EN6001 EIP Communication Configuration Application Note 700237A

The EN6001 Control follows the Common Industrial Protocol (CIPTM) through Ethernet/IP (EIP), providing full data exchange with any standard EIP device. For example, this includes the ability to exchange parameters with a PLC (Setting Data) on a user programmed event as well as the ability to read and write the control's Inputs and Outputs (I/O Data) on a user defined interval. Both of these methods are described below. This Application Note is a supplement to the Communication Specifications for EN6001 Series Controls (P/N 700231), which is referenced as needed below.

1. Configuration Rules

1.1. Ethernet cable:

If the EN6001 control is connected to an EIP device directly, a crossover Ethernet cable must be used. If the EN6001 control is connected to an EIP device through a router or switch, a straight Ethernet cable must be used.

1.2. Ethernet ports:

The EN6001 control uses the following communication ports:

- Port number 44818 and 2222 for EIP
- Port number 502 for Modbus
- Port number 30718 for UDP Control Searching

Port number 44818 and 2222 are the ports the EN6001 uses to exchange data between standard EIP devices. If ENTRON's software, ENLINK6001, is used to set the EN6001 parameters, port 502 and port 30718 will be used.

If a computer and/or a router are used for configuration, make sure the firewall does not block the above Ethernet ports.

1.3. Maximum number of connections:

The EN6001 can accept up to two EIP TCP/IP connections from <u>different</u> EIP devices. If two EIP devices have established connections with the EN6001, the control will not respond to a third connection request from the third device until one of those two EIP devices disconnects.

The EN6001 can accept only one EIP TCP/IP connection from the <u>same</u> EIP device. In other words, the PLC Programmer must ensure only one message is sent to the EN6001

at a time. It is recommended to wait for the DONE bit of a MESSAGE instruction to be TRUE before sending the next MESSAGE.

- 1.4. Note: The following three inputs must be hard wired and cannot be sent via EIP. Refer to the EN6001 User Manual (700230)
 - TLS1 SCR Temperature Limit Switch
 - ES1- Emergency Stop
 - NW1 No Weld
- 1.5. There are three ways to configure communication to the EN6001 as shown in Figure 1. Configuration may be performed in EnLink6001 as shown below, or through the keypad on the control.



1.5.1. Local – Hardwired I/O. When using this method, the I/O Map is set as shown in Figure 2.

Prog	grammable Input (F	PI)	Control Signal S	Source		
	Function	Source				
1	TT1 ~	Local v	FS1	Local ~	Back Step	PI v
2	Edit Lock ~	Local v	FS2	Local ~	2nd Stage	PI v
3	Stepper Reset ~	Local v	PS1	PI v	WCTR Reset	PI
4	Error Reset ~	Local v	Retraction	PI v	Stepper Reset	PI v
5	Back Step ~	Local v	PCTR Reset	PI v	Sch. Select1	PI
6	Escape ~	Local v	Error Reset	PI v	Sch. Select2	PI
			TT1	PI v	Sch. Select4	PI
rog	grammable Output	(PO) Function	Interlock	PI v	Sch. Select8	PI
			Edit Lock	PI v	Sch. Select16	Off
1	ANY Error ~		Escape	PI v	Sch. Select32	Off
2	AVC Error v					
3	Current Error ~					
4	Step End v					



1.5.2. EIP Explicit – Setting Data (Remote 1). When using this method, the I/O Map is set as shown in Figure 3.

Prog	grammable Inpu	t (P)	Control Signal S	ource				
	Function		Source						
1	TT1	~	Remote1 ~	FS1	Local	~	Back Step	PI	~
2	Edit Lock	\sim	Remote1 ~	FS2	Local	~	2nd Stage	PI	~
3	Stepper Reset	\sim	Remote1 ~	PS1	PI	~	WCTR Reset	PI	\sim
4	Error Reset	~	Remote1 ~	Retraction	PI	~	Stepper Reset	PI	~
5	Back Step	~	Remote1 ~	PCTR Reset	PI	~	Sch. Select1	PI	~
6	Escape	\sim	Remote1 ~	Error Reset	PI	~	Sch. Select2	PI	~
				TT1	PI	~	Sch. Select4	PI	\sim
Prog	grammable Outp	out (PO) Function	Interlock	PI	~	Sch. Select8	PI	~
				Edit Lock	PI	~	Sch. Select16	Off	~
1	ANY Error	~		Escape	PI	~	Sch. Select32	Off	~
2	AVC Error	~							
3	Current Error	~							
4	Step End	~							



1.5.3. EIP Implicit – I/O Data (Remote 2). When using this method, the I/O Map is set as shown in Figure 4. The setting of the Programmable Inputs do not matter here.

Function		Source				
TT1	~	Remote1 ~	FS1	Remote2 ~	Back Step	Remote2
Edit Lock	~	Remote1 ~	FS2	Remote2 ~	2nd Stage	Remote2
Stepper Reset	~	Remote1 ~	PS1	Remote2 ~	WCTR Reset	Remote2
Error Reset	~	Remote1 ~	Retraction	Remote2 ~	Stepper Reset	Remote2
Back Step	~	Remote1 ~	PCTR Reset	Remote2 ~	Sch. Select1	Remote2
Escape	~	Remote1 ~	Error Reset	Remote2 ~	Sch. Select2	Remote2
			TT1	Remote2 ~	Sch. Select4	Remote2
		O) Eunction	Interlock	Remote2 ~	Sch. Select8	Remote2
grammable Outp	put (P	of r unction				The second second second
grammable Outp	put (P	o) r uncuon	Edit Lock	Remote2 ~	Sch. Select16	Remote2
grammable Outp	р ит (Р	O) I unclion	Edit Lock Escape	Remote2 ∨ Remote2 ∨	Sch. Select16 Sch. Select32	Remote2 Remote2
ANY Error AVC Error	рит (Р 	of runction	Edit Lock Escape	Remote2 ∨ Remote2 ∨	Sch. Select16 Sch. Select32	Remote2 Remote2
ANY Error AVC Error Current Error	Put (P ~ ~ ~	of runction	Edit Lock Escape	Remote2 v Remote2 v	Sch. Select16 Sch. Select32	Remote2 Remote2

Figure 4 EIP Implicit I/O Map

2. I/O Data (Implicit Messaging)

2.1. Implicit message setting

Implicit messages are automatically sent on a cyclic basis. To exchange Implicit I/O data with the EN6001, the PLC or any Ethernet/IP device need to establish a Class 1 connection. The parameters for setting the connection are shown in Table1:

Origin	Originator->Target (O->T) Connection						
Connection point	170						
Data Size	4 bytes (2 words) with Run/Idle Header						
Connection Rate (RPI)	100—10000 mS						
Transport Type	Point to Point						
Target->Originator (T->O) Connection							
Connection point	120						
Data Size	200 bytes (100 words) without Run/Idle Header						
Connection Rate (RPI)	100—10000 mS						
Transport Type	Point to Point						
Transport Trigger	Cyclic						
	Configuration						
Configuration Instance	1						
Data Size	0						

 Table 1 Parameters for Class 1 Connection

The data structure of Originator->Target (O->T) message is shown in Table 4-11 of Communication Specifications 700231. And the data structure of Target ->Originator (T->O) message is shown in Table 5-2A and Table 5-2B. To clarify, the Originator is the PLC and the Target in the EN6001.

2.2. Example

As an example, an Allen Bradley CompactLogix PLC will be used to exchange Implicit Messages with the EN6001.

- 2.2.1. Add the EN6001 to the PLC project
 - 2.2.1.1. While Offline, right-click on the EtherNetIP gateway in the I/O configuration and select **New Module**, as shown in Figure 5.



Figure 5 Create new module

2.2.1.2. Select Generic Ethernet Module and click Create, as shown in Figure 6.

Logix Designer - EN6001 in File Edit View Search Lo Define Define No Forces No Edits	EN6001_IO.ACD [1769-116ER-8818 ogic Communications Tools elect Module Type Catalog Module Discovery Favo	Window Help		10,00 (16)	Chau Eilano Y
Controller Organizer Controller EN600 Controller EN6000 Controller EN6000 Controller EN6000 Contr	Catalog Number TRUETNET STNE SE ETHERNET-MODULE 2 of 410 Module Types Found Close on Create	Description Generic Ethernet Module Generic Ethernet Module	Vendor Nav-Booky Alen-Bradey	Category Communication Communication	Add to Favorites Close Help

Figure 6 Create Generic Ethernet Module

- 2.2.1.3. In the **Module Properties** window shown in Figure 7, enter the information below:
 - Name for the new Ethernet module. In this example, the module will be named EN. This will create Controller Tags in Studio 5000 for use in the program Tasks.
 - Select the **Comm Format** (data type) as **Data-INT**
 - Enter the **IP address** for the module. The default IP address of the EN6001 is 192.168.0.100.
 - Enter the **Assembly Instance** parameters. For theEN6001 these values should be Input = **120**, Output = **170** and Configuration = **1**.
 - Enter the **Size** of the input and output data corresponding to the data sizes configured for the EN6001, in this case **100** words In and **2** words Out. The Size of Configuration is **0**.
 - Click on **OK** to confirm the module properties and continue. The EN6001 has now been added to the I/O configuration in Studio 5000.

ype: /endor:	ETHERNET-MODU Allen-Bradley	JLE Generic Ether	net Module			
arcni. Jame:	EUCal		Connection Para	meters		
ionio.	EN		_	Assembly	Size:	-
escription		<u>.</u>	Input:	120	100	(16-bit)
	_	-	Output:	170	2	(16-bit)
omm Form	at: Data - INT	•	Configuration:	1	0	(8-bit)
Address /	'Host Name dress: 192 . 168	. 0 . 100				
) Host N	Name:		Status Output;			

Figure 7 Setup Module properties

2.2.1.4. Right click on the EN Ethernet module in the project explorer tree and select **Properties**. In the **Module Properties** window, click on the **Connection** tab. Enter the **Requested Packet Interval** (RPI), as shown in Figure 8. For this example, **100** (ms) is entered. Check the box **Use Unicast Connection over EtherNet/IP**. Click on **OK** to confirm.

_	Connection*	Module Info			-	
Reques	ted Packet Inte	erval (RPI):	100.0 🛫 ms	(1.0 - 3200.0 ms)	J	
Maio	ar Fault On Con	troller If Conne	ction Fails While	in Run Mode		
🔽 Use	Unicast Conne	ction over Eth	erNet/IP			
Modu	le Fault					

Figure 8 Enter Requested Packet Interval

- 2.2.2. Create three User Defined Data types (UDT) as described below. An example of this is shown in Figure 9. Right Click on the User Defined folder and select New Data Type. These data types will be used to map the EN6001 I/O to tags in the PLC. Each data type should have 16 bits. Any unused bit not dedicated to a specific function should be created as "Reserved".
 - UDT EN6001_Dig_In_0 contains the bits included in Control Signal [0] word from Table 4-11 in Communication Specifications 700231.
 - UDT **EN6001_Dig_In_1** contains the bits included in Control Signal [1] word from Table 4-11 in Communication Specifications 700231.
 - UDT EN6001_Dig_Out contains the bits included in Control Output word from Table 5-2A in Communication Specifications 700231. Note: Table 5-2A contains the Control Status of the EN6001. Any parameter included in Table 5-2A may be read by the PLC following the method of this example.

Inizer + 4 X Nam	EN6001_D	ig_In_0			Data Type Size: 4 byte
itroller Tags Desc	cription:			Digital InBit To EN6001 Word 0	
Mer	nbers:				
in Task	Name	Data Type	Description		
MainProgram	FS1	BOOL			
Parameters and Local T	100	0000			
MainRoutine	FSZ	BOOL			
duled	PS1	BOOL			
ps ≡	Retract_Request	BOOL			
d Axes	PCTR Reset	BOOL			
uctions	F OF SEE	0002			
	Error_REset	BOOL			
Dig In 0	TT1	BOOL			
_Dig_In_1	Interlock_Request	BOOL			
Dig_Out	Edit Lock	8001			
	Lun_Look	DOOL			
ined	Escape	BOOL			
	Back_Step	BOOL			
ined	Second Stage	BOOL			
	W/CTP Poret	8001			
	We IN_Neser	BOOL			
-	Stepper_Reset	BOOL			
	Rec 14	2001			

Figure 9 User Defined Data Type

2.2.3. Create the three tags described below with their corresponding data types. An example is shown in Figure 10.

Tag	Data Type
EN6001_Digital_Input_0	EN6001_Dig_In_0
EN6001_Digital_Input_1	EN6001_Dig_In_1
EN6001_Digital_Outputs	EN6001_Dig_Out

Name 🔡 🛆	Alias	Bas	Data Type	Description
+ EN:C			AB:ETHERNET_MODULE:C:0	
+ EN:1			AB:ETHERNET_MODULE_INT_200Bytes:I:0	
+ EN:O			AB:ETHERNET_MODULE_INT_4Bytes:O:0	
- EN6001_Digital_Input_0			EN6001_Dig_in_0	Digital InBit To EN6001 W
EN6001_Digital_Input_0.FS1	1		BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.FS2			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.PS1			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Retract_Requ			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.PCTR_Reset		0 0	BOOL	Digital InBit To EN6001 V
EN6001_Digital_Input_0.Error_REset		0 0	BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.TT1			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Interlock_Req			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Edit_Lock	Ú.		BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Escape			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Back_Step			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Second_Stage			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.WCTR_Reset		0 0	BOOL	Digital InBit To EN6001 V
EN6001_Digital_Input_0.Stepper_Reset			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Res_14			BOOL	Digital InBit To EN6001 W
EN6001_Digital_Input_0.Res_15			BOOL	Digital InBit To EN6001 W
+ EN6001_Digital_Input_1			EN6001_Dig_In_1	Digital InBit To EN6001 W
+ EN6001_Digital_Outputs			EN6001_Dig_Out	Digital Out Bit From EN60

Figure	10	Controll	er	Tags
1 15010	10	controlly		1455

- 2.2.4. Create the three copy (COP) instructions shown in Figure 11. Keep in mind the Outputs of the EN6001 are Inputs to the PLC and vice versa.
 - 2.2.4.1. Rung 0 is used to copy the EN6001 Digital Outputs to the EN6001_Digital_Outputs tag. The EN6001 Digital Outputs are located in the 6th word (Byte offset = 12) of Table 5.2A in Communication Specifications 700231. This instruction provides the ability for the PLC to read the EN6001 Digital Output status.
 - 2.2.4.2. Rungs 1 and 2 copy the EN6001_Digital_Input_0 and EN6001_Digital_Input_1 tags to the Control Signal [0] and Control Signal [1] registers in the E6001 shown in Table 4-11 of Communication Specifications 700231. These instructions allow the PLC to write to the EN6001 Digital Inputs.



Figure 11 Copy Instructions

2.2.5. Refer to Figure 12 to see how some of the tags created above are used to control the EN6001. The Cycle Start tag is not part of the EN6001. It is shown to represent the user's logic.

Cycle_start EN6001_Digital_Outputs.EDS EN6001_Digital_Outputs.Ctr_Not_Ready EN6001_Digital_Outputs.Any_Error EN6001_Digital_nput_ 3 Digital InBt To EN6001 Word 0		120000-02002	Digital Out Bit From EN6001	Digital Out Bit From EN6001	Digital Out Bit From EN6001	Digital InBit To EN6001 Word 0
ENBOIT Word 0	3			/		
EN6001_Digital_Input_0.FS1		EN6001 Word 0 EN6001_Digital_Input_0.FS1				



3. Setting Data (Explicit Messaging)

Explicit messages are sent based on a programmed logic event (i.e. pressing a button on an HMI). To send an Explicit Message to the EN6001, the EIP device needs to have four parameters set. These include Service Code, Class Code, Instance Code and Attribute Code. Depending on the type of message, some may need additional settings with the above four parameters.

3.1. Example

In this example, the PLC will be used to change the Weld Schedule in the EN6001. It is a continuation of the example in Section 2.

- 3.1.1. Refer to Section 4.2.2 of the Communication Specifications 700231 and note the following parameters:
 - Service Code = 0x10 (Set Data)
 - Class Code = 0x98 (General Control Data)
 - Instance Code = 1
 - Attribute Code = 0x68 (Use Schedule Data)
 - Data Structure = According to Table 5-8, a 16-word (32-byte) data block will be sent with the request. The first word of the data should be set to the weld schedule number being sent.
- 3.1.2. Create the following tags
 - Weld_Sch_Send with a data type of INT[16]. This tag is used as the data sent to the EN6001 Use Schedule Data (32 bytes) of Table 5-8. The first 2 bytes are contained in Weld_Sch_Send[0].
 - Weld_Sch_New with a data type of INT. In this example, this represents the new weld schedule selected on an HMI.
 - Weld_Sch_Old with a data type of INT. This is the last weld schedule selected before changing it on the HMI.
 - Weld_Sch_Msg with a data type of MESSAGE. This tag will be used in the Message Instruction in the routine.
- 3.1.3. Create the rung shown in Figure 13. The instructions are described below:
 - NEQ this instruction compares the new weld schedule with the old weld schedule. If they are different (i.e. changed on the HMI) the rung will execute.
 - MOV this instruction moves the new weld schedule into the word sent in the Message instruction.

- MOV this instruction moves the new weld schedule into the old weld schedule. The old weld schedule will then be compared to the new weld schedule the next program scan.
- MSG this is the explicit message instruction used to send the weld schedule to the EN6001. The instruction is found in the Input/Output Element Group as shown in Figure 14. Clicking the box on the right of the instruction will open the Message Configuration box shown in Figures 15 and 16. Enter the values shown.



Figure 13 Message Rung

82888999999999999999999999999999999999	ETHP-1/392 168 0 200"
	13.80 · 🚽
Forces 3 to the	
Ens 2 Erop Scope 4 H bol b rea w sw mr	2
B + + Favore & Alerna & BE & Tanene & Inputto A	pare & Compute. & MoveLo. & Field.

Figure 14 Message Instruction

Message	Type: CIP Ge	neric	•]	
Service Type:	Set Attribute Single	•	Source Element:	Weld_Sch_Send[0]
Code: Instance:	10 (Hex) Clas	s: 98 (Hex) e: 68 (Hex)	Destination Élement:	New Tag
) Enable	O Enable Waiting	O Start	O Done D	one Length: 0

Figure 15 Configuration Tab

Path: EN EN	Browse	Move Source Weld_Sch_t
Communication Method CIP DH+ Channet: CIP With Cip Source Link: 0 Destin Destin	eation Link:	Dont Mald Cab Ca
Connected Cache Connections		11B EN6001_App_Note_Mike Discrete_IO lodules EN6001_App_Note_Mike

Figure 16 Communication Tab

3.1.3.1. This program can simulate changing the weld schedule on an HMI by going on line and changing the value of the Weld_Sch_New tag in the Monitor tab of the Controller tags as shown in Figure 17.



Figure 17 Controller Tags