Technical Note

Converting Data from Sure Cross Wireless to Allen-Bradley PLCs

This is an Add on Instruction (AOI) for the Logix Designer software package from Rockwell Automation. This AOI handles the data conversion from a Banner Wireless radio system to an Allen-Bradley PLC.

Data conversion is necessary because Allen-Bradley assumes all data is in a signed (+/-) format. Most of the data in a Banner wireless system is unsigned (no negative numbers). This means when a 16-bit value greater than 32767 is sent to the PLC, it becomes a negative number. A similar situation occurs when a value is sent from the PLC to the Banner wireless system. The AOIs described here allow for the automatic conversion of the data to ensure that each device gets the data in the format it prefers.

The rest of the document describes how to use the AOI and what it actually does.

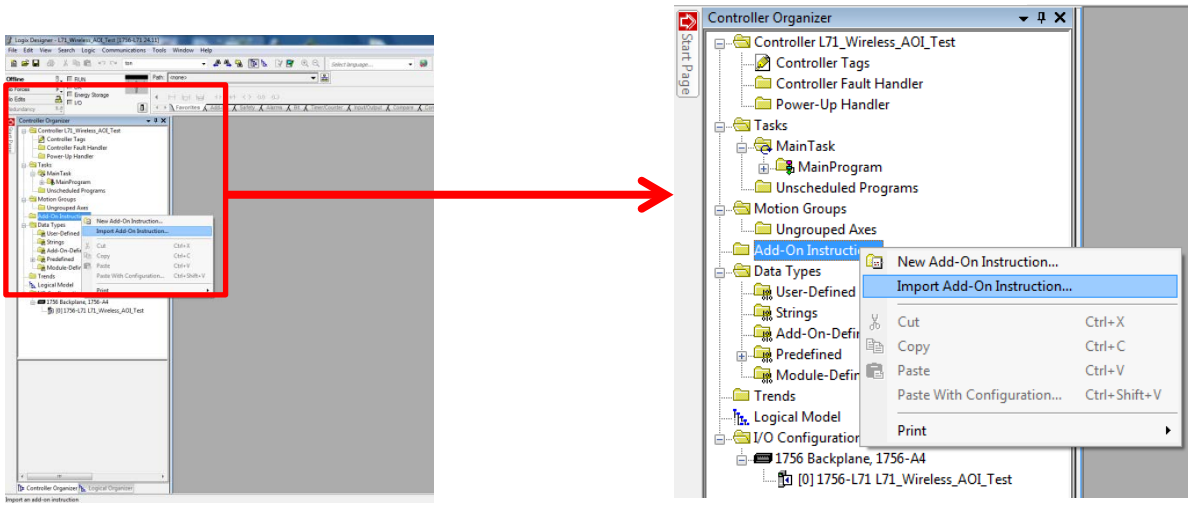
Components

Input AOI: Banner_Wireless_Input_Conversion
Input UDT: Banner_DX_Input
Output AOI: Banner_Wireless_Output_Conversion
Output UDT: Banner_DX_Output

Installation Process

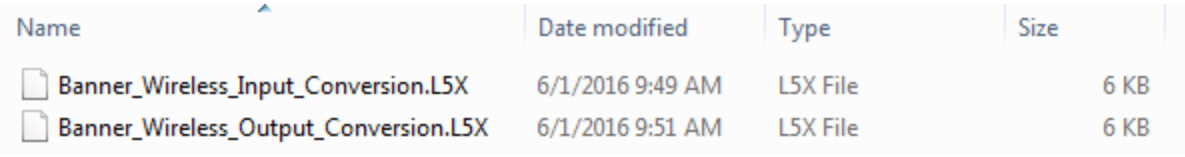
This section describes how to install the Input and Output AOIs into Logix Designer software.

1. Open up a project.
2. Right-click on the Add-On Instruction folder in the Controller Organizer window.
3. Select the Import Add-On Instruction option.

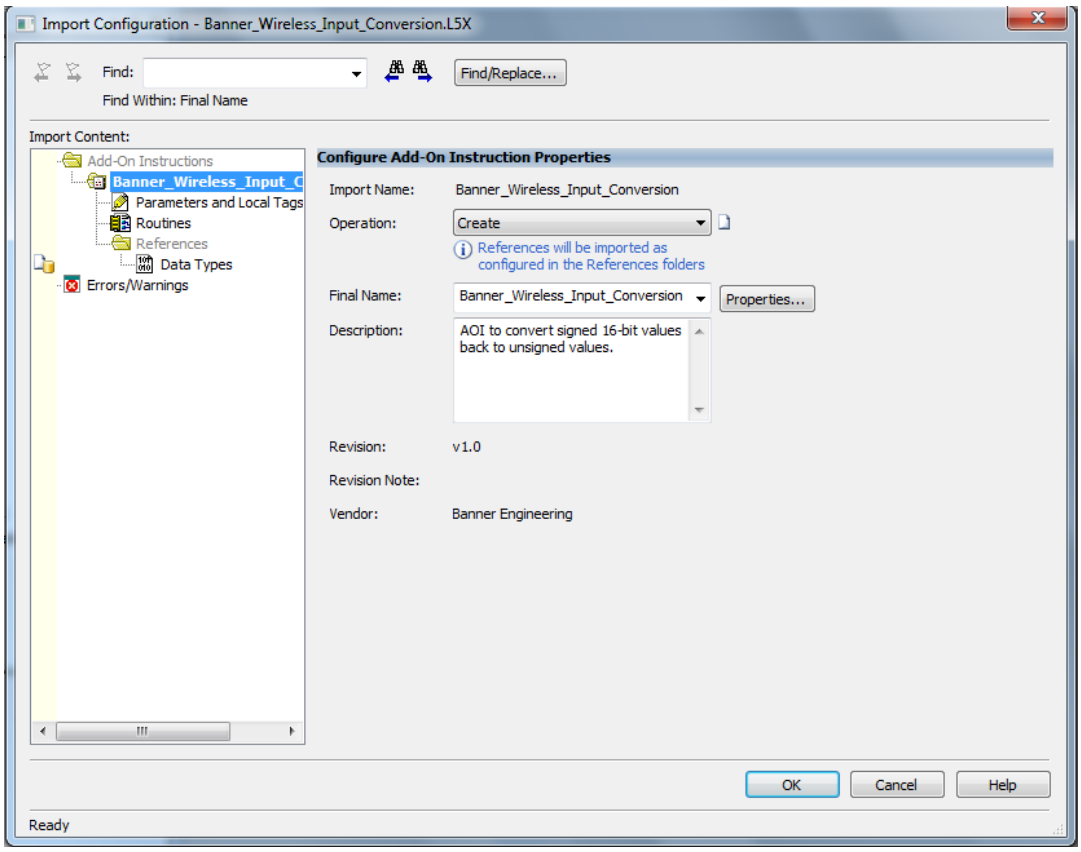


4. A standard windows selection box appears.

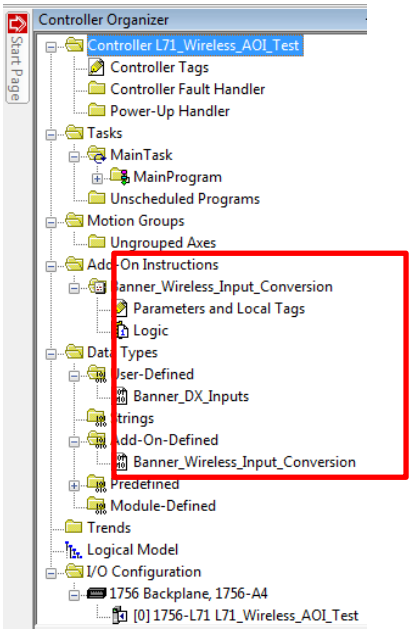
5. Navigate to the correct file location. Two L5X files should be present. One is for the Input and the other is the Output AOI.
6. Select the Banner_Wireless_Input_Conversion.L5X file, then click the Ok button. This is for the Input AOI.



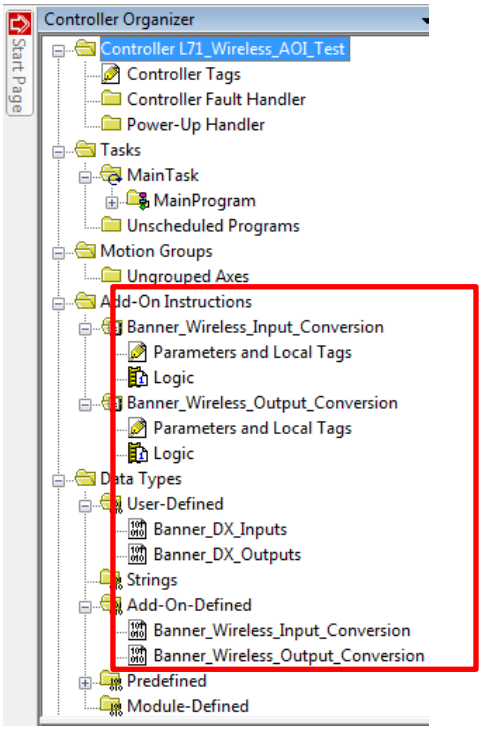
7. The Import Configuration window pops up. The default selection creates all the necessary items for the AOI. In this case a UDT (User Defined Tag) is also created along with the AOI. Press the OK button to complete the import process.



8. The following items should appear in the associated areas.



9. Repeat steps 2 to through 7 for the Output AOI.
10. The Controller Organizer should look similar to the following image after the import.

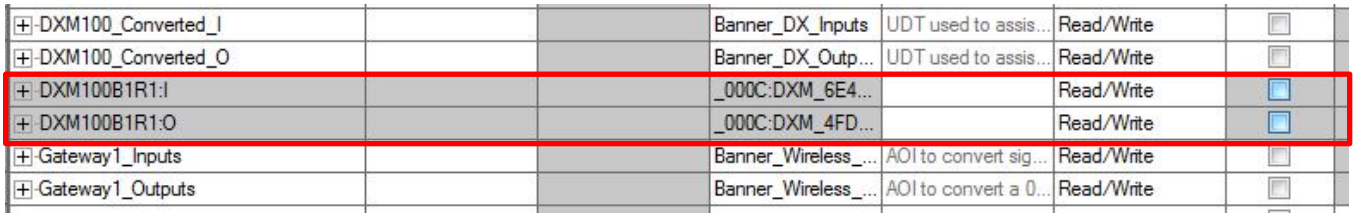


Input AOI – How to Use

The AOI is configured to take in the data array from the Banner wireless device. The user specifies which data points in the array they want to convert. The specified data points are converted from a 16-bit signed value ranging from -32768 to +32767 to a 32-bit value

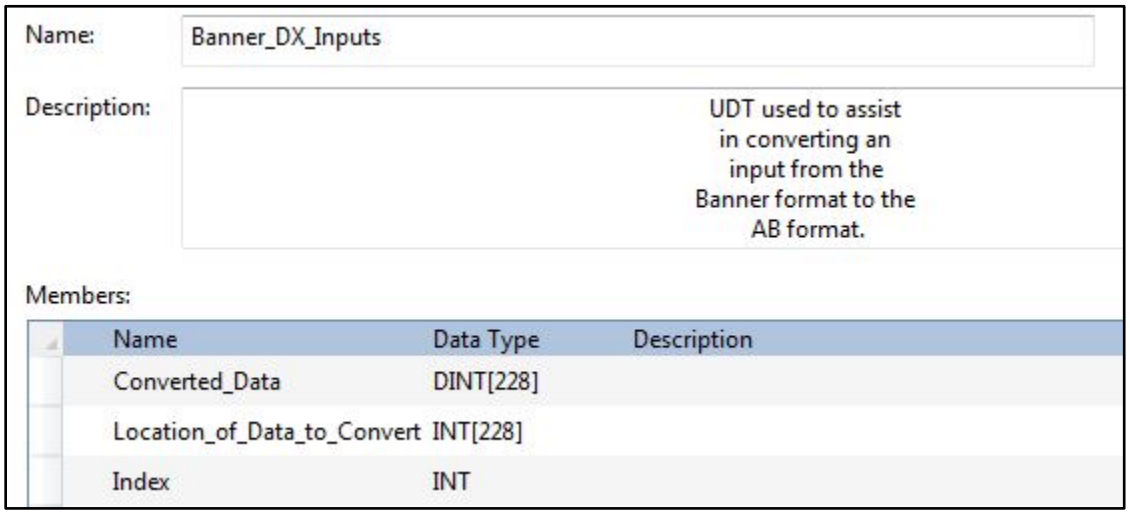
ranging from 0 to 65535. This conversion fixes the signed versus unsigned formatting difference between the Allen Bradley PLC and Banner wireless unit. Follow the steps below to use the Input AOI.

1. Create an Ethernet connection to a Banner Wireless device. In this example I have a connection to a DXM unit. I labeled the connection DXM100B1R1 in the PLC. If you look in the controller tags you should see an input and output data array associated to DXM100B1R1.



⊞ DXM100_Converted_I			Banner_DX_Inputs	UDT used to assis...	Read/Write	
⊞ DXM100_Converted_O			Banner_DX_Outp...	UDT used to assis...	Read/Write	
⊞ DXM100B1R1:I			_000C:DXM_6E4...		Read/Write	
⊞ DXM100B1R1:O			_000C:DXM_4FD...		Read/Write	
⊞ Gateway1_Inputs			Banner_Wireless_...	AOI to convert sig...	Read/Write	
⊞ Gateway1_Outputs			Banner_Wireless_...	AOI to convert a 0...	Read/Write	

2. The DXM100B1R1:I will be used by the Banner_Wireless_Input_Conversion AOI. This is the data we are going to convert.
3. The Banner_DX_Input UDT should have been installed automatically when the AOI was installed. Create a tag using this UDT. In the above example I used DXM100_Converted_I.
4. The Banner_DX_Input UDT is very important to the overall process. The converted values are stored as part of this UDT and the UDT has an array that controls which values coming from the Banner Wireless device need to be converted. The AOI only converts the values requested, allowing the AOI to be very efficient.
5. Open the Banner_DX_Inputs UDT.



Name: Banner_DX_Inputs

Description: UDT used to assist in converting an input from the Banner format to the AB format.

Members:

Name	Data Type	Description
Converted_Data	DINT[228]	
Location_of_Data_to_Convert	INT[228]	
Index	INT	

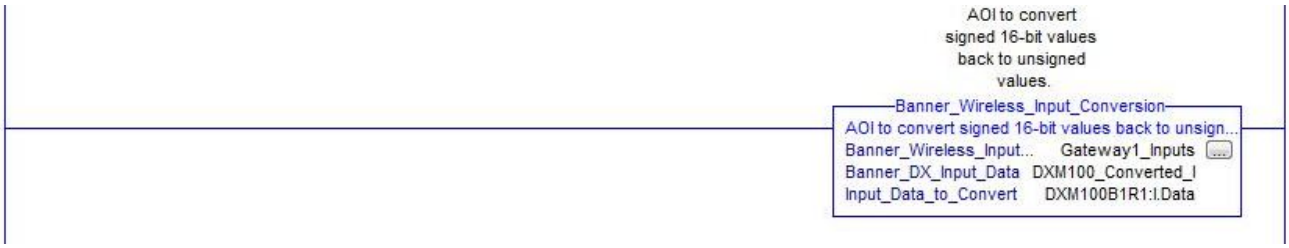
   a. **Converted_Data** is the array that the converted values are stored in.
   b. **Location_of_Data_to_Convert** is the array that stores the location of the values to convert. When a value greater than 227 is entered the conversion process stops. This will make more sense later when the AOI logic is discussed.
   c. **Index** is the variable used to control the current index location inside of the Location_of_Data_to_Convert array.

6. The **Location_of_Data_to_Convert** array inside the UDT needs to be manually configured. Below is a screenshot taken after values have been manually entered into the array. Only a small section of the array is shown.

Scope: Test Show: All Tags Enter Name Filter...

Name	Value	Force Mask	Style	Data Type	Description
AE_100	{...}	{...}	Decimal	INT[16]	
DXM100_Converted_I	{...}	{...}		Banner_DX_Inputs	UDT used to assis...
DXM100_Converted_I.Converted_D...	{...}	{...}	Decimal	DINT[228]	UDT used to assis...
DXM100_Converted_I.Location_of_...	{...}	{...}	Decimal	INT[228]	UDT used to assis...
DXM100_Converted_I.Location_o...	7		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	10		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	0		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	1		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	300		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	0		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	0		Decimal	INT	UDT used to assis...
DXM100_Converted_I.Location_o...	0		Decimal	INT	UDT used to assis...

7. The values 7, 10, 0, 1, and 300 have been entered into the array. Each value 227 or below is an array location in the Input_Data_to_Convert array that should be converted. In this example array location 7 would be converted first. Then 10, 0, and 1 would be converted. 300 signifies that no more locations should be converted. Each number signifies a node register location.
8. Next add an AOI to your ladder logic program. Optionally create logic to control when the AOI should be activated. In the example below I run the AOI continuously.



9. The AOI has two links that need to be connected to controller tags.
    a. Banner_DX_Input_Data is linked to DXM100_Converted_I
    b. Input_Data_to_Convert is linked to DXM100B1R1:I:Data

Output AOI – How to Use

The output AOI is configured to take a DINT (32-bit signed integer) that has a value from 0 to 65535 and convert it into an INT (16-bit signed integer). The value is converted into the format the Banner wireless device understands. Follow the steps below to use the Output AOI.

1. The same Ethernet connection used by the Input AOI will be used by the Output AOI. See "Input AOI – How to Use" step 1 for more information. Again a DXM controller is used in the example. The DXM is labeled as DXM100B1R1. We will be focusing on the Output Instance in this case.

⊞ DXM100_Converted_I			Banner_DX_Inputs	UDT used to assis...	Read/Write	
⊞ DXM100_Converted_O			Banner_DX_Outp...	UDT used to assis...	Read/Write	
⊞ DXM100B1R1:I			_000C:DXM_6E4...		Read/Write	
⊞ DXM100B1R1:O			_000C:DXM_4FD...		Read/Write	
⊞ Gateway1_Inputs			Banner_Wireless_...	AOI to convert sig...	Read/Write	
⊞ Gateway1_Outputs			Banner_Wireless_...	AOI to convert a 0...	Read/Write	

2. When the AOI was installed the UDT was also loaded. Create a tag using this UDT. The example shows the tag listed as DXM100_Converted_O. This tag will be used during the conversion process.
3. The Banner_DX_Input UDT is very important the overall process. It has one array with the values to convert and another array that controlling which values in the PLC should be converted before sending them to the Banner wireless device.
4. Open the Banner_DX_Outputs UDT.

Name: Banner_DX_Outputs

Description: UDT used to assist in converting an output to the Banner format.

Members:

Name	Data Type	Description
Data_to_Convert	DINT[228]	
Location_to_Store_Converted_Data	INT[228]	
Index	INT	

   a. **Data_to_Convert**: Data to convert from DINT to INT format. This is necessary before the data is sent to the Banner wireless device.
   b. **Location_to_Store_Converted_Data**: This is an array that states what elements in the Data_to_Convert array need to be converted. Only the necessary array elements are converted.
   c. **Index**: Is the variable used to control the current index location inside of the Location_to_Store_Converted_Data array.

5. Similar to the UDT for inputs the UDT for outputs needs to have **Location_to_Store_Converted_Data** array manually configured. This configuration tells the AOI which locations to pull data from and where to store the converted data.
6. Below is a screenshot taken after values have been manually entered into the array. Here the array shows that 0, 3, and 1 array locations needs to be converted. Similar to the Input UDT when a value greater than 227 is encountered it signifies an end to the conversion process. Each number represents a node register location.

Name	Value	Force Mask	Style	Data Type	Description
⊞ AE_100	{...}	{...}	Decimal	INT[16]	
⊞ DXM100_Converted_I	{...}	{...}		Banner_DX_Inputs	UDT used to assis...
⊟ DXM100_Converted_O	{...}	{...}		Banner_DX_Outputs	UDT used to assis...
⊞ DXM100_Converted_O.Data_to_Co...	{...}	{...}	Decimal	DINT[228]	UDT used to assis...
⊟ DXM100_Converted_O.Location_to_...	{...}	{...}	Decimal	INT[228]	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	0		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	3		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	1		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	300		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	0		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	0		Decimal	INT	UDT used to assis...
⊞ DXM100_Converted_O.Location_...	0		Decimal	INT	UDT used to assis...

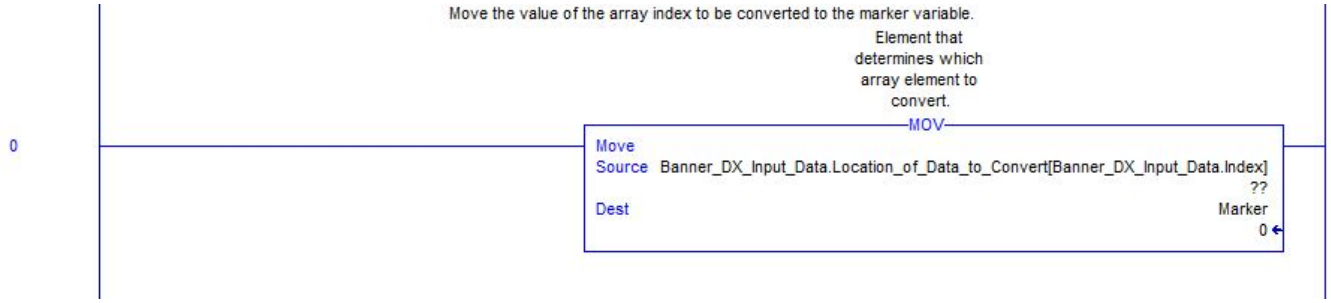
7. Next add the Output AOI to your ladder logic program. Optionally you may create logic to control when the AOI should be activated. In the example below I run the AOI continuously.

8. The AOI has two links that need to be connected to controller tags:
   a. Banner_DX_Output_Data is linked to DXM100_Converted_O
   b. Output_Data is linked to DXM100B1R1:O.Data

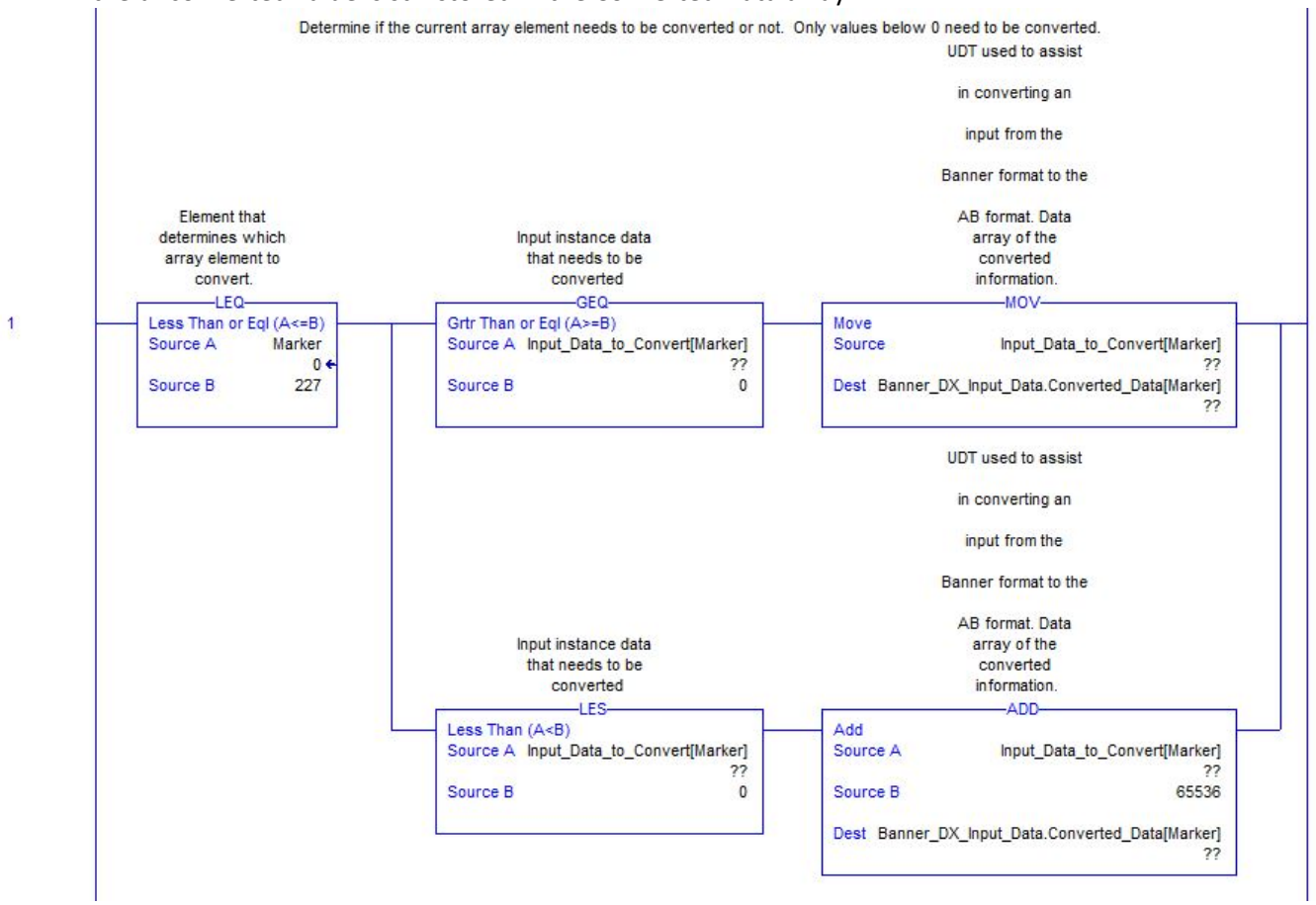
Appendix A

This section will go over the AOI one rung at a time.

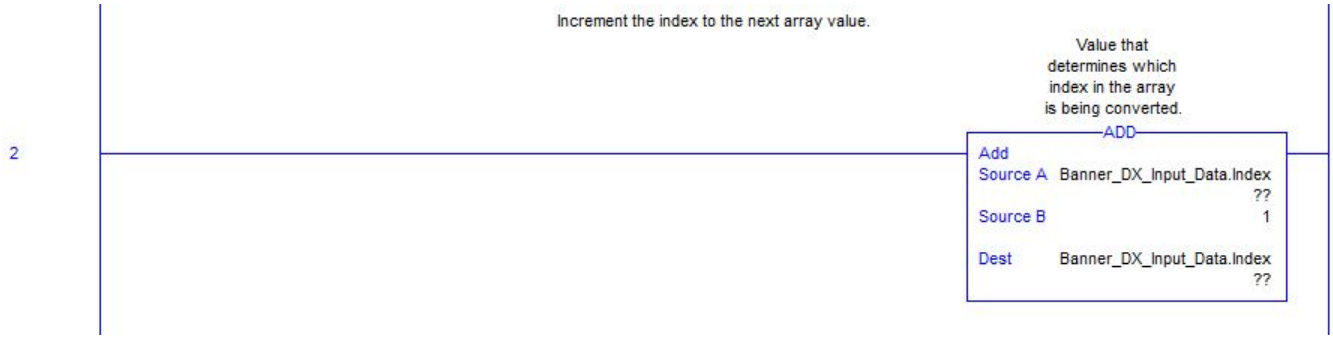
1. Rung 0 pulls the current location value out of the Location_of_Data_to_Convert array. This value is stored in the local variable Marker.



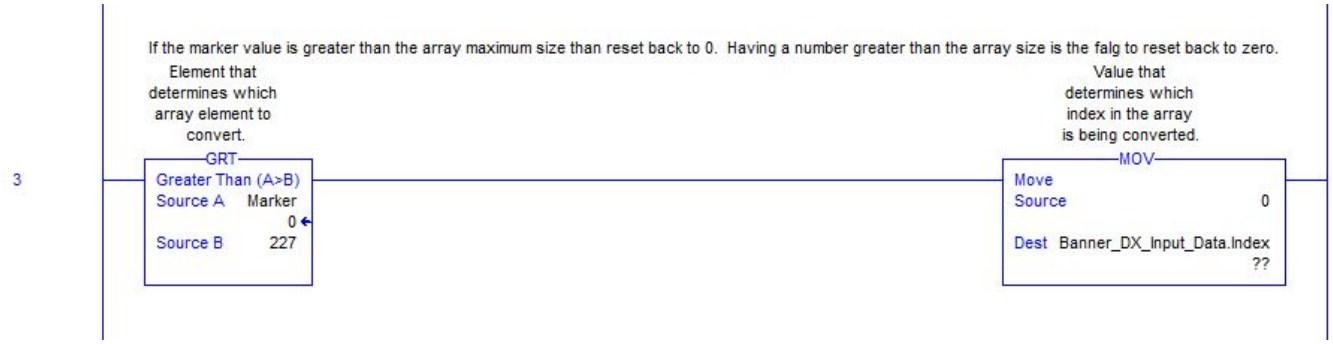
2. Rung 1 starts off by ensuring that the Marker variable is less than or equal to 227. Any number greater than 227 indicates the numbers that need to be converted have been converted. This rung should not run if that is the case.
3. The rest of the rung determines if the value needs to be converted. If it does, then the value is converted and stored in the Converted Data array. If the value does not need to be converted, the unconverted value is still stored in the Converted Data array.



4. Increment the Index parameter so that the next time the AOI runs the next value is converted.



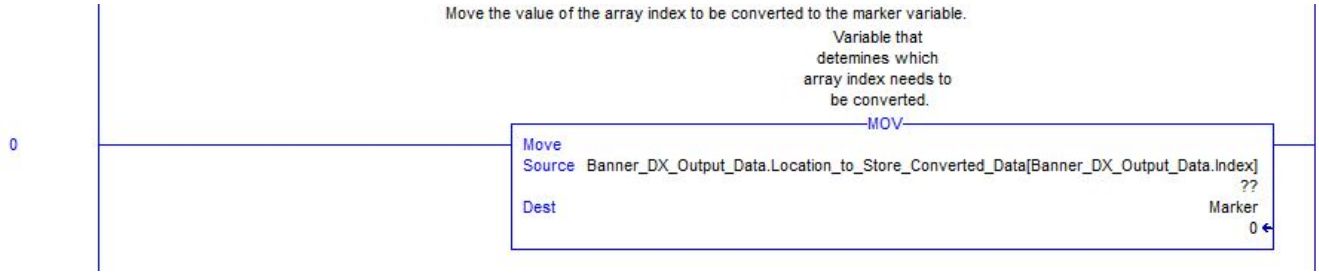
5. Rung 3 checks to see if the next array index is greater than 227. If it is, the data has been converted.
6. Reset the index value back to zero to restart the process.



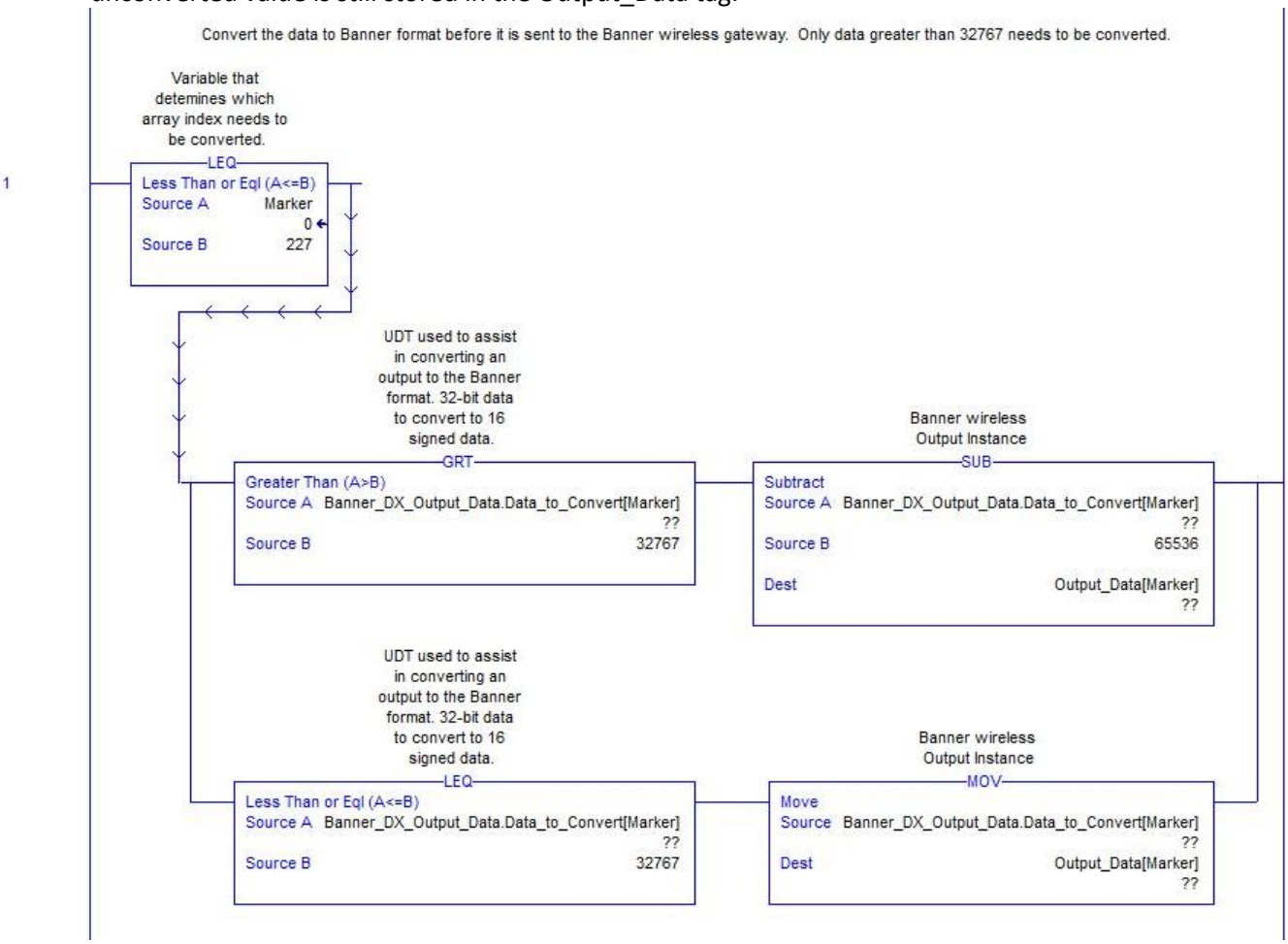
Appendix B

This section will go over the AOI one rung at a time.

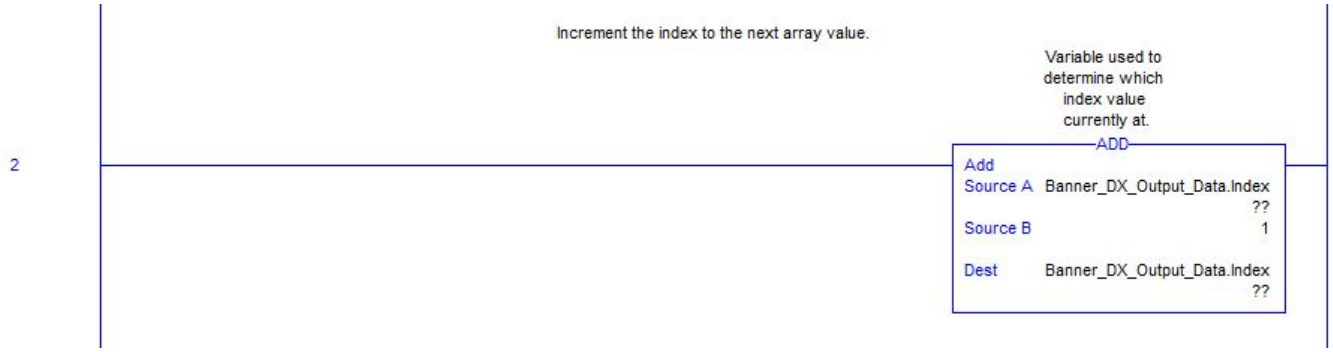
1. Rung 0 pulls the current location value out of the Location_to_Store_Convereted_Data array. This value is stored in the local tag Marker.



2. Rung 1 starts off by ensuring the Marker variable is less than or equal to 227. Any number greater than 227 means indicates the numbers that need to be converted have been converted. This rung should not run if that is the case.
3. The rest of the rung determines if the value needs to be converted. If it does, the value is converted and stored in the Output_Data tag. If the value does not need to be converted, the unconverted value is still stored in the Output_Data tag.



4. Increment the index parameter so that the next time the AOI runs the next value is converted.



5. Rung 3 checks to see if the next array index is greater than 227. If it is, the data has been converted.
6. Reset the index value back to zero to restart the process.