PERFECTION IN AUTOMATION
A MEMBER OF THE ABB GROUP

# Vision Demo Application

Version 2.0
Date: 06/03/2020

## Contents

# 1 Introduction

This sample application can be used as a starting point for vision applications. Google Chrome browser must be used for the mappView project; other browser may not show the correct crosshair position.

## 1.1 System requirements

This sample was developed and tested with Automation Studio 4.7

- PLC OS system C.72 or higher
- mappView 5.9
- Hardware files for camera (VSS112Q22.081P-E01 was used in this sample)
- Chrome Browser

# 2 Project Files

The following project files are vision related.

### 2.1 Logical View

All tasks starting with Vi_ should not be changed to make updating easier.

| | |
|---|---|
| Vision | Package with vision tasks |
| Vi_main | This task handles functions that are sensor related |
| Vi_light | This task handles functions that are light related |
| Vi_nettime | This task handles the nettime calculation |
| Vi_image | This task handles the image archive |
| YourTask | Customer specific task |
| setRouteToCamera | Make sure to adjust the IP address in the file “\Vision_1\Logical\Vision\setRouteToCamera.bat” and execute the batch file in Windows with right click (Run as administrator). Otherwise, the sensor image does not work in the Vision Cockpit or the demo visualization. |
| mappView | mappView visualization for vision |
| mappRecipe | Stores the camera configuration |

### 2.2 Configuration View

| | |
|---|---|
| mappView | mappView visualization for vision |
| mappVision | mappVision configuration for vision functions |
| mappService | Configuration for recipe managment |

### 2.3 Physical View

| | |
|---|---|
| Blob | Sensor for the blob function. Powerlink Node 1. |
| Measurement | Sensor for the edge measurement function. Powerlink Node 2. |
| CodeRead | Sensor for the code reader function. Powerlink Node 3. |
| Match | Sensor for the match function. Powerlink Node 4. |
| OCR | Sensor for the text recognition function. Powerlink Node 5. |

# 3 Parameter structures

The sample supports multiple cameras but only one is displayed at a time. The global structures begin with a “g” for example gVisionSensor, gBlob, … The global structures are arrays where the index represents the Powerlink node number. The variable “SelectedSensor” maps one of the global structures to the dynamic local variable.

## 3.1 Vision sensor structure

The vision structure handles all functions and parameters that are sensor related.

| | | |
|---|---|---|
| CMD | | Command structure to trigger an action |
| | ImageTrigger | Start a new image aquisition |
| | ImageTriggerReset | Abort future image acquisition (with TriggerDelay) |
| | AutoSetupStartStop | Start and stop automatic camera setup |
| | AutoSetupTransfer | Transfer parameters generated with auto setup |
| | BrowserReload | Reload browser widget image in mappView visualization |
| CFG | | Sensor configuration, see manual for details |
| STA | | Sensor status information, see manual for details |
| HW | | Sensor hardware information, see manual for details |

## 3.2 Functional structure

Each vision function has its own structure containing the following sub structures.

| | |
|---|---|
| CFG | Function related configuration, see manual for details |
| DATA | Vision data, see manual for details |

## 3.3 Vision image structure

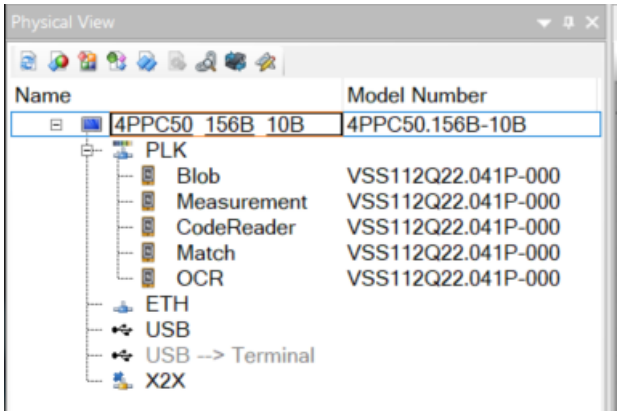
The image structure handles all functions and parameters that are related to the image archive.

| | | |
|---|---|---|
| CMD | | Command structure to trigger an action |
| | Upload | Upload an image from the camera and store it on the flash card |
| | Refresh | Reload image list from flash card |
| | Delete | Delete selected image |
| | ResetError | Acknowledge |
| | DeleteDir | Deletes the complete folder with all images |
| | CreateDir | Creates an empty folder for images |
| CFG | | Image Archiv configuration |
| | FileDevice | File device name where the images are stored |
| | DirName | Name of an automatically created folder on the FileDevice |
| | CameraIP | IP address of the sensor |
| | ConvertCycles | For saving the image with crosshair the imagedate needs to be converted with Base64. This is the number of converted bytes per TC8 cycle. So e.g. an 1.3MP bmp file has ca. 1.300.000 Bytes to convert. The defaultvalue of 10.000 needs 130 TC8 cycles. On high performance CPUs this value could be increased, maybe much more (> 6.000.000 makes all in one cycle, also with 5MP bmps). On low performace CPUs maybe this value needs to be decreased. |
| | Format | Image format (bmp (1) or jpg (0)) |
| | QualityJPG | For JPEG a quality can be defined |
| | UploadBmpJpg | If TRUe, the bmp/jpg images will be loaded from the camera |
| | UploadSVG | If TRUE, the bmp/jpg will be converted to SVG and the crosshairs with data will be embedded in the new SVG file |
| STA | | Status information |
| | Status | Status of the image operation |
| DATA | | Sensor status information, see manual for details |
| | Images | Images list as data provider for connection to mappView |
| | Croshair | Crosshairdata, will be copied from VisionMain |

# 4 Description

## 4.1 Hardware configuration

The sensor used in this sample is VSS112Q22.081P-E01. If this is not the sensor available right click on the hardware and choose “Replace Hardware Module” to select the correct hardware.




Each node number represents one vision function. So by changing the node number it is possible to quickly switch beteen different functions.

The hardware configuration uses the following Powerlink node numbers:

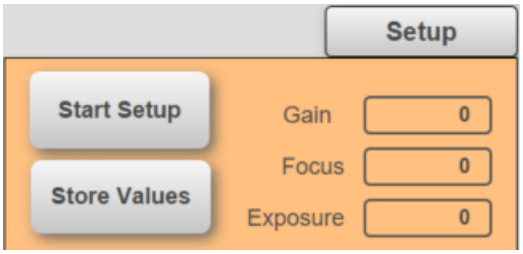
1: Blob
2: Measurement
3: Code Reader
4: Match
5: OCR

Make sure to adjust the IP address in the file “\ProjectName\Logical\Vision\setRouteToCamera.bat” and execute the batch file in Windows with right click (Run as administrator). Otherwise, the sensor image does not work in the Vision Cockpit or the demo visualization.

## 4.2 Demo application

The demo application consists of one page for each vision function. The main page is used to set up the sensor image. The bottom window shows the most important parameters and status information. The first step is to make sure that the sensor is connected and ready. All four elements at the top should be green.

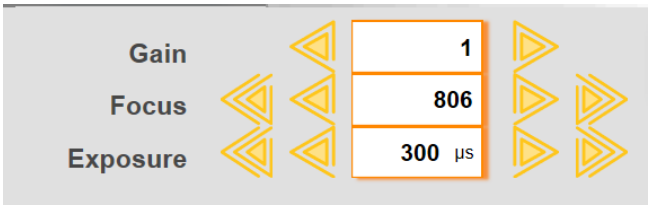

To start the auto setup process click on Setup on the bottom right corner. Click on “Start Setup” to initiate the auto setup that determines the values for gain, focus and exposure.




The sensor light should flash for about 20 seconds. When the proecess is finished click “Store Values” to transfer the automatic generated values to the active configuration or click “Stop Setup” to discard the generated values.

If the object is not aligned correctly, the Repetitive Mode can be activated. Now cyclic pictures are taken and the objebt or the camera can be aligned.

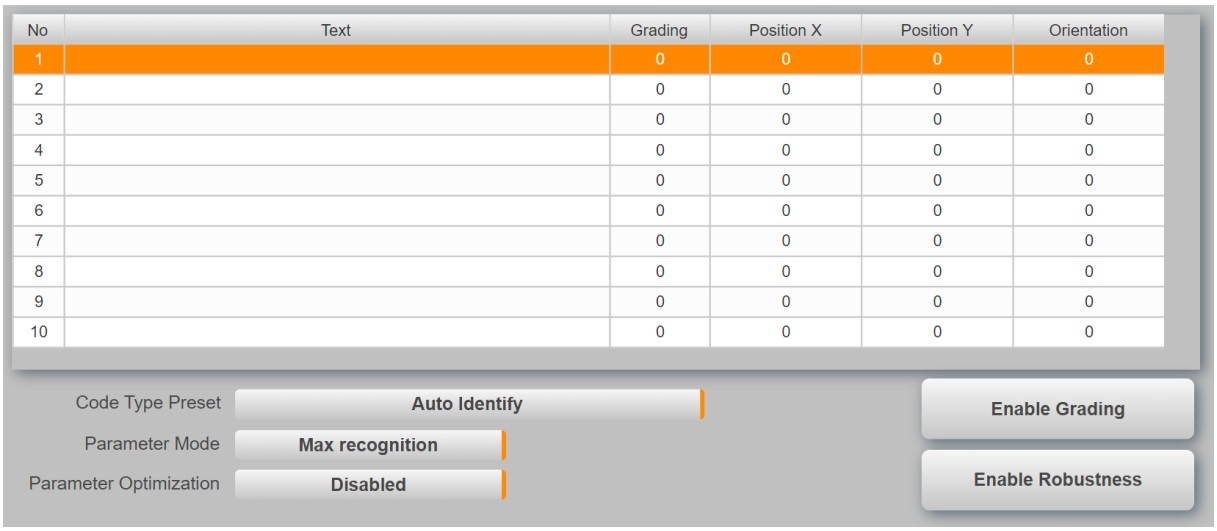
Click on Trigger to generate a new image. In some cases it may be necessary to adjust the automatically generated values. Use the numeric fields or arrows to optimize the image.




Use the crosshair toggle button to show additional information. Images are stored automatically when the checkbox “Auto Archiv Image” is set (see 0). Google Chrome browser must be used for the mappView project; other browser may not show the correct crosshair position.

## 4.3 Code Reader

The code reader page provides the information that are specific for the code reader functions. Select the code type from the drop down menu or use “Auto Identify” to start the process that tries to identify the code automatically. The identification process can run for up to 20s.



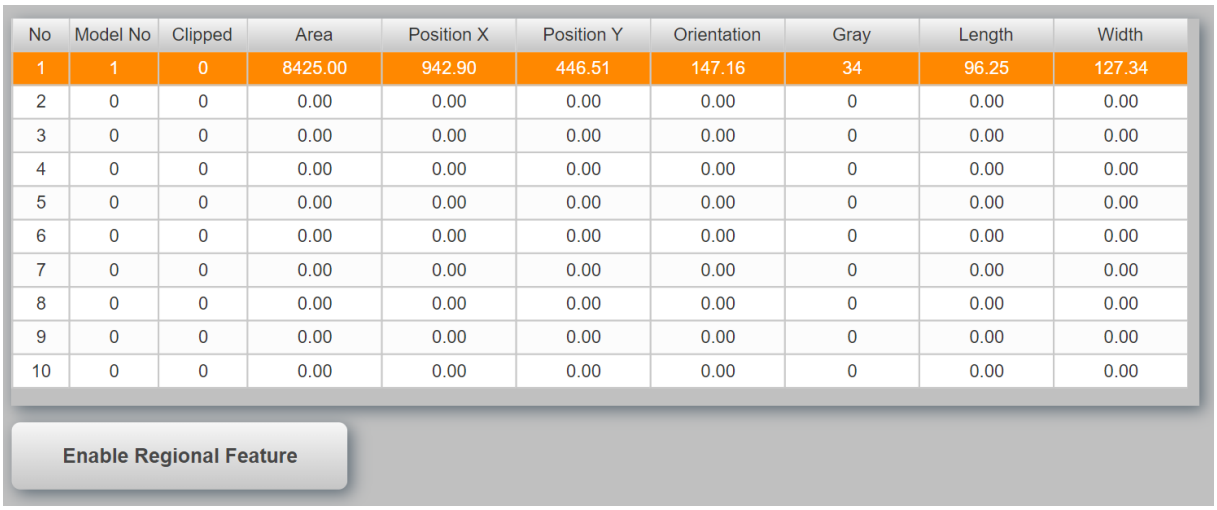
| No | Text | Grading | Position X | Position Y | Orientation |
|---|---|---|---|---|---|
| 1 | | 0 | 0 | 0 | 0 |
| 2 | | 0 | 0 | 0 | 0 |
| 3 | | 0 | 0 | 0 | 0 |
| 4 | | 0 | 0 | 0 | 0 |
| 5 | | 0 | 0 | 0 | 0 |
| 6 | | 0 | 0 | 0 | 0 |
| 7 | | 0 | 0 | 0 | 0 |
| 8 | | 0 | 0 | 0 | 0 |
| 9 | | 0 | 0 | 0 | 0 |
| 10 | | 0 | 0 | 0 | 0 |



The max item count value is set to 1 on Code Reader page.

## 4.4 Blob

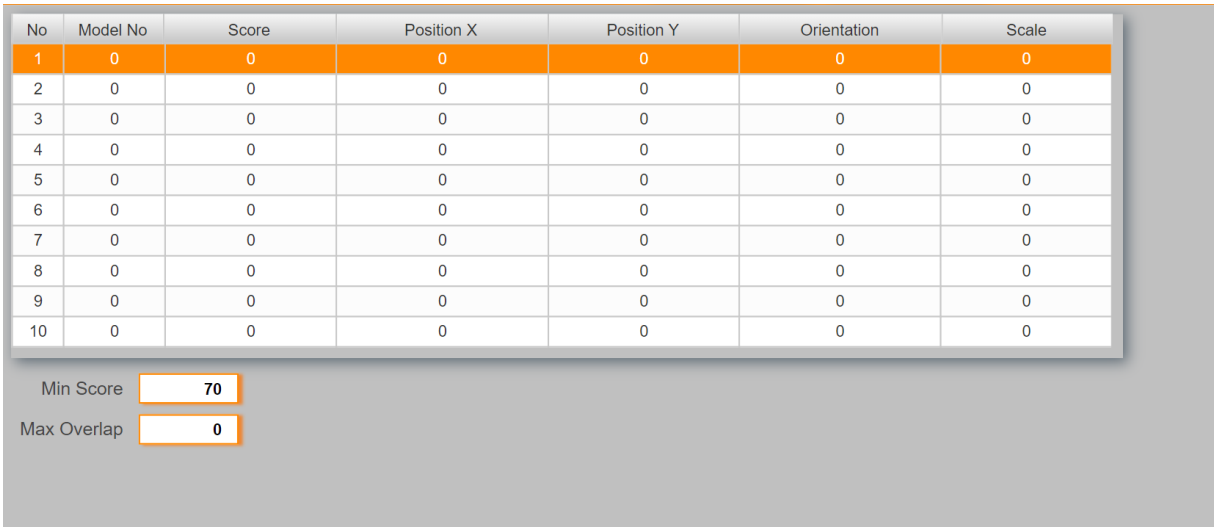
The blob page provides the information that are specific for the blob functions. The table shows the details for each blob that was detected by the sensor. Teaching must be done in the Vision Cockpit.



| No | Model No | Clipped | Area | Position X | Position Y | Orientation | Gray | Length | Width |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 8425.00 | 942.90 | 446.51 | 147.16 | 34 | 96.25 | 127.34 |
| 2 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 3 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 4 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 5 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 6 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 7 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 8 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 9 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |
| 10 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 | 0.00 |

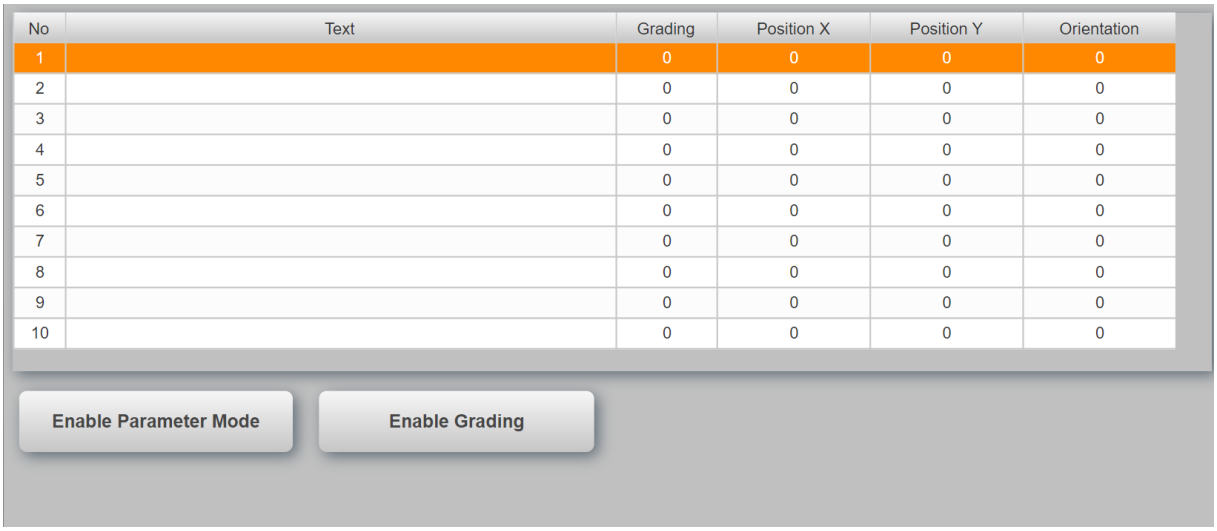
## 4.5 Match

The match page provides the information that are specific for the match functions. The table shows the details for each item that was detected by the sensor. Teaching must be done in the Vision Cockpit.



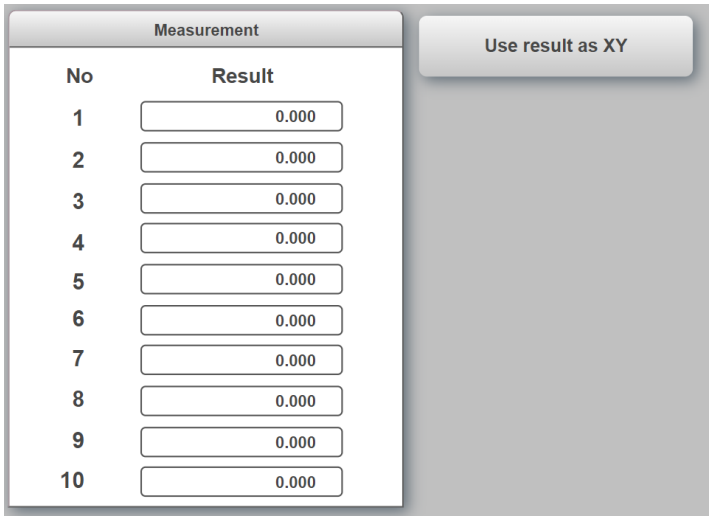

## 4.6 OCR

The OCR page provides the information that are specific for the OCR functions. The table shows the details for each text that was detected by the sensor.



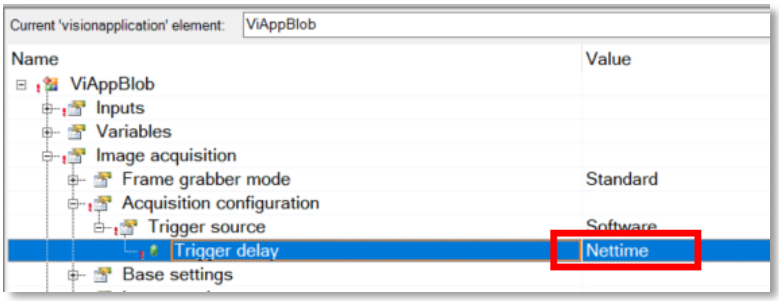
## 4.7 Measurement

The measurement page provides the information that are specific for the edge measurement functions. This page shows the results for the different measurement functions. What is measured must be configured in the vision cockpit. For edge detection in can be helpful to also draw crosshairs at the position where the edge was found. This can be enabled with the toggle button “Use result as XY”. In this case the first result must be defined as the X position and the second as the Y position. Repeat this pattern for all edges.



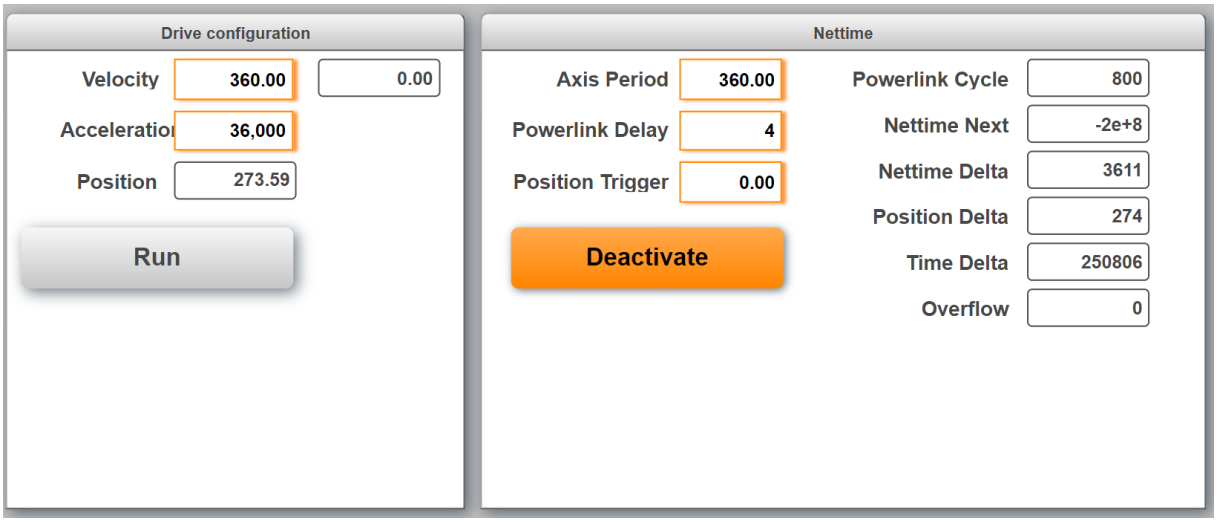
## 4.8 Using Nettime

In some applications it can be necessary to trigger the camera periodicly depending on a drive position. This can be accomplished with a high precision through nettime. To enable nettime change the trigger delay to nettime in the vision application settings.




When this feature is enabled the manual trigger gVisionSensor[].CMD.ImageTrigger only works when the parameter gVisionSensor[].CFG.NettimeDelay is set correct (Current nettime value plus offset, ex. 10ms).

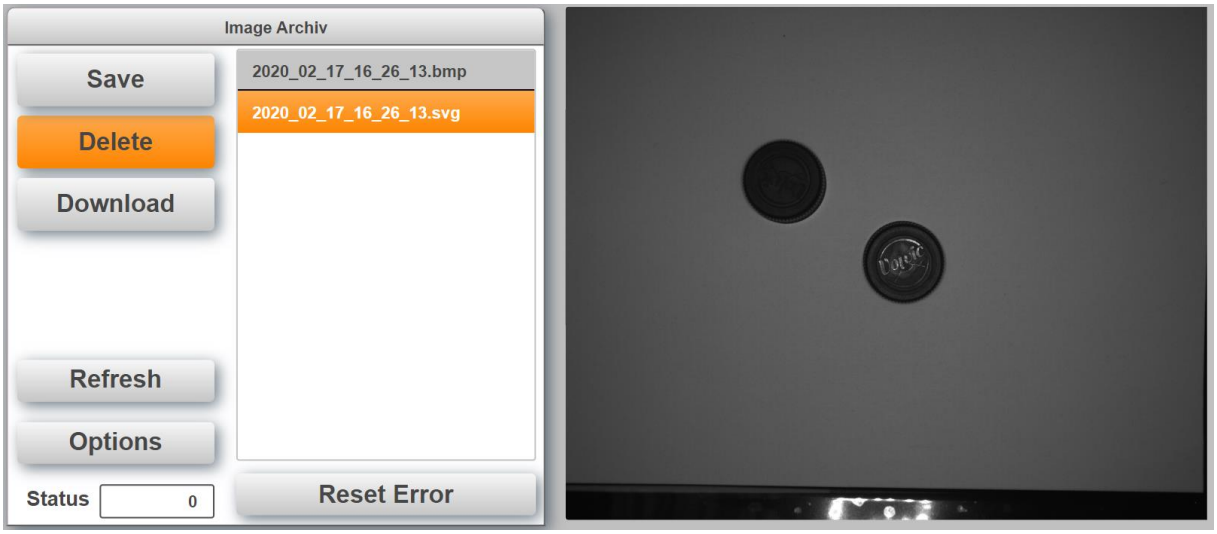
The task Vi_nettime provides the necessary calculation. It is curcial that this task runs in sync and at the same cycle as the Powerlink bus. The following page allows the configuration of the nettime function. On the left hand side are the basic drive settings.



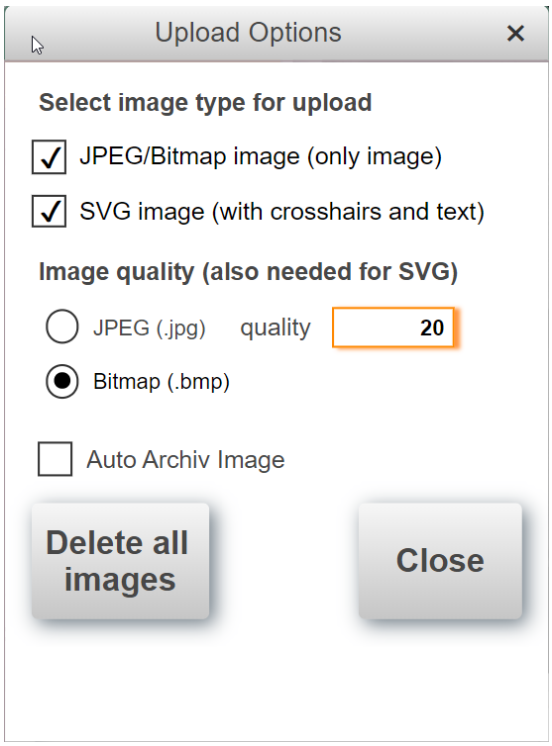

On the right hand side are the nettime settings. The axis period is number of units for one cycle (360 for a rotating axis). Powerlink delay is used to calculate the time when the nettime value must be set at the latest to make it to the camera in time. Position trigger is the moment when the trigger is fired within the period (0-360).

## 4.9 Image Archive

The image archive is used to store camera images on the PLC flash card. This can be necessary to inspect ‘bad’ products later in the process. The image archive is controlled by its own task “VisionImage” and structure (see 3.3).




The number of images that are stored depends on the size of array *VisionImage.DATA.Images*. The default size is 20. When the list is full and a new images are uploaded the oldest images will automatically be deleted. The task will also highlight and load the newest image after upload or refresh. Images are stored automatically when the checkbox “Auto Archiv Image” is set on the main page.




In the Options Dialog it can be selected, if the camera creates a BMP or JPEG image. For JPEG images the quality can be selected. Also 100% is possible. It can be selected if the BMP or JPG will be saved as it is and/or if a SVG with crosshairs will be created. All Options are possible, so only SVG Upload is possible or also both or only BMP/JPEG. “Reset” resets e.g. FileIO Erros, you can find in the “Status” information on image archive. “Delete all images” deletes the complete folder with all images and creates the new empty folder.

The PLC has the FTP server enabled to check the images remotely. The user name and password is “bundr”.

# 5 Tips and Hints

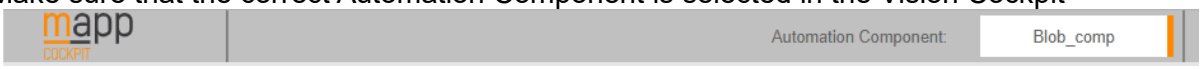
## 5.1 Sensor is connected and ready but the image on the main page is not refreshed

Make sure to adjust the IP address in the file "\ProjectName\Logical\Vision\setRouteToCamera.bat" and execute the batch file in Windows with right click (Run as administrator).

## 5.2 The Vision Cockpit does not work correct and/or does not show the camera image when the camera is connected and ready.

Make sure to adjust the IP address in the file "\Vision_1\Logical\Vision\setRouteToCamera.bat" and execute the batch file in Windows with right click (Run as administrator).

Make sure that the correct Automation Component is selected in the Vision Cockpit




## 5.3 How to setup a T50 to use demo?

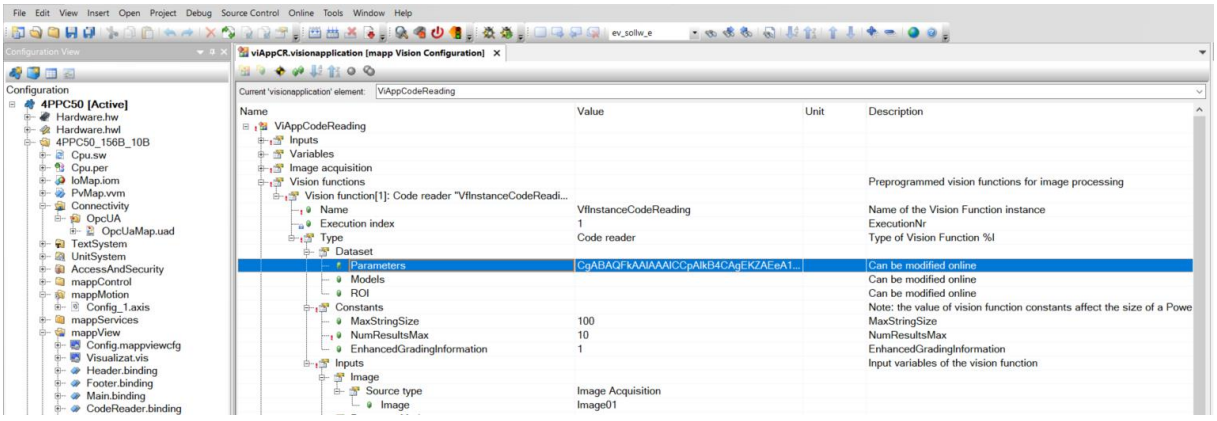
Assuming that the PLC has the IP address: 192.168.1.100. Go into the T50 and change the following settings

**Web:**
http://192.168.1.100:81/index.html?visuId=visVision

**Network:**
IP address: 192.168.1.98
Subnet mask: 255.255.255.0
Gateway: 192.168.1.100

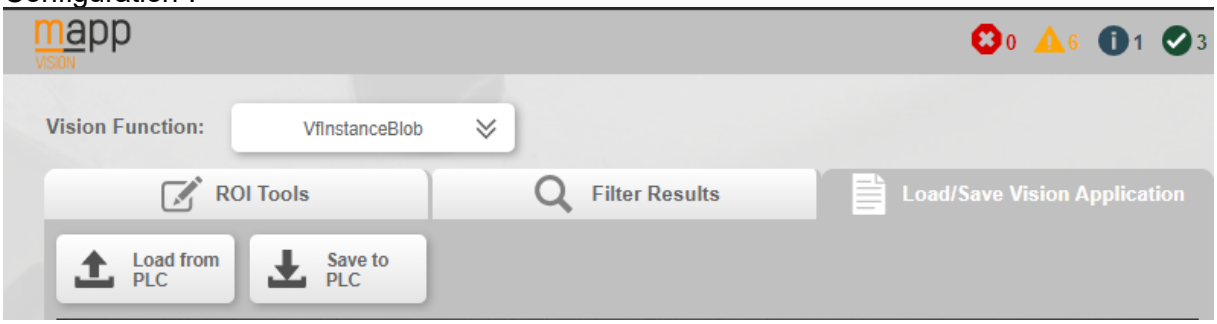
## 5.4 How is the camera configuration selected?

The default configuration is defined in the Automation Studio project under mappVision->…visionapplication.

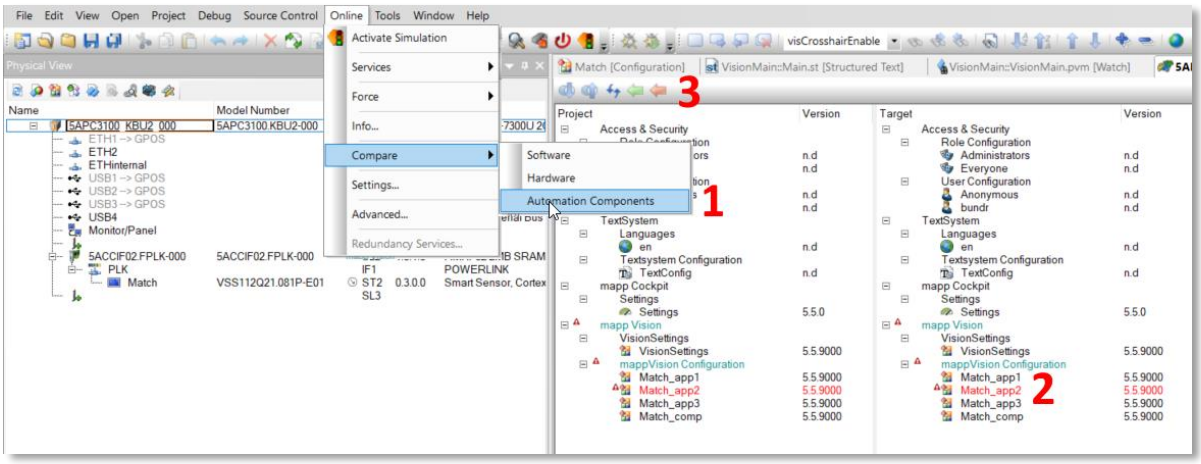


## 5.5 How to store a configuration taught in the vision Cockpit in the Automation Studio project.

Teach the configuration in the vision Cockpit and use the button “Save Vision Function Configuration”.




Go back into Automation Studio and select Online->Compare->Automation Components (1).




Select the vision application highlighted in red (2). Select orange arrow (3) at the top to transfer the camera configuration back to Automation Studio.

# 6 Revision History

➔ You can find the revision history also in the project (folder “Vision”/revision.txt)

Version 2.0

First public release