iVu and VE Functions AOI Guide
4/30/2021

This document covers the installation and use of an Add-On Instruction (AOI) for the Logix Designer software package from Rockwell Automation. This AOI handles the basic functions of the iVu or VE camera: trigger, product change, and remote teach. The AOI has six User-Defined Tag data types, and provides labeled data tags to make PLC programming easier. For more information see the Industrial Ethernet Overview chapter of the iVu or VE product manual.

Components

Banner_iVu_VE_Functions.L5X

UDT Packaged with the AOI

Banner_iVu_VE_Control
Banner_iVu_VE_Data
Banner_iVu_VE_Inputs
Banner_iVu_VE_Input_Bits_ACK
Banner_iVu_VE_Outputs
Banner_iVu_VE_Output_Status

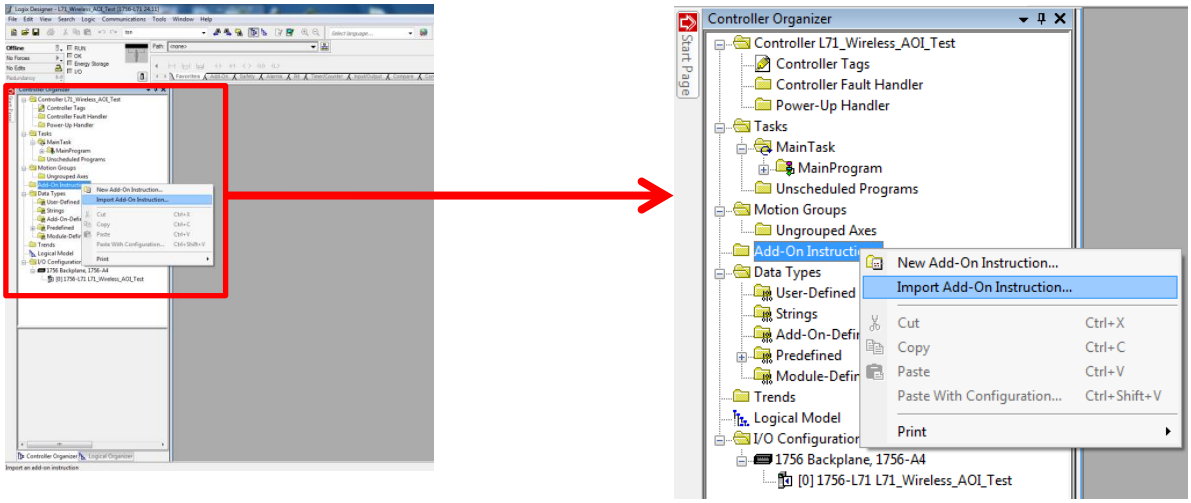
Contents

1. Installation Process

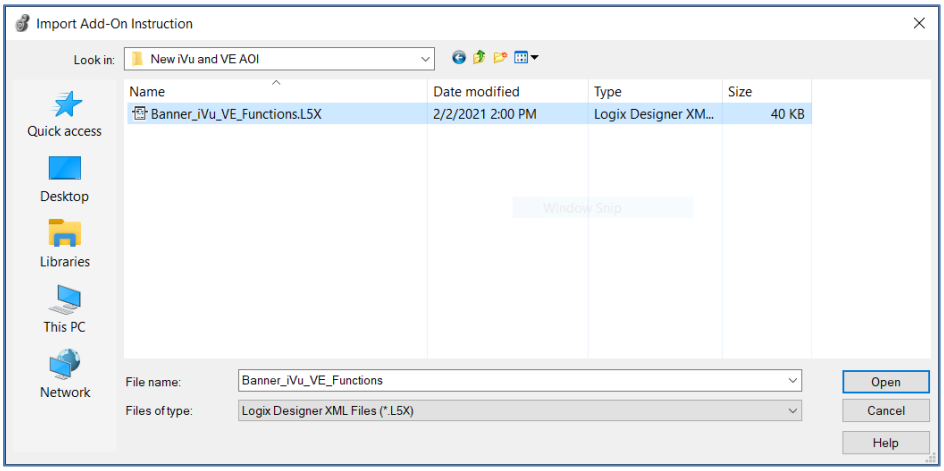
This section describes how to install the AOI in Logix Designer software.

1. Open up a project.

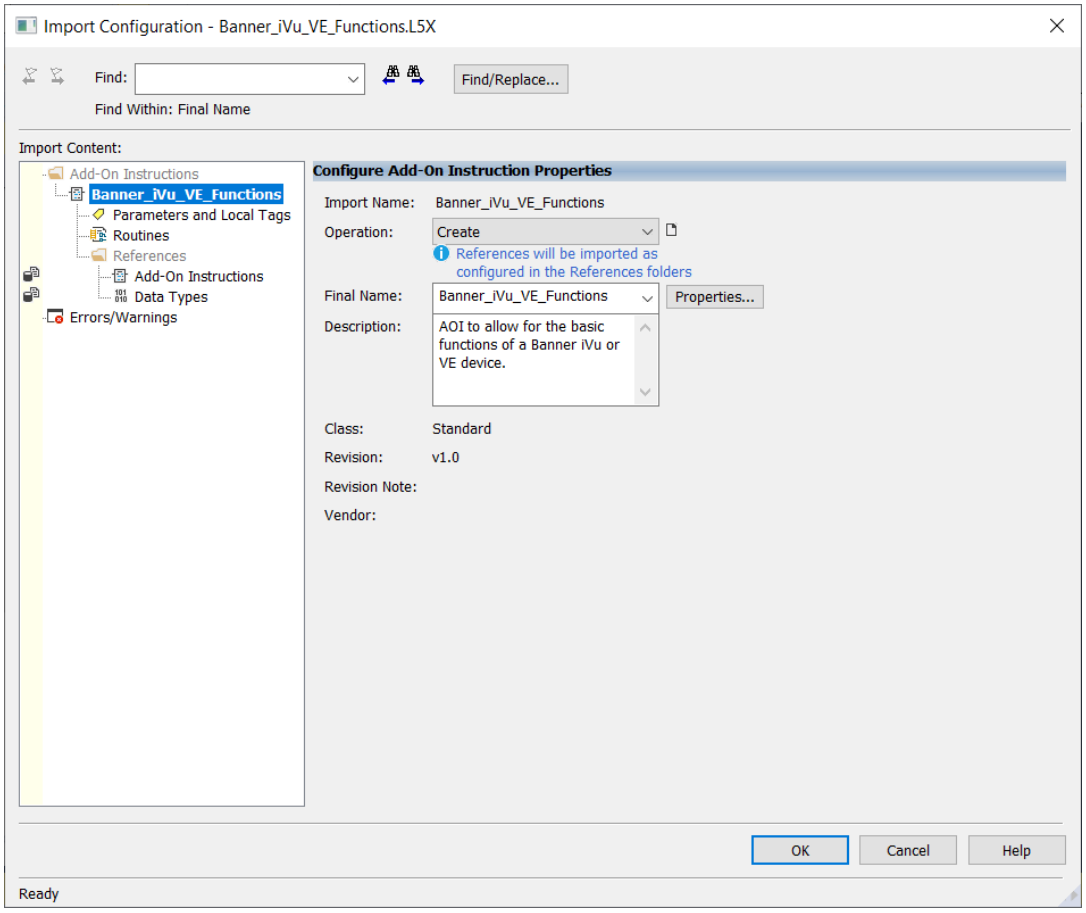
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.

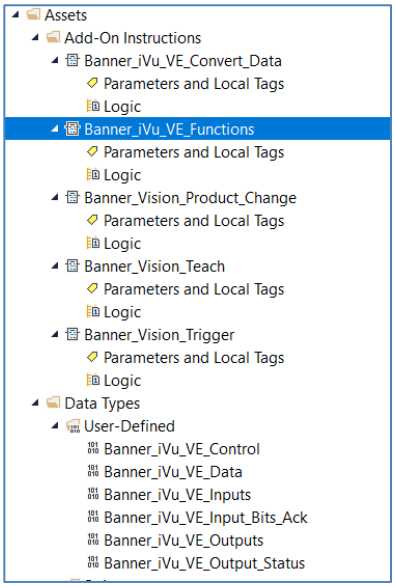


3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner_iVu_VE_Functions.L5X" file will be selected. Click the Open button.



4. The Import Configuration window will pop up. The default selection will create all of the necessary items for the AOI. Click the OK button to complete the import process.



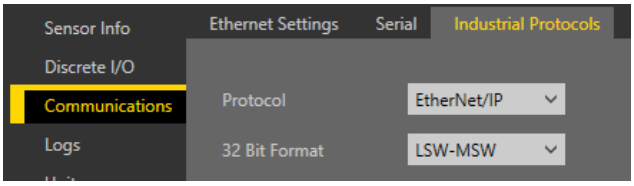


5. The AOI is added to the Controller Organizer window and should look similar to the picture at left.

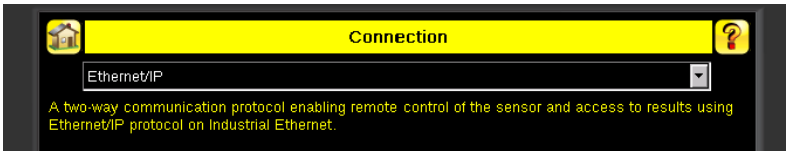
6. AOI installation into the Logix Designer software complete.

2. Connect the iVu or VE to the PLC

Make an EtherNet/IP connection between the PLC and the iVu or VE. If using a VE, be sure that EtherNet/IP is enabled and the format is also left at the default setting of LSW-MSW.



If using an iVu, EtherNet/IP just needs to be enabled.



Create an Ethernet communications module for the iVu or VE. Using the EDS file is easiest; that process is documented in the relevant Instruction Manual, Industrial Ethernet chapter. The controller tags generated include Input (I) and Output (O) Assembly Instances. Each Assembly has a corresponding tag array of INT (16-bit) words. Creating this Class 1 EtherNet/IP implicit IO connection will provide the PLC access to the raw camera data.

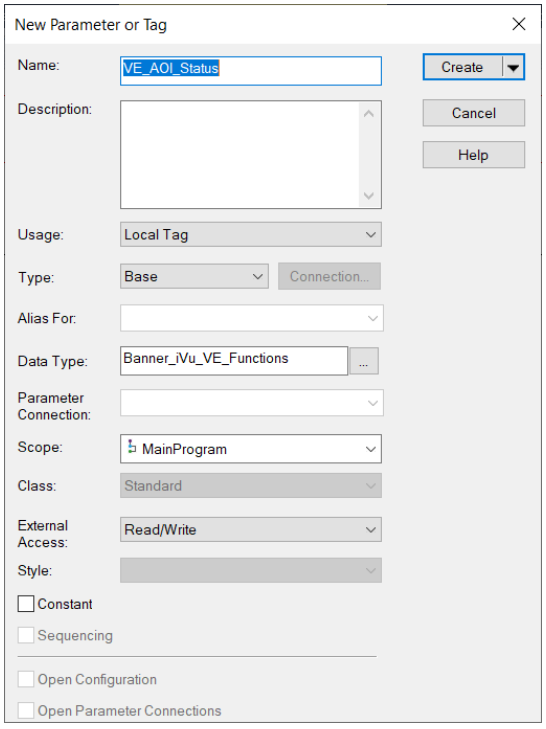
Make sure to select INT as the data size.

3. Configuring the AOI

1. Add the “Banner_iVu_VE_Functions” AOI to your ladder logic program. For each of the question marks shown in the instruction we need to create and link a new tag array. The AOI includes a new type of User Defined Tags (UDT): a custom array of tags meant specifically for this AOI.

2. In the AOI, right-click on the question mark on the line labeled “Banner_iVu_VE_Functions”. Click New Tag. Name the new tag. This example uses the name “VE_AOI_Status”.

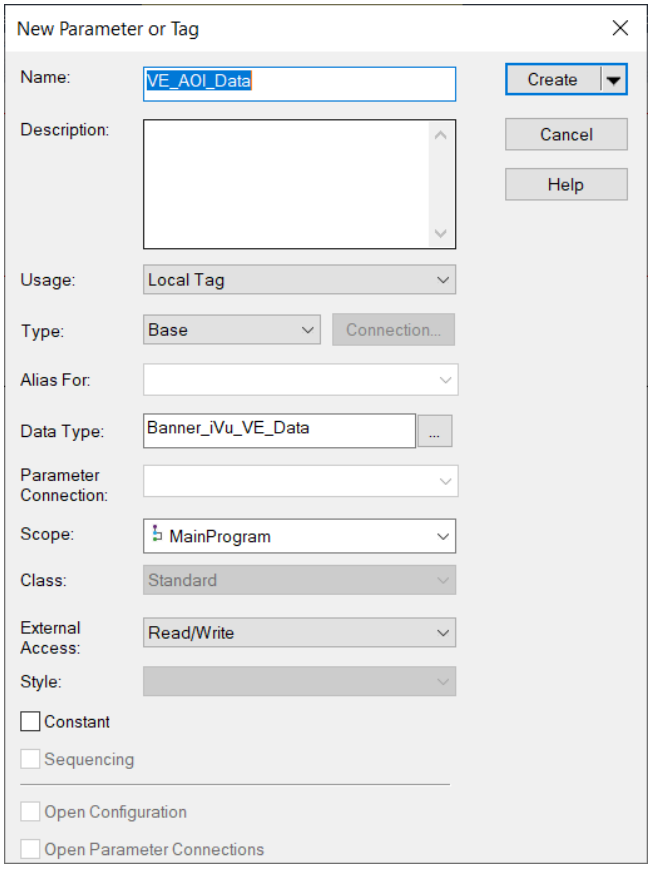
   Note that the Data Type is the User-Defined Data Type (UDT) entitled “Banner_iVu_VE_Functions”. This custom-made array of registers is specially built to handle the memory needs of this AOI. Click Create to make the tag array.



VE_AOI_Status	Local	{...}	{...}		Banner_iVu_VE_Functions
VE_AOI_Status.EnableIn		1		Decimal	BOOL
VE_AOI_Status.EnableOut		1		Decimal	BOOL

3. Now we will right-click on the question mark on the line labeled “Data” in the AOI. Click on “New Tag”. Give the tag a name. This example uses the name “VE_AOI_Data”. Notice that the Data Type is “Banner_iVu_VE_Data”. Click Create.

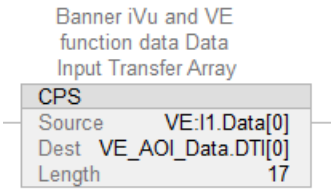
   This array includes the controls we will be using perform the basic camera control functions.



VE_AOI_Data	Banner_iVu_VE_Data	Banner iVu and VE function data
VE_AOI_Data.DTI	INT[17]	Banner iVu and VE function data Data Input Transfer Array
VE_AOI_Data.DTO	INT[34]	Banner iVu and VE function data Data Output Transfer Array
VE_AOI_Data.In	Banner_iVu_VE_Inputs	Banner iVu and VE function data Input Data
VE_AOI_Data.Out	Banner_iVu_VE_Outputs	Banner iVu and VE function data Output Data
VE_AOI_Data.Product_Change	BOOL	Banner iVu and VE function data Tag that controls when a product change should be activated.
VE_AOI_Data.Request_Product_Num	DINT	Banner iVu and VE function data Tag in which the requested product change number is entered.
VE_AOI_Data.Product_Change_Error	BOOL	Banner iVu and VE function data Tag that states when a product change error has occurred.
VE_AOI_Data.Teach	BOOL	Banner iVu and VE function data Tag that controls when a teach should be activated.
VE_AOI_Data.Teach_Request_Error	BOOL	Banner iVu and VE function data Tag that states when a teach request error has occurred.
VE_AOI_Data.Trigger	BOOL	Banner iVu and VE function data Tag that controls when a trigger should be activated.
VE_AOI_Data.Continuous_Trigger	BOOL	Banner iVu and VE function data Tag that controls if triggering will be continuous or not.
VE_AOI_Data.Trigger_Error	BOOL	Banner iVu and VE function data Tag that states when a trigger error has occurred.

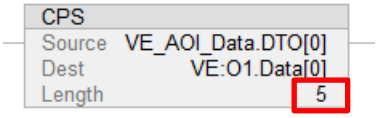
4. The final step required before we download and run the iVu and VE Functions AOI involves a pair of File Synchronous Copy (CPS) instruction. These instructions allow the AOI to grab the raw input data from camera input tags and also to write the output data to the camera's output tags

   Add a CPS instruction before and after the Banner_iVu_VE_Functions AOI on the same ladder rung. For the first CPS command, before the iVu_VE_AOI, choose word 0 of the raw camera input data as the source and choose word 0 of the AOI's incoming data internal array, "DTI", as the destination. Use a length of 17.

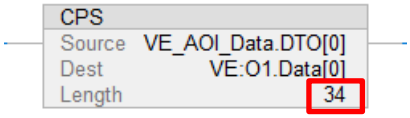


   For the CPS instruction after the iVu_VE_AOI, choose word 0 of the AOI's outgoing data internal array, "DTO", as the source. Select word 0 of the camera's raw output assembly instance tags as the destination. Use a length of 5, unless you are using the VE "String" functions, FTP file name control or barcode data compare control. In that case use 34 instead, and make sure you have configured your Ethernet Module to have an output array of INTs that is 34 words long, in Studio5000.

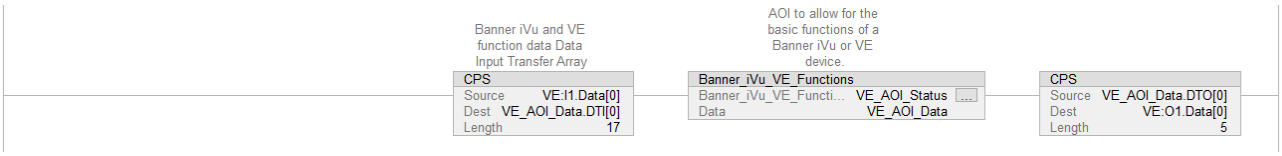
   Typical use:



   Or if using VE "Input String" functions:



   Here is what the entire rung looks like when completed.



   The "Banner_iVu_VE_Functions" AOI is now ready for use.

4. Using the AOI – Basic Control

The “Banner_iVu_VE_Functions” Add-On Instruction has created a group of tags representing the basic controls of the camera, separated out and labeled appropriately. Go to Controller Tags and look for the labeled array named in step 3, above. In this example, we used the name “VE_AOI_Data”. These tags can be used to control the camera correctly. The appropriate timing and logic will be handled behind the scenes by the AOI.

In the first example shown here, a single trigger is sent to the VE camera by simply changing the “Trigger” tag to a 1. The AOI will automatically write the value back to a 0 when the task is complete.

VE_AOI_Data	{...}	{...}		Banner_iVu_VE_Data	Banner iVu and VE function data
VE_AOI_Data.DTI	{...}	{...}	Decimal	INT[17]	Banner iVu and VE function data Data Input Transfer Array
VE_AOI_Data.DTO	{...}	{...}	Decimal	INT[34]	Banner iVu and VE function data Data Output Transfer Array
VE_AOI_Data.In	{...}	{...}		Banner_iVu_VE_Inputs	Banner iVu and VE function data Input Data
VE_AOI_Data.Out	{...}	{...}		Banner_iVu_VE_Outputs	Banner iVu and VE function data Output Data
VE_AOI_Data.Product_Change	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a product cha
VE_AOI_Data.Request_Product_Num	0		Decimal	DINT	Banner iVu and VE function data Tag in which the requested product c
VE_AOI_Data.Product_Change_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a product chan
VE_AOI_Data.Teach	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a teach shoul
VE_AOI_Data.Teach_Request_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a teach request
VE_AOI_Data.Trigger	1		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a trigger shou
VE_AOI_Data.Continuous_Trigger	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls if triggering will be
VE_AOI_Data.Trigger_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a trigger error h

A continuous trigger, provided by the PLC, can be achieved by changing both the “Continuous Trigger” tag and the “Trigger” tag to a 1. The AOI will start a new trigger each time the camera has finished the last inspection, continuing until the “Trigger” tag is changed to 0.

VE_AOI_Data	{...}	{...}		Banner_iVu_VE_Data	Banner iVu and VE function data
VE_AOI_Data.DTI	{...}	{...}	Decimal	INT[17]	Banner iVu and VE function data Data Input Transfer Array
VE_AOI_Data.DTO	{...}	{...}	Decimal	INT[34]	Banner iVu and VE function data Data Output Transfer Array
VE_AOI_Data.In	{...}	{...}		Banner_iVu_VE_Inputs	Banner iVu and VE function data Input Data
VE_AOI_Data.Out	{...}	{...}		Banner_iVu_VE_Outputs	Banner iVu and VE function data Output Data
VE_AOI_Data.Product_Change	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a product cha
VE_AOI_Data.Request_Product_Num	0		Decimal	DINT	Banner iVu and VE function data Tag in which the requested product c
VE_AOI_Data.Product_Change_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a product chan
VE_AOI_Data.Teach	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a teach shoul
VE_AOI_Data.Teach_Request_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a teach request
VE_AOI_Data.Trigger	1		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a trigger shou
VE_AOI_Data.Continuous_Trigger	1		Decimal	BOOL	Banner iVu and VE function data Tag that controls if triggering will be
VE_AOI_Data.Trigger_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a trigger error h

The camera Product Change function requires first a number equal to the desired inspection slot number to be written into the “Request_Product_Num” tag. Afterward, write the “Product Change” tag to a 1. The AOI will automatically write the value back to a 0 when the task is complete.

◢ VE_AOI_Data	{...}	{...}		Banner_iVu_VE_Data	Banner iVu and VE function data
▷ VE_AOI_Data.DTI	{...}	{...}	Decimal	INT[17]	Banner iVu and VE function data Data Input Transfer Array
▷ VE_AOI_Data.DTO	{...}	{...}	Decimal	INT[34]	Banner iVu and VE function data Data Output Transfer Array
▷ VE_AOI_Data.In	{...}	{...}		Banner_iVu_VE_Inputs	Banner iVu and VE function data Input Data
▷ VE_AOI_Data.Out	{...}	{...}		Banner_iVu_VE_Outputs	Banner iVu and VE function data Output Data
VE_AOI_Data.Product_Change	1		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a product cha
▷ VE_AOI_Data.Request_Product_Num	3		Decimal	DINT	Banner iVu and VE function data Tag in which the requested product c
VE_AOI_Data.Product_Change_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a product chan
VE_AOI_Data.Teach	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a teach shoul
VE_AOI_Data.Teach_Request_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a teach request
VE_AOI_Data.Trigger	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a trigger shou
VE_AOI_Data.Continuous_Trigger	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls if triggering will be
VE_AOI_Data.Trigger_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a trigger error h

The camera Teach function applies to any Match or Barcode tool in the camera inspection. Set the “Teach” tag to 1, and any Match tool will try to save the pattern it sees when the next image is triggered, and use this new pattern for comparing against in future images. Any Barcode tool will try to save the first barcode read as the Data Compare string. If there is no Match or Barcode tool in the inspection, the “Teach_Request_Error” bit will be set to 1. If there is a Match tool, but the teach fails due to poor image quality, then a different bit will turn on, the “In.Output_Bits_Status.Teach_Error” bit.

◢ VE_AOI_Data	{...}	{...}		Banner_iVu_VE_Data	Banner iVu and VE function data
▷ VE_AOI_Data.DTI	{...}	{...}	Decimal	INT[17]	Banner iVu and VE function data Data Input Transfer Array
▷ VE_AOI_Data.DTO	{...}	{...}	Decimal	INT[34]	Banner iVu and VE function data Data Output Transfer Array
▷ VE_AOI_Data.In	{...}	{...}		Banner_iVu_VE_Inputs	Banner iVu and VE function data Input Data
▷ VE_AOI_Data.Out	{...}	{...}		Banner_iVu_VE_Outputs	Banner iVu and VE function data Output Data
VE_AOI_Data.Product_Change	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a product cha
▷ VE_AOI_Data.Request_Product_Num	0		Decimal	DINT	Banner iVu and VE function data Tag in which the requested product c
VE_AOI_Data.Product_Change_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a product chan
VE_AOI_Data.Teach	1		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a teach shoul
VE_AOI_Data.Teach_Request_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a teach request
VE_AOI_Data.Trigger	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls when a trigger shou
VE_AOI_Data.Continuous_Trigger	0		Decimal	BOOL	Banner iVu and VE function data Tag that controls if triggering will be
VE_AOI_Data.Trigger_Error	0		Decimal	BOOL	Banner iVu and VE function data Tag that states when a trigger error h

5. Using the AOI – Parsed Data

Another feature of this AOI is the raw camera data is parsed out and labeled. The group of tags created in step 3 of the setup above (page 6 of this document) includes both “In” and “Out” tag arrays. Below, we see the “In” tag array expanded. This tag array includes information coming into the PLC from the camera, labeled and displayed with the correct data type. Notice Current_Inspection_Time is shown as a Float, Fail_Count is a DINT (32-bit integer), and the various Input_Bit_Ack (Acknowledge bits) are each BOOL. A similar list of labeled bits can be found in the “Output_Bits_Status” tag array. This nice ordering of the data is possible through the use of the “Banner_iVu_VE_Data” User-Defined Data Type used in the AOI. This data includes all result data EXCEPT for individual tool result values that are added to the custom mapping by the user, such as distance measurements. This type of data is not found in the AOI, it is accessed through the raw data INT tags created upon adding the Ethernet Module to the PLC project.

◢ VE_AOI_Data.In	{...}	{...}		Banner_iVu_VE_Inputs
◢ VE_AOI_Data.In.Input_Bits_Ack	{...}	{...}		Banner_iVu_VE_Input_Bits_Ack
VE_AOI_Data.In.Input_Bits_Ack.Product_Change_Ack	0		Decimal	BOOL
VE_AOI_Data.In.Input_Bits_Ack.Teach_Latch_Ack	0		Decimal	BOOL
VE_AOI_Data.In.Input_Bits_Ack.Trigger_Ack	0		Decimal	BOOL
VE_AOI_Data.In.Input_Bits_Ack.iVu_Command_Ack	0		Decimal	BOOL
VE_AOI_Data.In.Input_Bits_Ack.VE_Barcode_Data_Compare_Input_String_Ack	0		Decimal	BOOL
VE_AOI_Data.In.Input_Bits_Ack.VE_FTP_Input_String_Ack	0		Decimal	BOOL
◢ VE_AOI_Data.In.Output_Bits_Status	{...}	{...}		Banner_iVu_VE_Output_Status
VE_AOI_Data.In.Output_Bits_Status.Ready	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Pass_Fail	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Read_No_Read_iVu	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Ready_to_Latch	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_1	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_2	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_3	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_4_VE	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_5_VE	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Missed_Trigger	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Teach_Error	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.System_Error	0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Execution_Error	0		Decimal	BOOL

▸ VE_AOI_Data.In.Error_Code	0		Decimal	DINT
▸ VE_AOI_Data.In.Inspection_Num	0		Decimal	DINT
▸ VE_AOI_Data.In.Iteration_Count	0		Decimal	DINT
▸ VE_AOI_Data.In.Pass_Count	0		Decimal	DINT
▸ VE_AOI_Data.In.Fail_Count	0		Decimal	DINT
▸ VE_AOI_Data.In.Missed_Triggers	0		Decimal	DINT
VE_AOI_Data.In.Current_Inspection_Time	0.0		Float	REAL
◂ VE_AOI_Data.In.Sensor_Pass_Fail	0		Decimal	INT
VE_AOI_Data.In.Sensor_Pass_Fail.0	0		Decimal	BOOL
VE_AOI_Data.In.Sensor_Pass_Fail.1	0		Decimal	BOOL
VE_AOI_Data.In.Sensor_Pass_Fail.2	0		Decimal	BOOL
VE_AOI_Data.In.Sensor_Pass_Fail.3	0		Decimal	BOOL
VE_AOI_Data.In.Sensor_Pass_Fail.4	0		Decimal	BOOL
VE_AOI_Data.In.Sensor_Pass_Fail.5	0		Decimal	BOOL

= These tags only update at the end of an inspection when the Ready signal turns back ON

The “Out” tag array is expanded below. Again, the raw data from the communication module connection to the camera has been parsed. Here we see that the Control Register has individual BOOL commands, and that the native data type for the Input_String is correctly shown as a 26 character string. Tags with “Reserved” in them are meant for use by the AOI logic, and should not be overwritten. The other tags here can be used for advanced functions like sending input strings to the VE.

◂ VE_AOI_Data.Out	{...}	{...}		Banner_iVu_VE_Outputs
◂ VE_AOI_Data.Out.Control_Registers	{...}	{...}		Banner_iVu_VE_Control
VE_AOI_Data.Out.Control_Registers.Reserved_PC	0		Decimal	BOOL
VE_AOI_Data.Out.Control_Registers.Reserved_TL	0		Decimal	BOOL
VE_AOI_Data.Out.Control_Registers.Reserved_TR	0		Decimal	BOOL
VE_AOI_Data.Out.Control_Registers.iVu_Command	0		Decimal	BOOL
VE_AOI_Data.Out.Control_Registers.VE_Barcode_Data_Compare_Input_String	0		Decimal	BOOL
VE_AOI_Data.Out.Control_Registers.VE_FTP_Input_String	0		Decimal	BOOL
▸ VE_AOI_Data.Out.Reserved_PCN	0		Decimal	DINT
▸ VE_AOI_Data.Out.VE_Input_String_Tool_Index	0		Decimal	SINT
▸ VE_AOI_Data.Out.VE_Input_String	''	{...}		STRING_26

6. Execution Errors

The “Output_Bits_Status” tag array has an Execution_Error bit that is worth mentioning. This bit turns to a 1 when something goes wrong. Most likely, the issue is a Product Change attempt to an invalid memory location (a numbered slot in memory with no program stored in it). In the case below, we see that the Execution_Error bit is a 1, and, further down the list, we also see a report of a Product_Change_Error being equal to 1, too. This was a test case, wherein the Request_Product_Num value was kept at 0, then the Product_Change bit was written to a 1. There is never a valid inspection stored in memory location 0, therefore the error.

Name	Usage	Value	Force Mas	Style	Data Type
▸ VE_AOI_Data.In.Missed_Triggers		0		Decimal	DINT
◂ VE_AOI_Data.In.Output_Bits_Status		{...}	{...}		Banner_iVu_VE_Output_St
VE_AOI_Data.In.Output_Bits_Status.Execution_Error		1		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Missed_Trigger		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_1		1		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_2		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_3		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Pass_Fail		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Read_No_Read		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Ready		1		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Ready_to_Latch		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.System_Error		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Teach_Error		0		Decimal	BOOL
▸ VE_AOI_Data.In.Pass_Count		0		Decimal	DINT
▸ VE_AOI_Data.In.Sensor_Pass_Fail		15		Decimal	INT
▸ VE_AOI_Data.Out		{...}	{...}		Banner_iVu_VE_Outputs
VE_AOI_Data.Product_Change		0		Decimal	BOOL
VE_AOI_Data.Product_Change_Error		1		Decimal	BOOL
▸ VE_AOI_Data.Request_Product_Num		0		Decimal	DINT
VE_AOI_Data.Teach		0		Decimal	BOOL
VE_AOI_Data.Teach_Error		0		Decimal	BOOL

Clearing the Execution_Error is simply a matter of performing a valid PLC command, either a valid Product Change attempt or, as shown below, a successful Trigger. In either case, the Execution_Error bit is written back to a 0.

Name	Usage	Value	Force Mas	Style	Data Type
VE_AOI_Data.In.Missed_Triggers		0		Decimal	DINT
VE_AOI_Data.In.Output_Bits_Status		{...}	{...}		Banner_iVu_VE_Output_Stat
VE_AOI_Data.In.Output_Bits_Status.Execution_Error		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Missed_Trigger		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_1		1		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_2		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Output_3		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Pass_Fail		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Read_No_Read		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Ready		1		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Ready_to_Latch		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.System_Error		0		Decimal	BOOL
VE_AOI_Data.In.Output_Bits_Status.Teach_Error		0		Decimal	BOOL
VE_AOI_Data.In.Pass_Count		0		Decimal	DINT
VE_AOI_Data.In.Sensor_Pass_Fail		15		Decimal	INT
VE_AOI_Data.Out		{...}	{...}		Banner_iVu_VE_Outputs
VE_AOI_Data.Product_Change		0		Decimal	BOOL
VE_AOI_Data.Product_Change_Error		1		Decimal	BOOL
VE_AOI_Data.Request_Product_Num		0		Decimal	DINT
VE_AOI_Data.Teach		0		Decimal	BOOL
VE_AOI_Data.Teach_Error		0		Decimal	BOOL
VE_AOI_Data.Trigger		0		Decimal	BOOL