar la opción *Set Attribute Single* (automáticamente el Service Code se le asigna el valor de 0x10). En *Class* colocar el valor de 0x93 (Class ID del Objeto DPI Parameter). En *Instance* colocar el valor de 52 (el Costo Promedio por kWh es el parámetro p052). En *Attribute* colocar el valor de 0x09 (Atributo de Instancia DPI Parameter Value). En *Source Element* crear una tag de tipo INT llamada “Write_MeanCost_kWh”. En *Source Length* colocar el valor de 2 bytes (el tamaño de un dato tipo INT). Dar click en Apply, luego OK.

Figura 40. Escritura de un solo parámetro (Average kWh Cost).

- Ir a Controller Tags, luego a Monitor Tags para modificar el valor de la variable Write_MeanCost_kWh. Por ejemplo, darle el valor de 1.

Figura 41. Valor a escribir al parámetro p052.

- Verificar que no se haya activado la bandera ER de la instrucción MSG.

- Para revisar que efectivamente se ha realizado la escritura se puede buscar el parámetro p052 mediante hardware o, de forma más sencilla, ir a la sección de Parameters del PowerFlex525 en Studio 5000 Logix Designer.

0	52	Average kWh Cost	0.01	1	0.00	0.00	655.35
---	----	------------------	------	---	------	------	--------

Figura 42. Escritura del parámetro p052.

Nota: Nótese que se escribió 1 por mensajería, esto debido a que si se hubiera escrito el valor lógico de 0.01 **el software hubiera mandado un error**. Recordar que la escritura de parámetros *debe coincidir con el formato interno* del parámetro correspondiente.

5.4. Escritura de Varios Parámetros

- Para realizar la escritura de más de un solo parámetro se utilizará el servicio específico de objeto *Set_Attributes_Scattered* con valor 0x34. Se pretenderá en este ejemplo escribir 7 parámetros del Drive.
- En *Service Type* seleccionar la opción *Custom*. En *Service Code* colocar el valor de 0x34 (ID del servicio *Set_Attributes_Scattered*). En *Class* colocar el valor de 0x93 (Class ID del Objeto DPI Parameter). En *Instance* colocar el valor de 0 (la instancia se refiere al propio Drive). En *Attribute* colocar el valor de 0x00 (Atributo de Clase Number of Instances). En *Source Element* crear un array de 20 posiciones de tipo INT llamado “Scattered_Write_Request”. En *Destination Element* crear un array de 20 posiciones de tipo INT llamado “Scattered_Write_Response”. En *Source Length* colocar el valor de 40 Bytes (la suma de las dos longitudes). Dar click en Apply, luego OK.

Figura 43. Escritura de varios parámetros.

- Ir a Controller Tags, luego a Monitor Tags para modificar el contenido de la variable Scattered_Write_Request.
- Según el formato que dicta el servicio, se debe escribir *en grupos de 2*, primero el número del parámetro a ser escrito (por ejemplo p031 es Volt Placa Motor) y luego el valor que se le quiere escribir.

Scattered_Write_Request	{...}	{...} Decimal	INT[20]	
Scattered_Write_Request[0]	31	Decimal	INT	Volt Placa Motor
Scattered_Write_Request[1]	230	Decimal	INT	Parameter Value
Scattered_Write_Request[2]	32	Decimal	INT	Hz Placa Motor
Scattered_Write_Request[3]	50	Decimal	INT	Parameter Value
Scattered_Write_Request[4]	33	Decimal	INT	Intens SC Motor
Scattered_Write_Request[5]	20	Decimal	INT	Parameter Value
Scattered_Write_Request[6]	34	Decimal	INT	Amps Placa Motor
Scattered_Write_Request[7]	16	Decimal	INT	Parameter Value
Scattered_Write_Request[8]	35	Decimal	INT	Polos Placa Motor
Scattered_Write_Request[9]	2	Decimal	INT	Parameter Value
Scattered_Write_Request[10]	36	Decimal	INT	RPM NP Motor
Scattered_Write_Request[11]	1600	Decimal	INT	Parameter Value
Scattered_Write_Request[12]	37	Decimal	INT	Pot NP Motor
Scattered_Write_Request[13]	40	Decimal	INT	Parameter Value

Figura 44. Contenido de la variable Scattered_Write_Request.

Nota: Se recomienda comentar cada elemento del arreglo para evitar errores.

- Verificar que no se haya activado la bandera ER de la instrucción MSG. Ir a Controller Tags, luego a Monitor Tags para visualizar el contenido de la variable Scattered_Write_Response.

Scattered_Write_Response	{...}	{...} Decimal	INT[20]	
Scattered_Write_Response[0]	31	Decimal	INT	Volt Placa Motor
Scattered_Write_Response[1]	0	Decimal	INT	Pad
Scattered_Write_Response[2]	32	Decimal	INT	Hz Placa Motor
Scattered_Write_Response[3]	0	Decimal	INT	Pad
Scattered_Write_Response[4]	33	Decimal	INT	Intens SC Motor
Scattered_Write_Response[5]	0	Decimal	INT	Pad
Scattered_Write_Response[6]	34	Decimal	INT	Amps Placa Motor
Scattered_Write_Response[7]	0	Decimal	INT	Pad
Scattered_Write_Response[8]	35	Decimal	INT	Polos Placa Motor
Scattered_Write_Response[9]	0	Decimal	INT	Pad
Scattered_Write_Response[10]	36	Decimal	INT	RPM NP Motor
Scattered_Write_Response[11]	0	Decimal	INT	Pad
Scattered_Write_Response[12]	37	Decimal	INT	Pot NP Motor
Scattered_Write_Response[13]	0	Decimal	INT	Pad

Figura 45. Contenido de la variable Scattered_Write_Response.

Nota: Si se efectuó correctamente la escritura los Pad tendrán el valor de 0.

- Para revisar que efectivamente se ha realizado la escritura se pueden buscar los parámetros mediante hardware o, de forma más sencilla, ir a la sección de Parameters del PowerFlex525 en Studio 5000 Logix Designer.

0	31	Motor NP Volts	230	V	230	230	20	230	...
0	32	Motor NP Hertz	50	Hz	50	60	15	500	...
0	33	Motor OL Current	2.0	A	20	2.5	0.0	5.0	...
0	34	Motor NP FLA	1.6	A	16	1.7	0.1	5.0	...
0	35	Motor NP Poles	2		2	4	2	40	...
0	36	Motor NP RPM	1600	RPM	1600	1750	0	24000	...
0	37	Motor NP Power	0.40	kW	40	0.40	0.00	655.35	...

Figura 46. Escritura de los parámetros del motor.