Communicating between 2500 Series® Processors and PowerFlex 525 Drives using Ethernet/IP and 2500P-ACP1

The 2500P-ACP1 Application Coprocessor supports Ethernet/IP communications with up to 40 Ethernet/IP devices via I/O Scanner, I/O Adapter, Explicit Message Adapter, and Tag Client interfaces. This Application Note shows how to configure the ACP1 for communications with a Rockwell PowerFlex 525 drive using Workbench.

IMPORTANT NOTE: Configuring Ethernet/IP communications requires 2500P-ACP1 Firmware V3.03 or above, and Workbench V1.3 or above.
Step 1: Open a Project.

File \ Add New Project

Specify Folder and File name then click Next
Step 2: Specify Target’s (ACP1 module) IP address. **Language** specifies the start-up mode and can be changed later. **T5 Runtime** is the protocol native to Workbench and the ACP1 module (and the Zenon HMI software as well). Port# 1100 is the defined port for interface between Workbench and the ACP1 module. Then click **Next**.
Step 3: The ACP1 is the Scanner and the PowerFlex drive is the Adapter, so select **Ethernet/IP I/O Scanner (client)**. We are not using **Binding over Ethernet** but this can remain checked with the default Port# of 9000. (This is used for communications between ACP1 and similar devices using the Data Exchange protocol.) Then click **Finish**.
Step 4: If you are connected over the network, choosing **Auto-Detect** will connect to the specified IP address and return the **Firmware Version** of the ACP1 module. Choose 2500P-ACP1 as the **CTI Product Type** and then click **OK**.
Step 5: Double-click on **Fieldbus Configurations**. Because we already specified the Ethernet/IP I/O Scanner (client) in Step 3, this driver automatically appears in the configuration window.
Step 6: Click on the **Insert Master/Port** symbol, then type in the **Address** of the PowerFlex drive in the Server pop-up box. Add optional **Description**. The **Configuration Instance** is defined by Rockwell as “6” (ref. “PowerFlex 525 Embedded EtherNet/IP Adapter” User Manual, page 56). Then click **OK**.
Step 7: Expand the **Server** and double-click the **Target to Originator (Input)**. In the **IO/Object** pop-up box, change the **Instance** to “1” (refer to table in previous Step) and the **Size** (in bytes) to “16”. Referring to the table below (page 62), we are using the Generic Profile for the Input which has up to 8 Integers. Change the **Priority** to “High” and leave the **32 bit idle header** unchecked. Then click **OK**.
Step 8: Double-click the **Originator to Target** for the Output definition. The 32 bit header is assumed here and therefore the **32 bit idle header** box is checked. The **Instance** and **Size** are similarly derived from the table referenced in the previous steps. Once again, click **OK**.
Step 9: There are three system variables we want to add; it does not matter if they are listed under **Target to Originator** or **Originator to Target**. Select **Insert Variable** and then the **Diagnostic/Control** button. Select one of the variables and type a name in the **Symbol** box. No spaces or punctuation allowed. Then select **OK**.
Step 10: A pop-up box appears with options to declare this new variable. Select INT in the **Type** drop down menu and leave the **Where** specified as **Global**. Then select **Yes**.
Step 11: Repeat the process for the other two system variables. These definitions will now appear in the **Global variables** window. (UCMM stands for Unconnected Message Manager)
Step 12: Now we can add the other variables for the PowerFlex drive. Referring to the chart in Step 7, add Logic Command (no spaces) as shown below. The **Originator to Target** section is highlighted so this variable will be added as an Output. It is the first variable in the table so its **Offset** is zero. Change **Format** to **16 bit – signed** then click **OK**.
Step 13: Since this is a new variable, a pop-up appears. Change the **Type** to **INT** then click **Yes**.
Step 14: Repeat Steps 12 & 13 for the other Output variables from the table in Step 7 with their respective Offset (in bytes):

- Reference 2
- OUTDatalink1 4
- OUTDatalink2 6
- OUTDatalink3 8
- OUTDatalink4 10

Highlight the **Target to Originator** to add the Input variables. We are using the Generic Profile (third column) so must take account of the Padwords in the Offset:

- LogicStatus 4
- Feedback 6
- INDatalink1 8
- INDatalink2 10
- INDatalink3 12
- INDatalink4 14

The **Server** and **Global variables** should appear as in the picture below when finished.
Step 15: Compile the program (Build Startup Project) ... but have a Compile error due to incorrect Size in the Output definition. Change this to “12”. Also notice that Priority set to Low so change to “High”. Recompile with no errors.
Step 16: **Download** the project to the ACP1.
Step 17: Go **On Line** with the PowerFlex drive. Notice **Server OK** and **I/O connection OK** variables are both “1” and **Last UCMM error** is “0”.

*Image showing a software interface with a button labeled “On Line” and variables such as Server OK, I/O connection OK, and Last UCMM error.*
Type the IP address of the ACP1 module into a browser’s URL to connect to the webserver. Select the Ethernet/IP CIP Statistics to view the connection status.

```
| Last Counter Clear: Thu Jan 01, 1970 22:20:27.056 |

Client Connection Status

<table>
<thead>
<tr>
<th>Server Address</th>
<th>c1</th>
<th>t2</th>
<th>Socket</th>
<th>Port</th>
<th>Process</th>
<th>UOM</th>
<th>IO Exch</th>
<th>Ready</th>
<th>IO</th>
<th>Last UOM Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.018.032.160</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>58924</td>
<td></td>
<td>IO</td>
<td></td>
<td></td>
<td></td>
<td>Connected</td>
</tr>
</tbody>
</table>

UDP (port 44818)  UDP IO (Port 2222)

Packets received: 0 2414
Packets transmitted: 0 2413

TCP Servers  TCP Clients

<table>
<thead>
<tr>
<th>Packets received</th>
<th>Packets transmitted</th>
<th>Closes on failed receive</th>
<th>Closes on failed send</th>
<th>Connections Active</th>
<th>Successful Connections</th>
<th>Failed Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

IO Data Receipt Statistics

<table>
<thead>
<tr>
<th>Lost record count</th>
<th>Max Records/Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Step 18: The last task is to associate the variables defined in ACP1 to memory locations in the PLC. Click the **On Line** button to take the ACP1 off line, then in the IO Drivers window, click **Insert Configuration**, choose the **CTI 2500 Data Cache** option and click **OK**.
Step 19: Highlight the **CTI 2500 Data Cache** entry, then **Insert Master/Port**. In the pop-up window, enter the IP address of the Host PLC. Then click **OK**.
Step 20: Click on the **CTI 2500 IP = “xxx.xxx.xxx.xxx”** entry, then **Insert Slave/Data Block**. Choices for **Memory Type** include V-mem (INT, REAL, DINT), K-mem, X/Y/C, WX/WY, STW, TCP, TCC. **Starting Address** is V700 in this example. **Cache Direction** is either Read From PLC, Write to PLC, or Read and Write. Then click **OK**.
Step 21: If the **Child Properties** pane is not visible, select **View** from the toolbar, then the **Grid** option. Highlight the defined variables (click on first variable, hold down Shift, click on last variable) then drag/drop them into the Child Properties pane.
Step 22: Click on each Symbol (variable), type in the correct Offset, then hit Enter. Offsets increase by “1” since these variables are defined as INT.
Step 23: As before, Build Startup Project to compile, then Stop and download the new version, and finally On Line.
Open up WorkShop, connect to PLC, build a Data Window, and observe the values from the PowerFlex drive displayed. Tags can be added as desired.