# BANNER PRESENCEPLUS APPLICATION NOTE

## ABSTRACT

The purpose of this document is to describe the G3's support for Banner's PresencePlus series of vision systems, and the configuration steps necessary to use this feature.

## INTRODUCTION

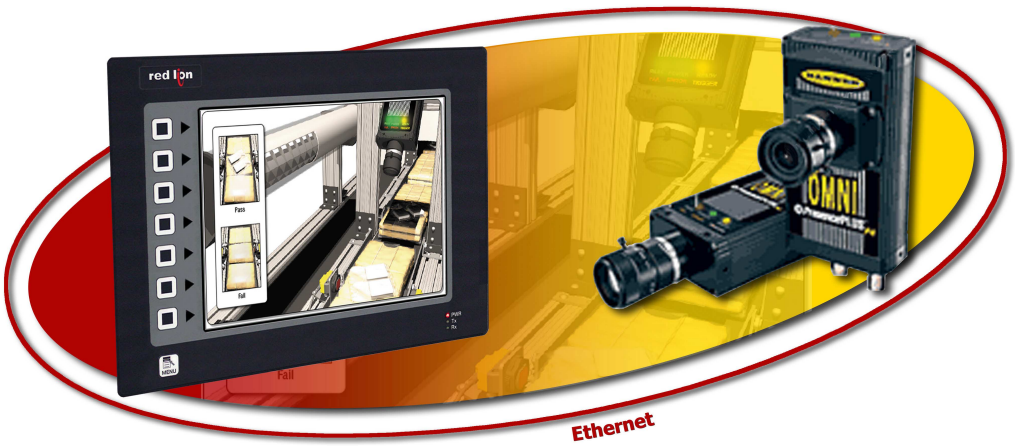
Banner is a manufacturer of vision sensors. One of their specialties is vision and object recognition. All the PresencePlus series of vision sensors are now supported by Red Lion Controls G3 series HMIs. Only graphical color units support this feature.

This document will first describe how to configure a G310 HMI to communicate and transfer data with a PresencePlus vision sensor, then how the sensor should be configured.

This document will step though the procedures to setup the communication for the following functions:

- Trigger (Input Flag)
- Pass Output (Output Flag)
- Pass Count (Integer value contained in a register)
- Measurement (Floating Point Number value contained in a register)
- Tool name (String value contained in a register)

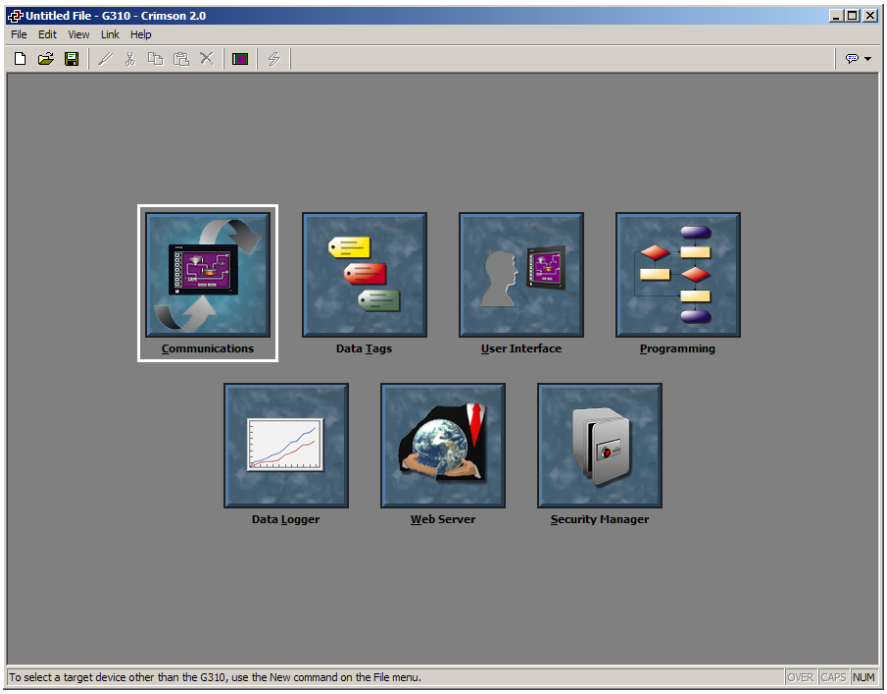
As an example, the PresencePlus P4 OMNI vision sensor will be measuring the opening of a Banner Fiber cutter (PFC-2).



# TABLE OF CONTENTS

# G3 CONFIGURATION – DISPLAYING THE IMAGE

Red Lion Controls' G3 HMIs are programmed using Crimson 2.0 software. This software is free and available on the internet at www.redlion.net



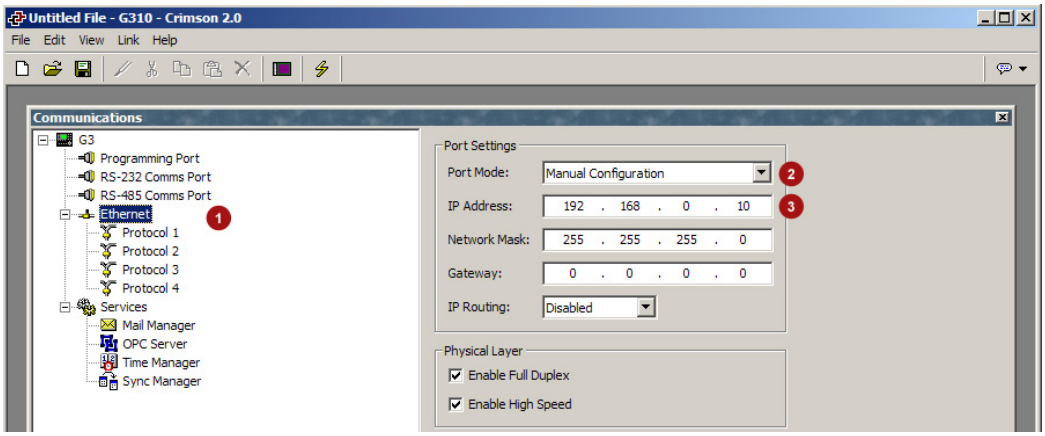

Displaying the camera image on the G3's display is achieved with the following steps:

a. Activating the Ethernet port,
b. Setting up the Banner Camera driver,
c. Inserting the camera primitive on the screen.

**Activating the Ethernet port**

The configuration of the Ethernet port is done in the Communication Module.

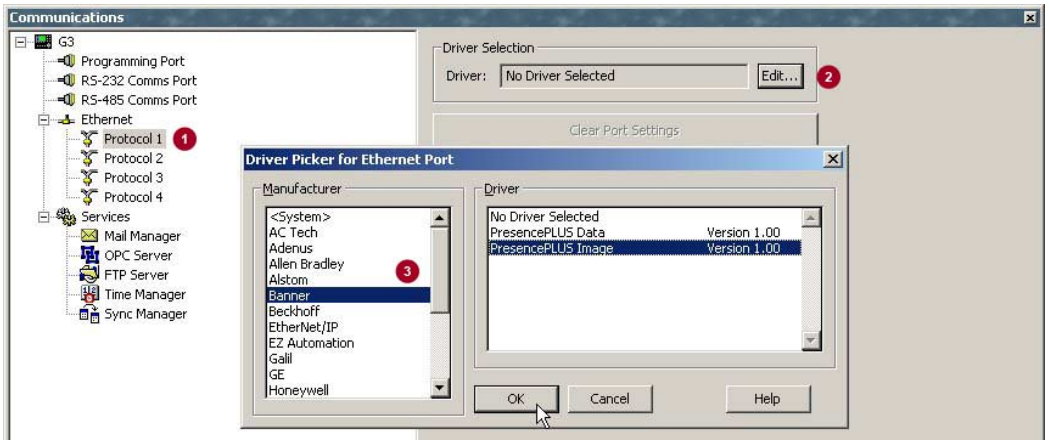
1. Click on the word "Ethernet". This will display the Ethernet settings of the HMI.
2. Change Port Mode: to **Manual Configuration**.
3. Change the IP address of the HMI to an appropriate IP address (i.e. 192.168.0.10)



**Setting up the Banner Camera driver**

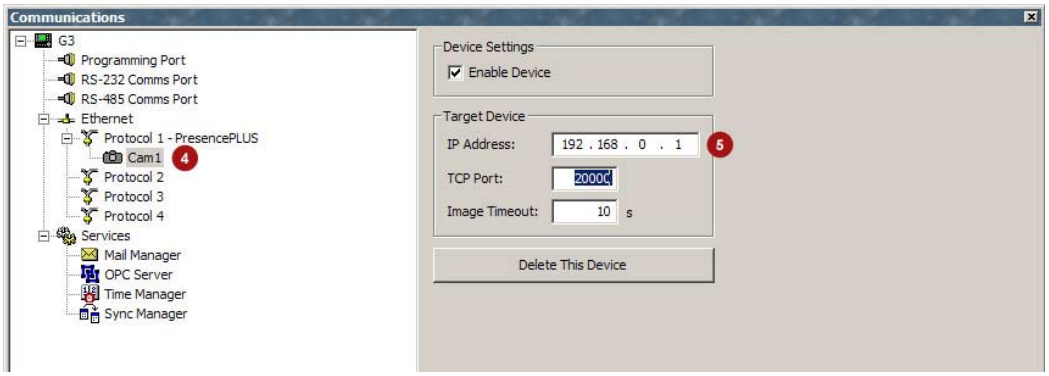
The driver for the Banner Camera is configured on one the Ethernet protocols.

1. Click on Protocol 1
2. Click Edit to open the driver selection box
3. Pick the Banner PresencePlus Image driver




Then, configure the camera settings. Multiple cameras can be supported under this protocol by clicking Add Additional Device

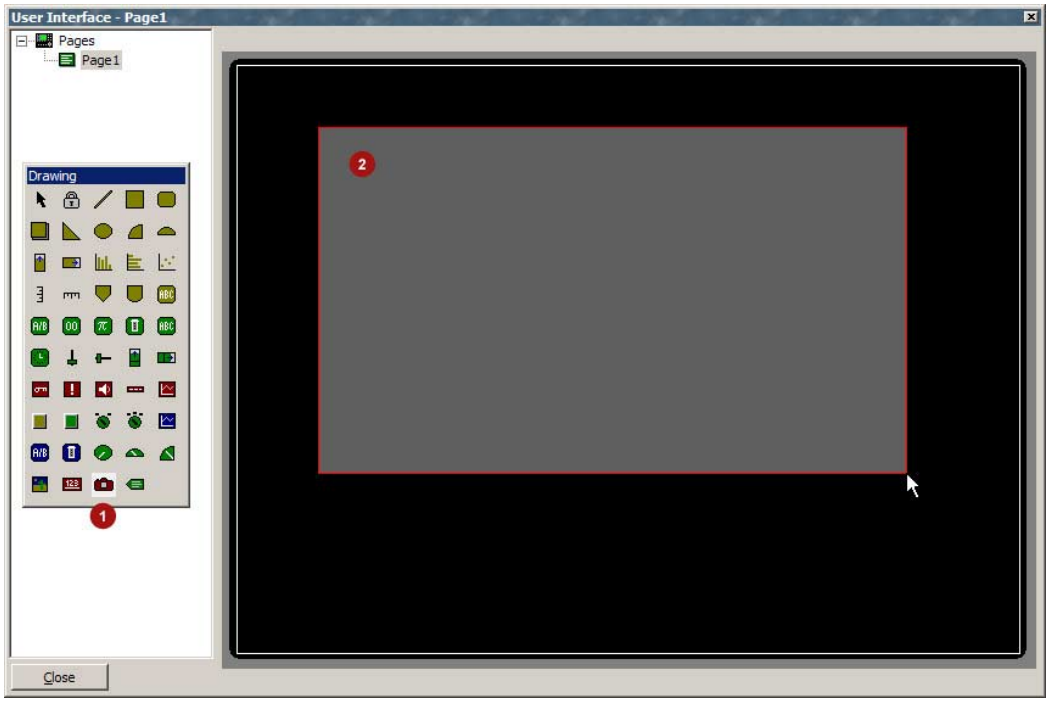
4. Click on Cam1 to select the camera
5. Enter the camera's IP Address and TCP port



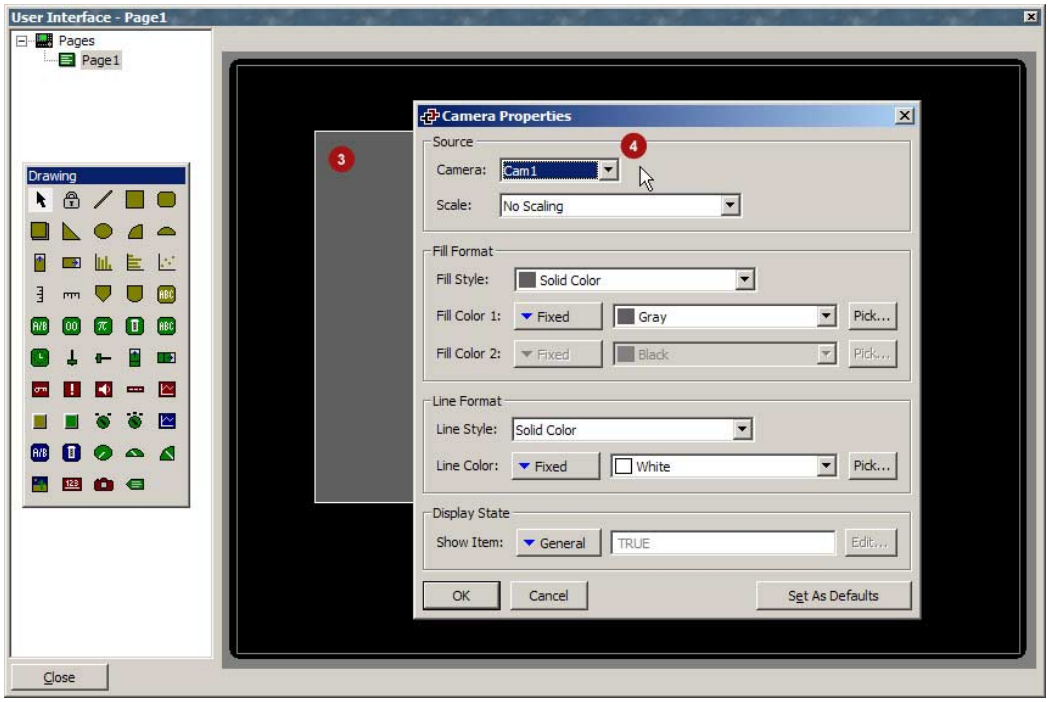

**Inserting the camera primitive on the screen**

To insert the camera picture on the screen, enter the User Interface module.

1. Select the Camera primitive in the Drawing box
2. Click and drag on the screen to insert the primitive




3. Double-click the primitive to open the properties
4. Select the camera that was configure in the Communication module




The HMI is now ready to receive the Banner Camera images. The image will be displayed in the gray area defined by the primitive. If the image is too small, the scale property in the camera primitive can be adjusted.

**NOTE**: The HMI will only display an image if the camera is configured to export the image, as shown later in this document.

## SETTING UP THE G3 – INSPECTION FILE TRANSFER

This functionality allows the transfer of inspection files from the G3 Compact Flash card to the camera memory. Built in functions provide both download and upload as well as inspection file selection.

This feature uses the Banner camera driver, therefore the Ethernet port has to be setup as shown in the first section of this document; Setting up the G3 – Displaying the picture.'

Refer to the Crimson 2.0 manual for more details on how to run user programs.

**NOTE**: For the vision sensor to accept product changes from the G3 HMI, the run time setting in the camera must be set to hardware input in the Select tab of the run screen. Please refer to the section *Setting up the camera - Configure the vision sensor to accept product changes* later in this document for more details.

### Loading files to the Camera

The following function loads the inspection file from the G3 CompactFlash card into the camera's memory. This function can be called in a user program.

***NOTE**: The program should be run in the background so the G3 has enough time to access the CompactFlash card.

**LOADCAMERASETUP(PORT, DEVICE, FILENB, FILENAME)**

| ARGUMENT | TYPE | DESCRIPTION |
|---|---|---|
| port | int | Port number where the Banner camera is connected |
| device | int | Camera device number |
| fileNb | int | Inspection file number in the camera |
| fileName | cstring | Path and filename for the inspection file on the G310 Compact Flash card |

This function returns true if the transfer is successful, false otherwise.

Example: Result = LoadCameraSetup(4, 0, 1, "\\in0.isp");

### Saving files from the Camera

The following function saves the inspection file uploaded from the camera on the G3 CompactFlash card. This function can be called in a user program.

***NOTE**: The program should be run in the background so the G3 has enough time to access the CompactFlash card.

**SAVECAMERASETUP(PORT, DEVICE, FILENB, FILENAME)**

| ARGUMENT | TYPE | DESCRIPTION |
|---|---|---|
| port | int | Port number where the Banner camera is connected |
| device | int | Camera device number |

| | | |
|---|---|---|
| fileNb | int | Inspection file number in the camera |
| fileName | cstring | Path and filename for the inspection file on the G310 Compact Flash card |

This function returns true if the transfer is successful, false otherwise.

Example: Result = SaveCameraSetup(4, 0, 1, "\\in0.isp");

### Selecting the inspection file in the camera

The following function will select the inspection file to be used by the camera.

***NOTE**: The program should be run in the background to allow the camera enough time to change files.

**USECAMERASETUP(PORT, DEVICE, FILENB)**

| ARGUMENT | TYPE | DESCRIPTION |
|---|---|---|
| port | int | Port number where the Banner camera is connected |
| device | int | Camera device number |
| fileNb | int | Inspection file number in the camera |

This function returns true if the file change is successful, false otherwise.

Example: Result = UseCameraSetup(4, 0, 1);

### Program examples for inspection file transfer and selection

In this example, the G3 has one internal variable named "status" to display the status of the upload/download/change process.

1. Open the tag window
2. Create an internal string variable called status

There are three basic programs needed for this demo; one to store an inspection to the CF, one to load an inspection from to CF, and one to initiate a product change. Storing and loading inspections run in the background, the vision sensor can processes inspections while transferring inspections.
**Note: A product change must be initiated to run an inspection that has been loaded onto the sensor.**

This example will store inspections from memory location 1 on the OMNI to the CompactFlash card. Program name: "SaveAsInp1".

```
Status = "Saving";

Sleep(10);

if ( SaveCameraSetup(4, 0, 1, "\\in1.isp))

       Status = "Done";

else

       Status = "Failed";
```

This example will load the inspection from the CompactFlash card to memory location 1 in the OMNI Vision sensor. Program name: "LoadInp1"

```
Status = "Loading";

Sleep(10):

if( LoadCameraSetup(4, 0, 1, "\\in1.isp") )

       Status = "Done";

else

       Status = "Failed";
```

This example will initiate a product change by selecting the inspection in memory location 1 of the OMNI vision sensor. Program name: "UseInp1"

```
Status = "Working";

Sleep(10);

if( UseCameraSetup(4, 0, 1) )

       Status = "Done";

else

       Status = "Failed";
```

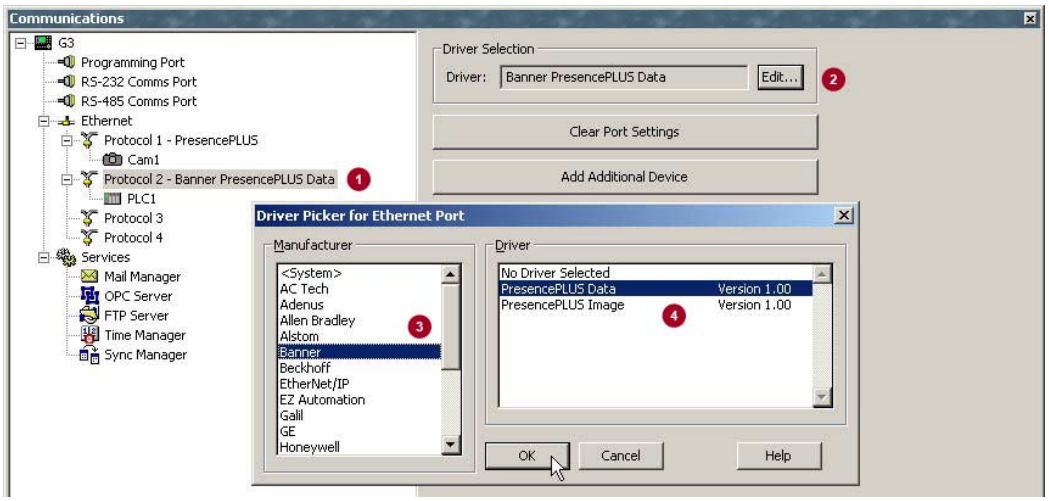
Programs can be executed using a push button or an event action field in the G3.

# SETTING UP THE G3 – DATA EXCHANGE

In order to read/write data from the camera, the PresencePlus data driver is used. This driver provides access to standard camera information and also user defined Modbus registers in the camera.

**Configuring the PresencePLUS data driver.**

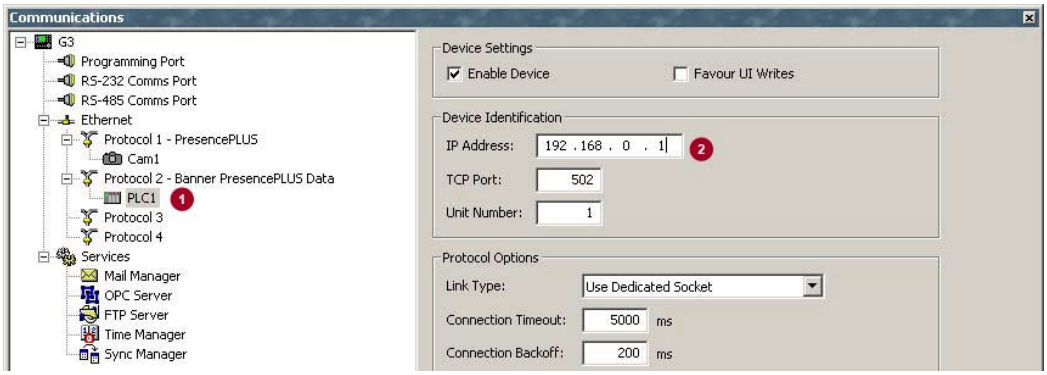
1. Click on Protocol 2
2. Edit the Driver Selection
3. Choose Banner
4. Choose PresencePLUS Data




**Configure the Device Settings**

1. Click on "PLC1" (If desired, rename it to "PresencePLUS")
2. Change the IP address to match the Sensor's IP address (i.e. 192.168.0.1)

Close out the Communications Window



**Information on Data Tags**

All the standard information available in the PresencePLUS vision sensor can be accessed directly with the driver and do not require Modbus mapping.

In this example, the following information can be accessed directly:

- Trigger (Input Flag)
- Pass Output (Output Flag)
- Pass Count (Integer value contained in a register)

However, the following two will be user defined when programming the vision sensor:

- Measurement (Floating Point Number value contained in a register)
- Tool name (String value contained in a register)

**NOTE: In the HMI driver, Input and Output are defined from the PresencePLUS Sensor point of view. Therefore, the G3 will write data the sensor input and read data from the sensor output.**

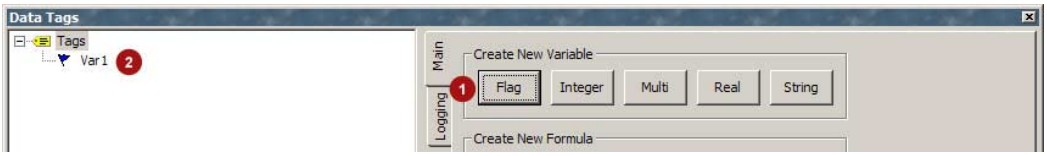
For more information on how Modbus/TCP registers are organized for the PresencePLUS vision sensors, refer to the Appendix.

**Configuring Data Tags**

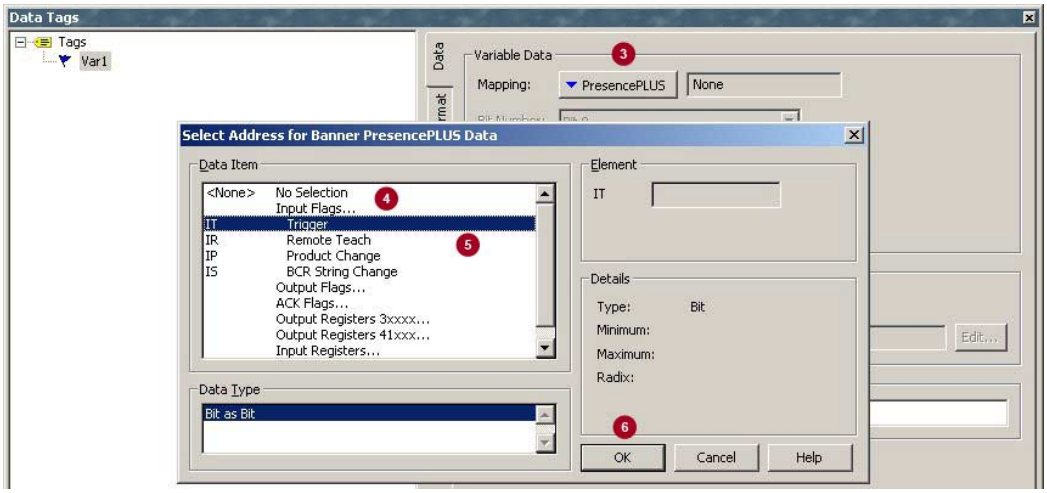
<u>Configuring sensor defined tags</u>

Create a Trigger Coil (Flag) Tag

1. Click on Flag
2. Click on the newly created Flag variable




3. Under Mapping, Select PresencePLUS (PLC1)
4. Choose **Input Flags**
5. Select **IT** – Trigger
6. Click OK.



7. Change Access to **Write Only** (not shown)

This Tag is finished; click "tags" on the explorer bar window to get back to the main tags page. Creating the other tags is very similar.

Create a Pass Output (Flag) Tag

1. Click "Create a new variable" **Flag** Button
2. Select the new variable in the explorer window
3. Under Mapping select PresencePLUS (PLC1)
4. Select **OP** – Pass
5. Click OK
6. Change Access to Read Only

Create a Pass Count (Integer) Tag

1. Click "Create a new variable" **Integer** Button
2. Select the new variable explorer window
3. Under Mapping select PresencePLUS (PLC1)
4. Choose **Output Registers 31xxx**
5. Select **O3P** – Pass Count
6. Click OK
7. Change Access to Read Only
8. Change Sign mode to Treat as Unsigned

<u>Configuring user defined tags</u>

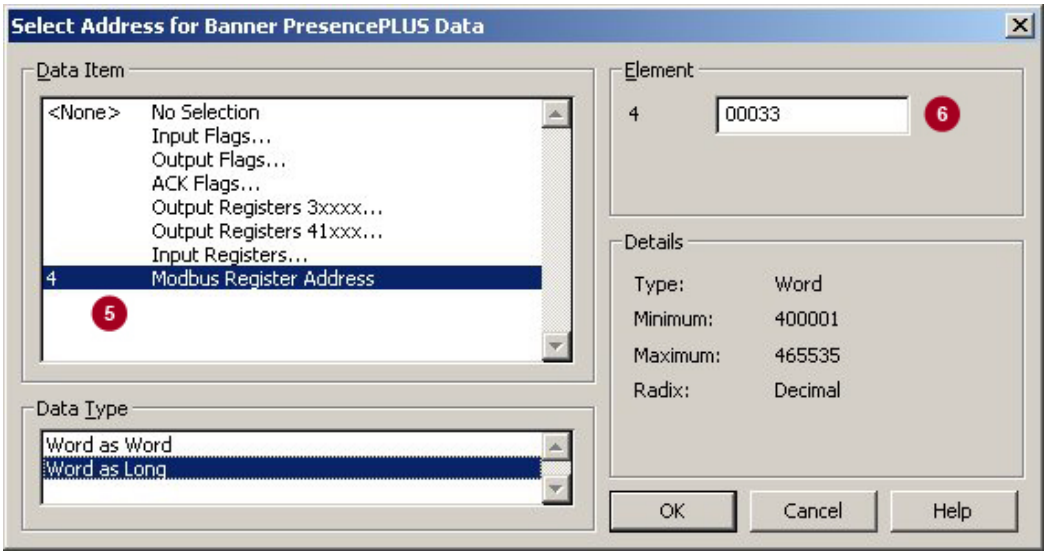
The following tags will be set up in the PresencePlus sensor in the section "Setting up the camera". Tag values will be linked to Modbus register addresses in the sensor. The set up will be as follow:

- Measurement – Register 35 (400033)
- Tool name – Register 37 (400035)

The generic part of the PresencePLUS data driver has to be used to access these registers.

Create a Height Measurement (Real) Tag

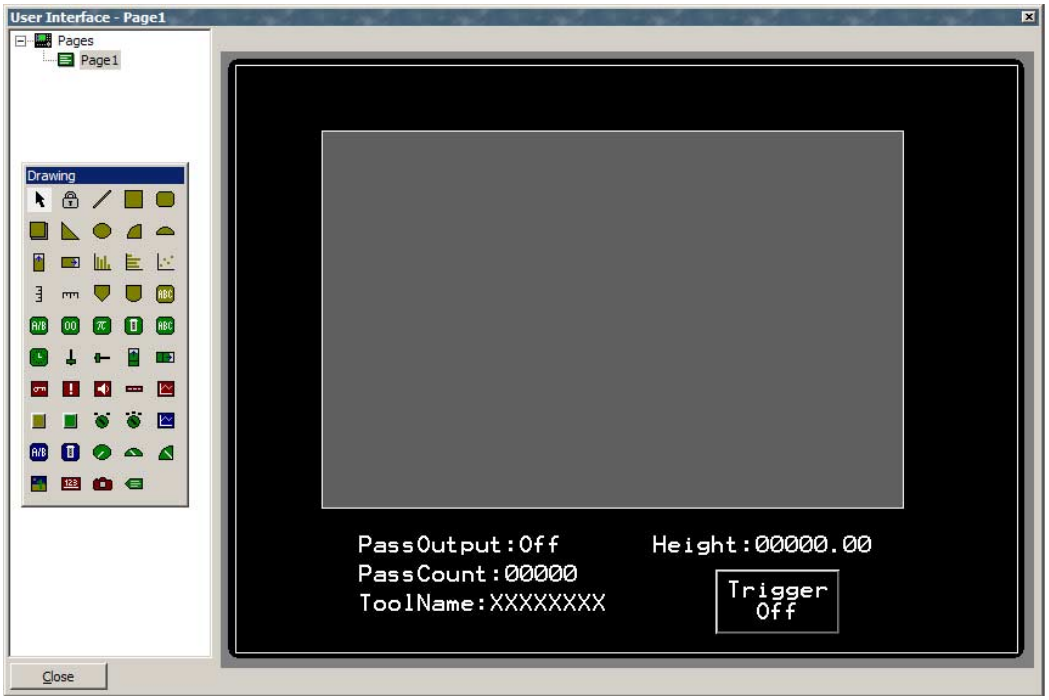
1. Click "Create a new variable" **Real** Button
2. Select the new variable explorer window
3. Under Mapping select PresencePLUS (PLC1)
4. Choose **Modbus Register Address**
5. Enter Element **400033** (shown below) (See Appendix A)
6. In Data Type choose **Word as Long** (shown below)
7. Click OK
8. Change Access to Read Only
9. Click on the Format tab
10. Change the digits after the DP (Decimal Point) to 2




Create a Tool Name (String) Tag

1. Click "Create a new variable" **String** Button
2. Select the new variable explorer window
3. Select the Data tab
4. Under Mapping select PresencePLUS (PLC1)
5. Choose **Modbus Register Address**
6. Enter Element **400035** (See Appendix A)
7. Click OK
8. Choose Encoding: **Packed High-to-Low**
9. Choose Access: **Read Only**
10. Choose Length: **20**

All the necessary tags are created. Close out the Data Tag window and open the User Interface window. In the User Interface Window add text to display the tag values, and a button for the trigger.



# SETTING UP THE CAMERA

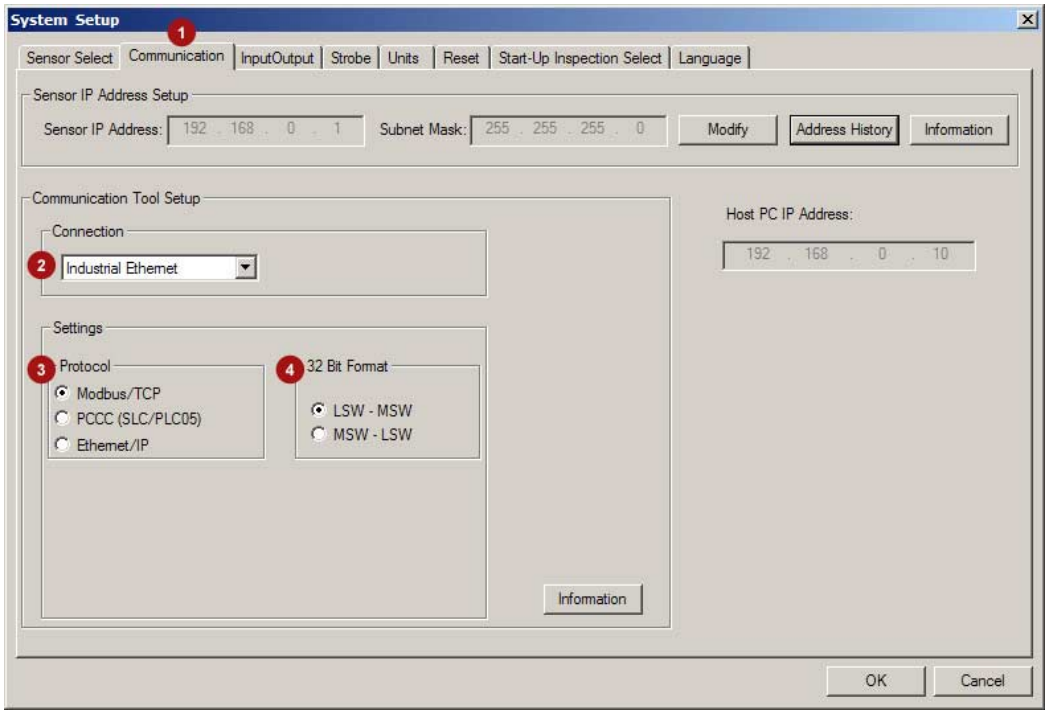
The PresencePlus vision sensor setup is done in 3 steps:

a. Configure the System settings
b. Configure the Inspection
c. Configure the communication tool

For this example PresencePLUS PC software version 2.7.0 was used with a P4 OMNI version 1.1.0. Setting up Modbus/TCP is identical for all PresencePlus vision sensors.

### Configuring the system settings

Click on the "System" button in the upper right hand corner, and the following dialog box should appear:

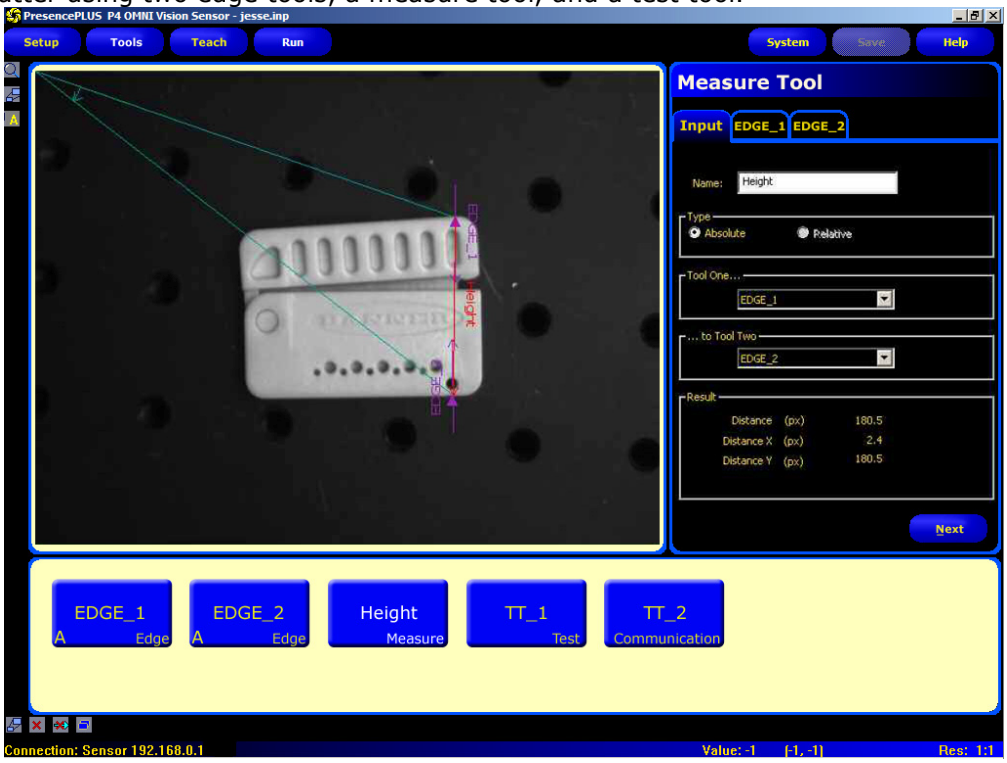



1. Go to the Communication tab
2. Choose Industrial Ethernet
3. Choose the Protocol called "Modbus/TCP"
4. Leave the 32 Bit format at the default setting "LSW-MSW". Press the OK button to accept this selection.

After the system settings are configured, the Banner vision sensor automatically transfers the discrete I/O and inspection Pass / Fail information through the Industrial Ethernet. The communication tool data is only needed if you would like to send vision tool results, like measurement values, tool names, BLOB Counts, or Bar Code strings to the remote device.

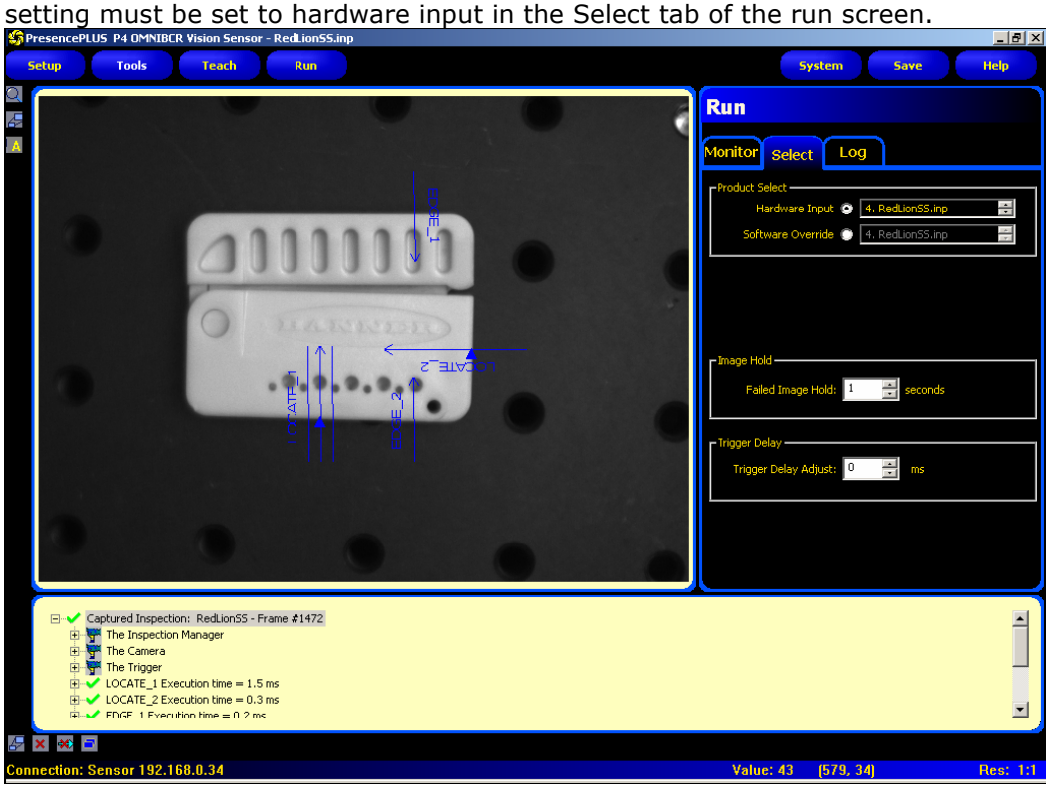
## Configuring the Inspection

In this example, we will create an inspection measuring the height of the fiber cutter using two edge tools, a measure tool, and a test tool.




## Configure the vision sensor to accept product changes

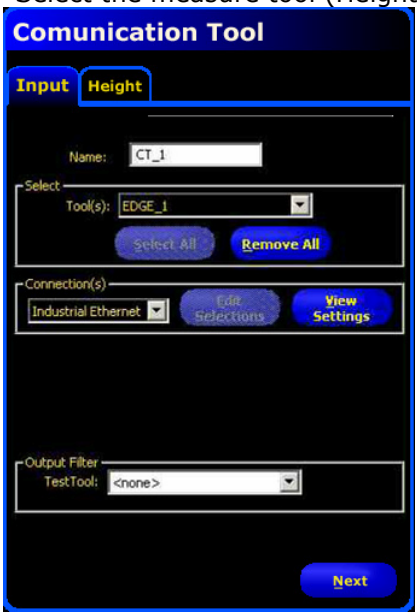
For the vision sensor to accept product changes from the G3 HMI, the run time setting must be set to hardware input in the Select tab of the run screen.

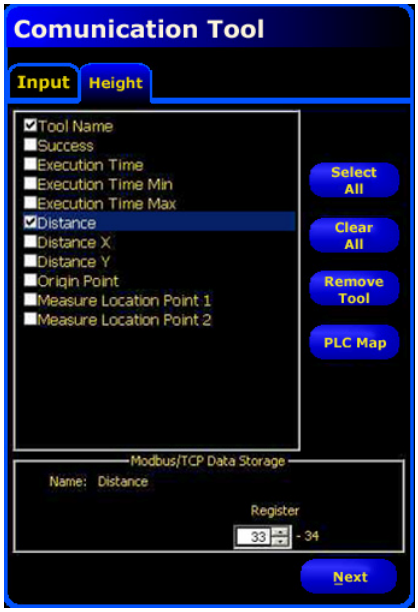


**Configuring the Communication Tool**

After the test tool we add a communication tool to export the measure tool name (Height) and the measurement distance (Height). The communication tool is configured as the following:

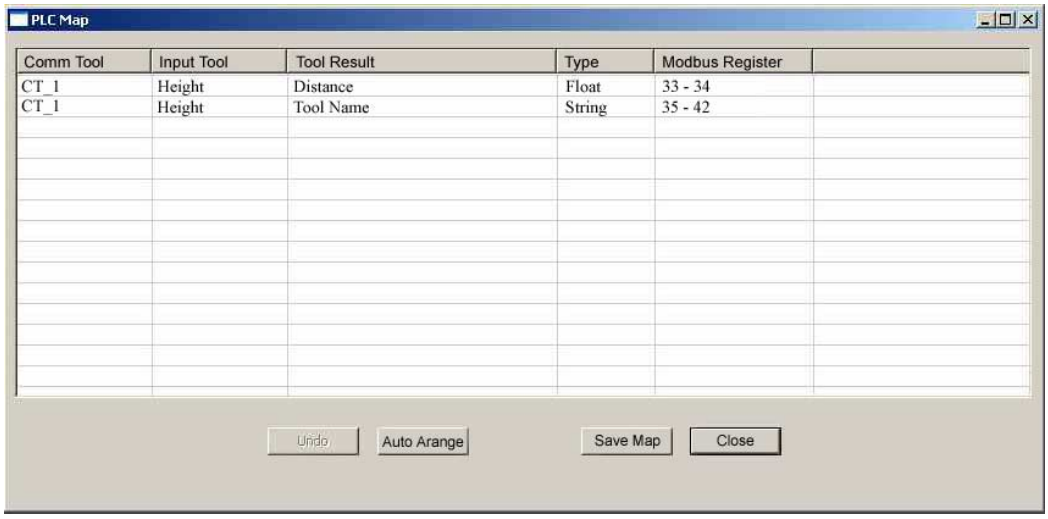
1. Connection(s) is set to Industrial Ethernet
2. Select the measure tool (Height)
3. Select Distance and Tool Name
4. View the Register number




Notice that the Register number for the distance measurement is 33 – 34 (30033). That number is the default number but it can be changed by click on the up/down arrows under Register.

Note: Click on the PLC Map to view all the items selected in the communications tool.



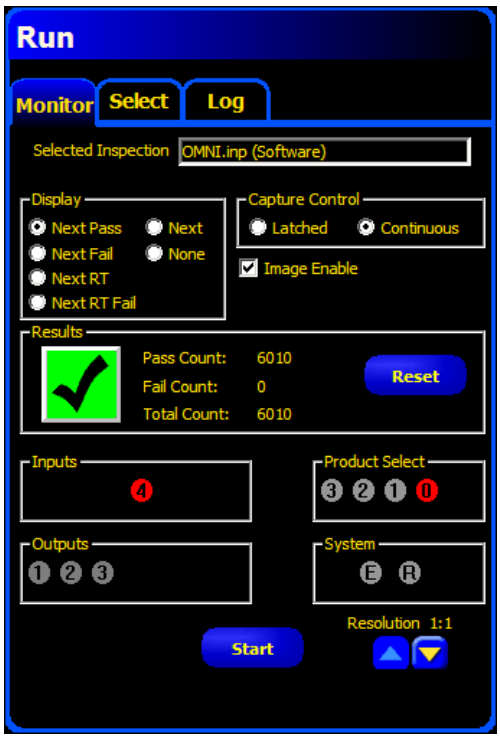

The PLC map will display the tool name, the results that will be exported, the data type, and the Modbus Register. This map can be saved as a text file on your PC. Then save the inspection.

**Activating image export**

Finally, the last step before starting the camera is to check Image Enable under the Monitor tab so the G3 can display it in the primitive.
The image to be displayed in that primitive is defined under the display window. Possible choices are available such as Next Pass for a continuous display or Next Fail to get only failed images or any other setting relevant to your application.

Click Start to start the camera.

At this point, you can minimize the vision sensor software, or close it out entirely.




This concludes the configuration of the camera. Connect the PresencePLUS P4 OMNI to the HMI with a crossover Ethernet cable.

## Appendix - Modbus/TCP Register Summary

**Modbus Register Summary**

**Coils**

**Input Flag (Input coils 00001-00016)**

| Register | Bit Position | WORD NAME |
|---|---|---|
| 00001 | 0 | Trigger |
| 00002 | 1 | Remote Teach |
| 00003 | 2 | Product Change |
| 00004 | 3 | reserved |
| 00005 | 4 | reserved |
| 00006 | 5 | reserved |
| 00007 | 6 | reserved |
| 00008 | 7 | reserved |
| 00009 | 8 | BCR String Change |
| 00010 | 9 | reserved |
| 00011 | 10 | reserved |
| 00012 | 11 | reserved |
| 00013 | 12 | reserved |
| 00014 | 13 | reserved |
| 00015 | 14 | reserved |
| 00016 | 15 | reserved |

**Output Flags (Output Coil 10001-10016)**

| Register | Bit Position | WORD NAME |
|---|---|---|
| 10001 | 0 | Pass |
| 10002 | 1 | Fail |
| 10003 | 2 | Error |
| 10004 | 3 | reserved |
| 10005 | 4 | Ready |
| 10006 | 5 | reserved |
| 10007 | 6 | reserved |
| 10008 | 7 | reserved |
| 10009 | 8 | I/O 1 |
| 10010 | 9 | I/O 2 |
| 10011 | 10 | I/O 3 |
| 10012 | 11 | I/O 4 |
| 10013 | 12 | I/O 5 |
| 10014 | 13 | I/O 6 |
| 10015 | 14 | reserved |
| 10016 | 15 | reserved |

**ACK Flag (Output coils 10017-10032)**

| Register | Bit Position | WORD NAME |
|---|---|---|
| 10017 | 0 | Trigger Ack |
| 10018 | 1 | Remote Teach Ack |
| 10019 | 2 | Product Change Ack |
| 10020 | 3 | reserved |
| 10021 | 4 | reserved |
| 10022 | 5 | reserved |
| 10023 | 6 | reserved |
| 10024 | 7 | reserved |
| 10025 | 8 | BCR Str Change ACK |
| 10026 | 9 | reserved |
| 10027 | 10 | reserved |
| 10028 | 11 | reserved |
| 10029 | 12 | reserved |
| 10030 | 13 | reserved |
| 10031 | 14 | reserved |
| 10032 | 15 | reserved |

**Words**

**Output Registers 30001-30960**

| Register | WORD NAME |
|---|---|
| 30001 | Output Flags (coils 10001-16) |
| 30002 | ACK Flags (coils 10017-32) |
| 30003 | Inspection Number |
| 30004 | System Error Count |
| 30005 - 6 | Frame Count |
| 30007 - 8 | Pass Count |
| 30009 - 10 | Fail Count |
| 30011 - 12 | Missed Triggers |
| 30013 - 14 | Iteration Count |
| 30015 - 32 | *reserved* |
| 30033 | Location 1 |
| 30034 | |
| 30035 | Location 2 |
| 30036 | |
| 30037 | Location 3 |
| 30038 | |
| ... | |
| 30957 | Location 463 |
| 30958 | |
| 30959 | Location 464 |
| 30960 | |

**Input Registers 40001-41000**

| Register | WORD NAME |
|---|---|
| 40001 | Input Flags (Coils 00001-16) |
| 40002 | Product Select |
| 40003 - 39 | *reserved* |
| 40040 - 139 | BCR Compare String |
| 40140 - 239 | BCR Compare Mask |
| 40240 - 1000 | *reserved* |

**Output Registers 41001-42000 (same as 30000 Reg)**

| | |
|---|---|
| 41001 | Output Flags |
| 41002 | ACK Flags |
| 41003 | Inspection Number |
| 41004 | System Error Count |
| 41005 - 1006 | Frame Count |
| 41007 - 1008 | Pass Count |
| 41009 - 1010 | Fail Count |
| 41011 - 1012 | Missed Triggers |
| 41013 - 1014 | Iteration Count |
| 41015 - 1032 | *reserved* |
| 41033 | Location 1 |
| 41034 | |
| 41035 | Location 2 |
| 41036 | |
| 41037 | Location 3 |
| 41038 | |
| ... | |
| 41957 | Location 463 |
| 41958 | |
| 41959 | Location 464 |
| 41960 | |
| 41961 - 2000 | *reserved* |

Output, ACK and input flags

**Output Coils (10001-10032)**

| | Bit Position | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output Flags | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Output Coil # | Coil 16 | Coil 15 | Coil 14 | Coil 13 | Coil 12 | Coil 11 | Coil 10 | Coil 9 | Coil 8 | Coil 7 | Coil 6 | Coil 5 | Coil 4 | Coil 3 | Coil 2 | Coil 1 |
| Name | *reserved* | *reserved* | I/O 6 | I/O 5 | I/O 4 | I/O 3 | I/O 2 | I/O 1 | *reserved* | *reserved* | *reserved* | Ready | *reserved* | Error | Fail | Pass |

| | Bit Position | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACK Flags | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Output Coil # | Coil 32 | Coil 31 | Coil 30 | Coil 29 | Coil 28 | Coil 27 | Coil 26 | Coil 25 | Coil 24 | Coil 23 | Coil 22 | Coil 21 | Coil 20 | Coil 19 | Coil 18 | Coil 17 |
| Name | *reserved* | . | . | . | . | . | . | . | . | . | . | . | *reserved* | Product Change Ack | Remote Teach Ack | Trigger Ack |

**Input Coils (00000-00015)**

| | Bit Position | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input Reg 1 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Input Coil # | | | | | | | | Coil 9 | | | | | | Coil 3 | Coil 2 | Coil 1 |
| Name | *reserved* | *reserved* | *reserved* | *reserved* | *reserved* | *reserved* | *reserved* | *BCR String Change* | *reserved* | *reserved* | *reserved* | *reserved* | *reserved* | Product Change | Remote Teach | Trigger |