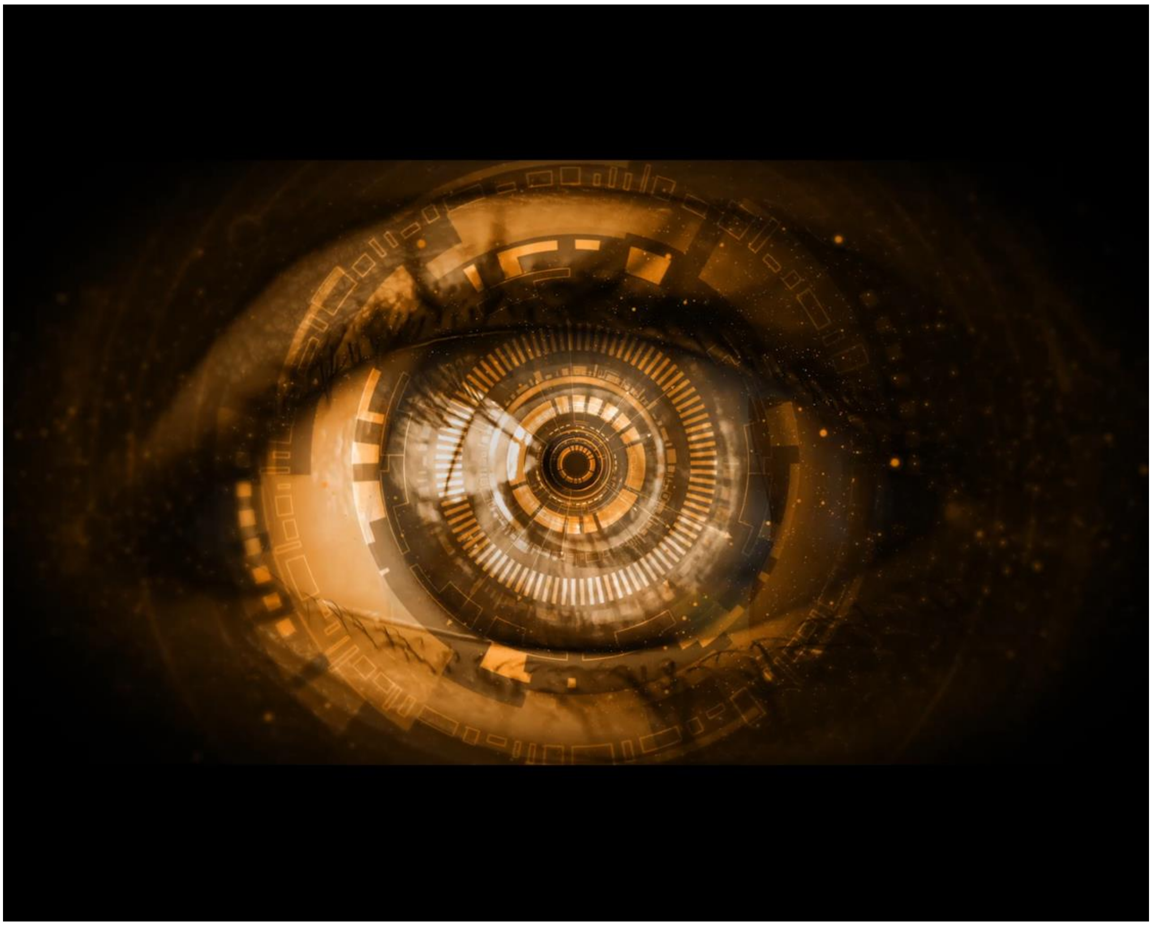
Mapp Vision

Demo Application Manual



Date: May 26, 2023

I Versions

Version	Date	Comment	Edited by
1.0	May 26, 2023	First Edition	Jacob Buhl

Table 1: Versions

II Distribution

Name	Company, Department	Amount	Remarks

Table 2: Distribution

III Safety Notices

Safety notices in this document are organized as follows:

Safety notice	Description
Danger!	Disregarding the safety regulations and guidelines can be life-threatening.
Warning!	Disregarding the safety regulations and guidelines can result in severe injury or heavy damage to material.
Caution!	Disregarding the safety regulations and guidelines can result in injury or damage to material.
Information:	Important information used to prevent errors.

Table 3: Safety notices

IV Table of Contents

2 Introduction

This sample application can be used as a starting point for vision applications. Google Chrome browser must be used for the map view project; another browser may not show the HMI correctly.

1.1 System requirements

This sample was developed and tested with Automation Studio 4.12

- PLC OS system B4.93 or higher
- mappView 5.22
- Chrome Browser

Hardware files in use

- VSS112Q12.023P-000 1.7.0.605 VSS, 800MHz 2C, 1.3MP LS, 4.6mm, RGBL
- VSLB11Q00.54DP-000 1.1.0.323 BL 1x1, IP54, PLK, RGBL, Diff
- VSLL11R30.67AP-000 1.1.0.494 LB Std, PLK, RBIRW, 23°, clear

3 Project Files

The following project files are vision sensor related.

3.1 Logical View

All programs are located it own package with the same name under the "programs" package in Logical view

Axis	Task with axis related code
Files	Task with file system related code
Vision	Task with handles functions that are sensor related
Recipe	Store/load/create etc. the different tasks recipe data structures camera

3.2 Configuration View

mappView	mappView visualization for vision
mappVision	mappVision configuration for vision functions
mappService	Configuration for recipe management

3.3 Physical View

Blob	Sensor for the blob function. Powerlink Node 1.
CodeRead	Sensor for the code reader function. Powerlink Node 2.
Match	Sensor for the match function. Powerlink Node 3.
Measurement	Sensor for the edge measurement function. Powerlink Node 4.
OCR	Sensor for the text recognition function. Powerlink Node 5.
PixelCounter	Sensor for the text pixel counter function. Powerlink Node 6.

This is an example configuration with one camera for each vision function. You can quickly switch between the different vision functions by changing the node number.

It also made possible to add up to 9 lights for each camera. In the demo project 3 lights have been added one backlight and 2 bar lights. The 16x node switch on the light is used to choose which camera it is connected to so for camera 1 its set to 1 and camera 2 to 2. The 1x node switch is used for addressing the individual lights for the camera 0-8.

3.4 Constants

The project provides several constants to adjust the configuration, which are located in programs

Vision:
Programs\Vision\Vision \variables.var

Name	Default	Description
MAX_NUM_CAMS	6	Maximum number of camera's
MAX_NUM_LIGHTS	9	Maximum number of light's
MAX_NUM_RESULTS	10	Maximum number of results
START_IDX	1	Start index of arrays
MAX_NUM_RESULTS_M1	MAX_NUM_RESULTS-1	Max number of results -1

Files:
Programs\Files\Files\variables.var

Name	Default	Description
FILES_LIST_SIZE	13	Number of files to be displayed on one page of the list in the HMI application

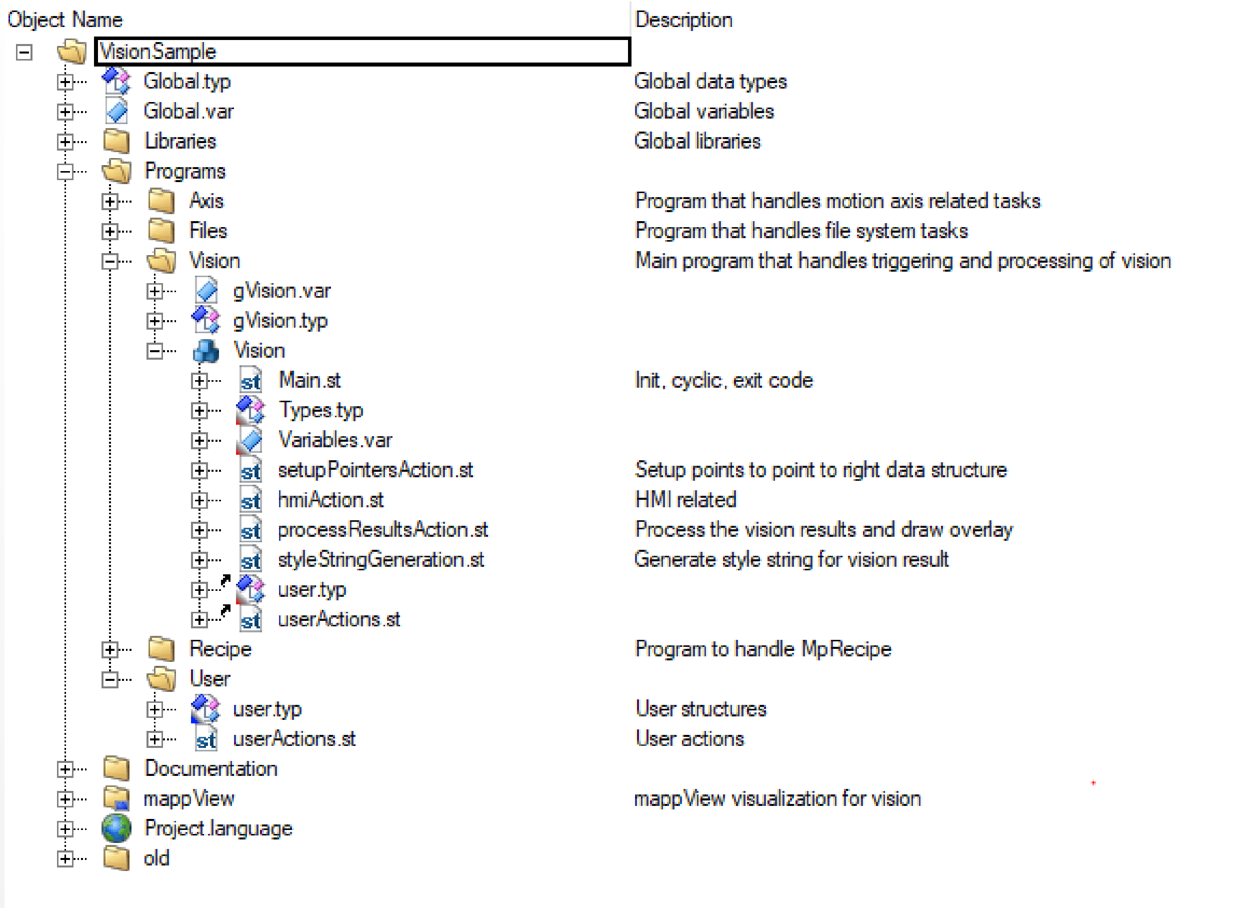
3.5 Parameter structures

The sample supports multiple cameras, by using arrays in the Vision program. Each array instance of hwInstance, hmiInstance and localInstance correspond to a camera. The program the uses pointer hw, hmi, this to access the right instance for a given camera. All cameras are always process as the Vision program loop through all instances.

The parameter structures are used for the other program, with the exception of using arrays and points as they only contain a single instance but the names hw, hmi and this are used for the single instance.

3.5.1 User

To simplify modification to project two user files are defined, a type file and an action file.
These are defined under the user package under programs. These two files are then referenced into the Vision. The actions are called at there respective places and instances of the data structure type are created.



3.5.2 HMI

The hmi data structure contains and in and out structure. The in is writable on OPC UA/ HMI where the output is read only. It do also contain a user instance which has its own in and out and is defined in the type file under the user package.

HMI	
hmi_typ	
in	hmi_in_typ
out	hmi_out_typ
user	user_hmi_typ

The hmi in structure contains an instance of the recipe data structure such that its directly accessible from the HMI, this data structure is saved to file on request from the recipe page. It also contains a cmd structure that are used for the different hmi commands such as button clicks etc.

hmi_in_typ			
cmd	hmi_in_cmd_typ		☐
recipe	recipe_typ		☐
cameraInfo	brdkViBase_camInfo_typ		☐
diagnosticFilename	STRING[50]		☐
vaName	STRING[50]		☐
lightNum	USINT		☐

3.5.3 HW

The hardware data structure contain all the variables that are connected though IO map to the actual hardware. Each of these data structures contains an in and out structure to indicate the direction. In is input to the PLC and out is output from the PLC.

It has structures for both the camera hardware, all the vision functions, all the lights, IO's for io triggering and the user defined IO's.

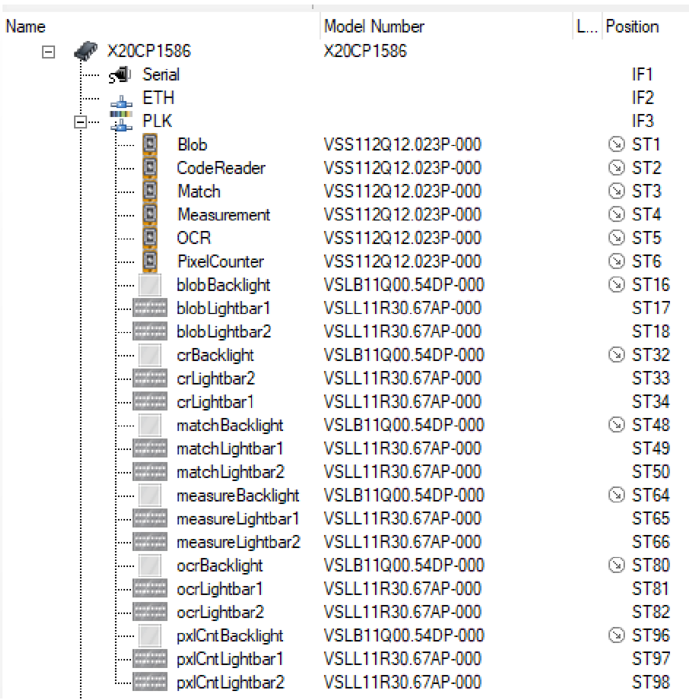
hw_typ	
camera	brdkViBase_cam_hw_typ
VF	hw_VF_typ
lights	brdkViBase_light_hw_typ[START_IDX..MAX_NUM_LIGHTS]
io	hw_io_typ
user	user_hw_typ

4 MappVision demo description

4.1 Hardware configuration

The sensor used in this sample is VSS112Q12.023P-000. If this is not the sensor available right click on the hardware and choose “Replace Hardware Module” to select the correct hardware.

In the demo application, each sensor represents one vision function. By changing the node number, it is possible to quickly switch between different functions.



Name	Model Number	L...	Position
X20CP1586	X20CP1586		
Serial			IF1
ETH			IF2
PLK			IF3
Blob	VSS112Q12.023P-000		ST1
CodeReader	VSS112Q12.023P-000		ST2
Match	VSS112Q12.023P-000		ST3
Measurement	VSS112Q12.023P-000		ST4
OCR	VSS112Q12.023P-000		ST5
PixelCounter	VSS112Q12.023P-000		ST6
blobBacklight	VSLB11Q00.54DP-000		ST16
blobLightbar1	VSLL11R30.67AP-000		ST17
blobLightbar2	VSLL11R30.67AP-000		ST18
crBacklight	VSLB11Q00.54DP-000		ST32
crLightbar2	VSLL11R30.67AP-000		ST33
crLightbar1	VSLL11R30.67AP-000		ST34
matchBacklight	VSLB11Q00.54DP-000		ST48
matchLightbar1	VSLL11R30.67AP-000		ST49
matchLightbar2	VSLL11R30.67AP-000		ST50
measureBacklight	VSLB11Q00.54DP-000		ST64
measureLightbar1	VSLL11R30.67AP-000		ST65
measureLightbar2	VSLL11R30.67AP-000		ST66
ocrBacklight	VSLB11Q00.54DP-000		ST80
ocrLightbar1	VSLL11R30.67AP-000		ST81
ocrLightbar2	VSLL11R30.67AP-000		ST82
pxlCntBacklight	VSLB11Q00.54DP-000		ST96
pxlCntLightbar1	VSLL11R30.67AP-000		ST97
pxlCntLightbar2	VSLL11R30.67AP-000		ST98

The hardware configuration uses the following Powerlink node numbers:

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

It also made possible to add up to 9 lights for each camera. In the demo project 3 lights have been added one backlight and 2 bar lights. The 16x node switch on the light is used to choose which camera it is connected to so for camera 1 its set to 1 and camera 2 to 2. The 1x node switch is used for addressing the individual lights for the camera 0-8.

4.2 Camera image

The mappView visualization shows the image of the camera on the main page. This is done through the brdkVisionImage widget that both support the image and showing an overlay. No port forwarding or routing is needed as the mappView function blocks in Vision handles the communication with the widget.

4.3 Demo application

The demo application consists of a header, footer and navigation area from where its possible to navigate to all the page, see Figure 1. In the following section each of these areas are described along with all the pages.

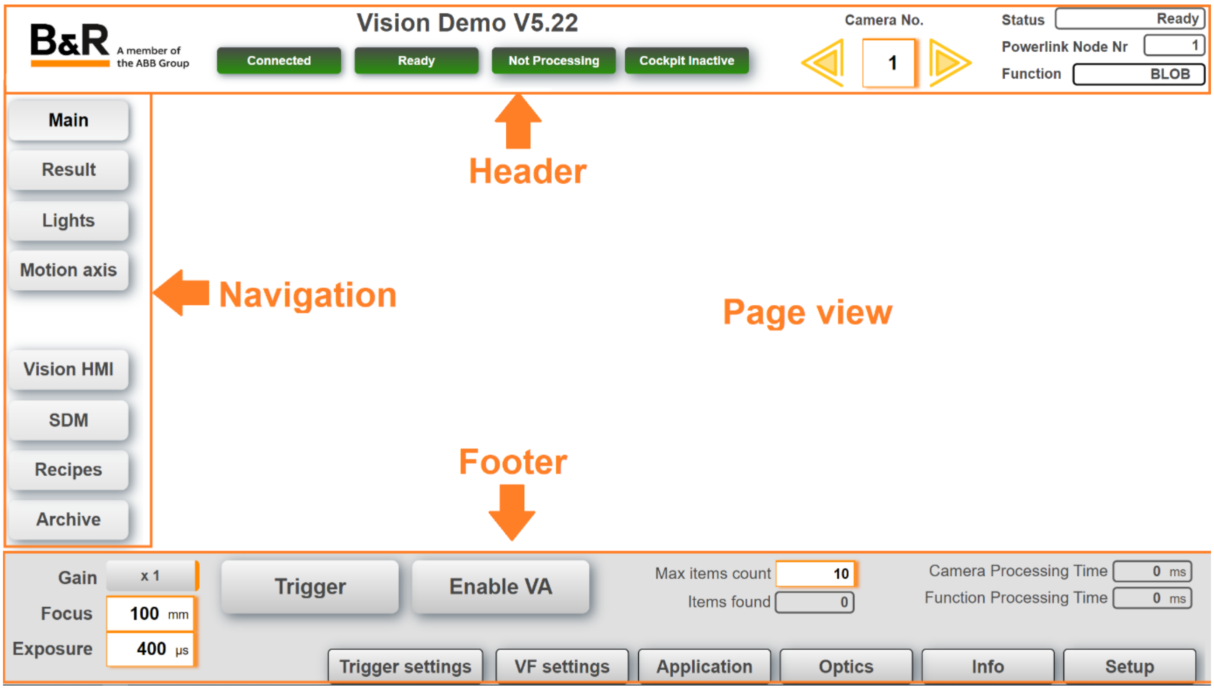


Figure 1 MappVision demo layout

4.3.1 Header

The header contains the B&R logo along with name of this application and its version number. To change between different camera the camera number can be changed directly or by the toggle up and down button.

It also contains four status indicators for the selected camera number, if running in simulation a yellow indicator is shown with simulation instead of the four status indicators. On the right side the current camera status, Powerlink node number and the vision function description is shown.

4.3.2 Footer

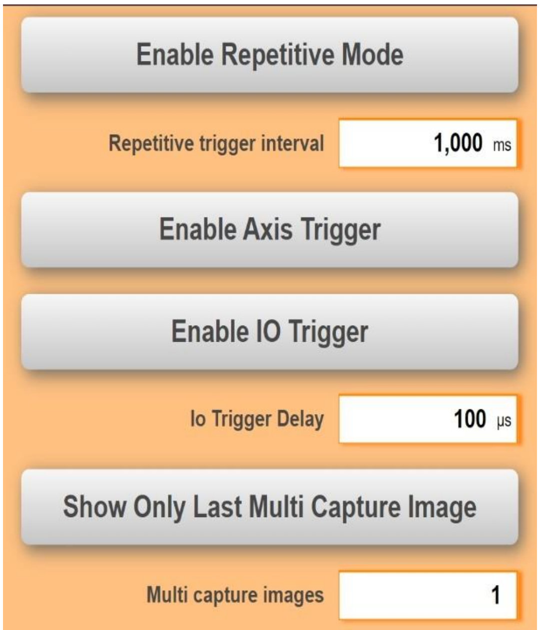
The footer contains the basic image acquisition and vision function settings along with multiple fly outs with more advance information and settings.

- **Gain** can be used to apply a gain on the captured image
- **Focus** sets the used focus length of the camera, this is specified in mm from the camera glass
- **Exposure** set the used exposure time of the image acquisition.
- **Trigger** on click it will trigger an image acquisition
- **Enable VA** when on the vision function processing is enabled
- **Max items count** set the number of results that should be searched for, 10 is maximum as this is defined for the vision function in the vision application.
- **Items found** show the actual number of fund results
- **Camera Processing Time** is the time that it took the camera to process all the vision functions. If only one vision function (smart sensor) is configures it will be almost equal to the function processing time.
- **Function Processing time** is the time it took to process the vision function.

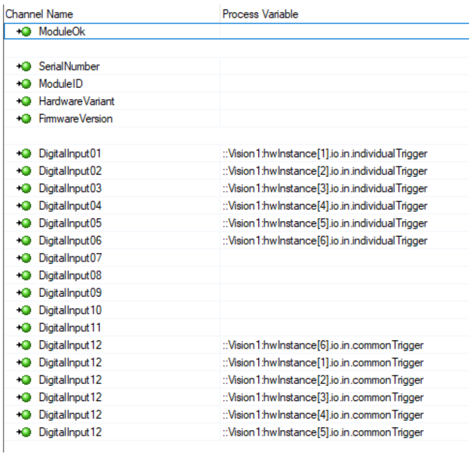
In the following section the six fly-out are described

4.3.2.1 Flyout: Trigger Settings

In this flyout its possible to configure different functionalities concerning the image triggering.



- **Enable Repetitive mode**, if this is true then image acquisitions are executed with the defined interval in **Repetitive trigger interval.**
- **Enable Axis Trigger**, this will enable the image acquisition based on the motion axis configuration on the motion axis page.
- **Enable IO Trigger**, this will enable image acquisition from an IO point. There is defined both individual IO for each camera and on common. A delay from IO goes active to image trigger can be set with the **Io Trigger Delay**



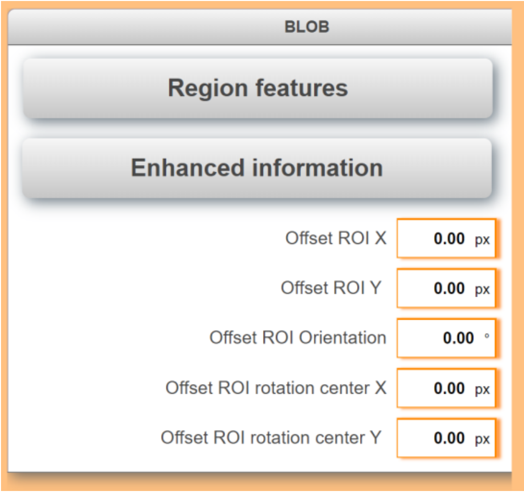
Channel Name	Process Variable
ModuleOk	
SerialNumber	
ModuleID	
HardwareVariant	
FirmwareVersion	
DigitalInput01	::Vision1:hwInstance[1].io.in.individualTrigger
DigitalInput02	::Vision1:hwInstance[2].io.in.individualTrigger
DigitalInput03	::Vision1:hwInstance[3].io.in.individualTrigger
DigitalInput04	::Vision1:hwInstance[4].io.in.individualTrigger
DigitalInput05	::Vision1:hwInstance[5].io.in.individualTrigger
DigitalInput06	::Vision1:hwInstance[6].io.in.individualTrigger
DigitalInput07	
DigitalInput08	
DigitalInput09	
DigitalInput10	
DigitalInput11	
DigitalInput12	::Vision1:hwInstance[6].io.in.commonTrigger
DigitalInput12	::Vision1:hwInstance[1].io.in.commonTrigger
DigitalInput12	::Vision1:hwInstance[2].io.in.commonTrigger
DigitalInput12	::Vision1:hwInstance[3].io.in.commonTrigger
DigitalInput12	::Vision1:hwInstance[4].io.in.commonTrigger
DigitalInput12	::Vision1:hwInstance[5].io.in.commonTrigger

- **Show Only Last Multi Capture Image**, if this is true on the last image of a multi capture is retrieved from the camera and show on the HMI. The results are still updated for each image acquisition.
- **Multi capture images**, here is possible to define multiple image acquisitions based on one trigger (HMI, Repetitive, Axis or IO). If **Show Only Last Multi Capture Image** is false, then new image acquisitions are only performed when the image have been retrieved from the camera.

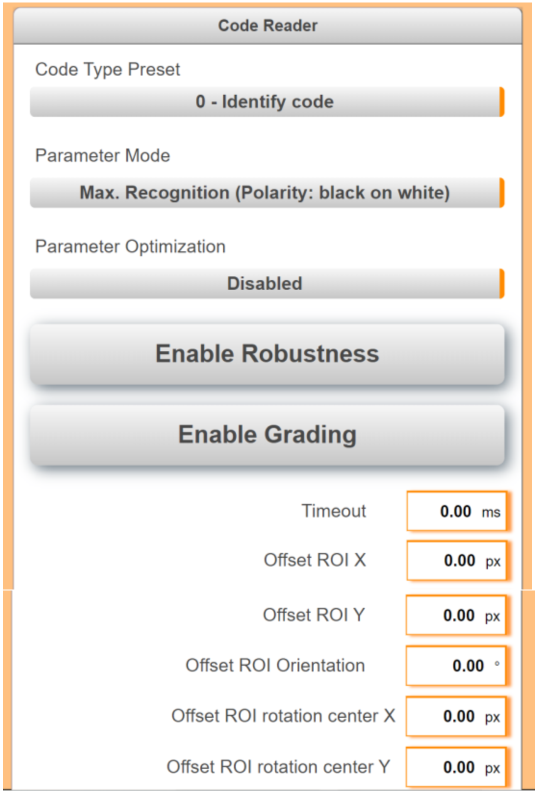
4.3.2.2 Flyout: VF Settings

This flyout will contain the relevant settings for the used vision function in the current vision application. For more information about the individual vision function settings please see the Automation Studio Help file.

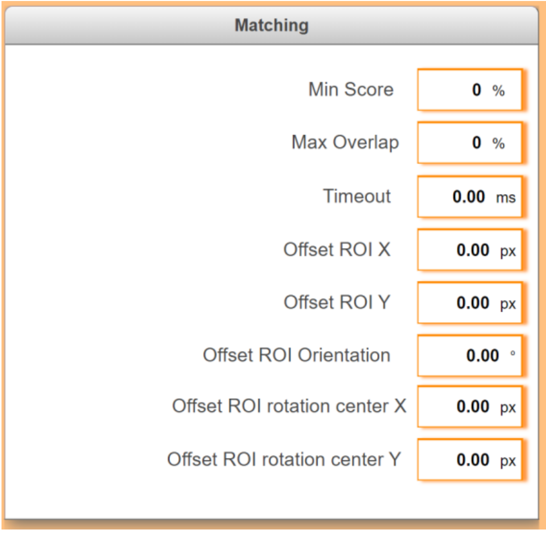
4.3.2.2.1 BLOB



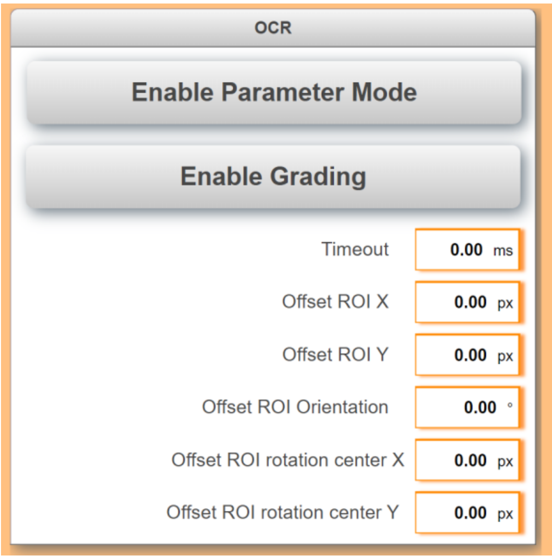
4.3.2.2.2 Code Reader



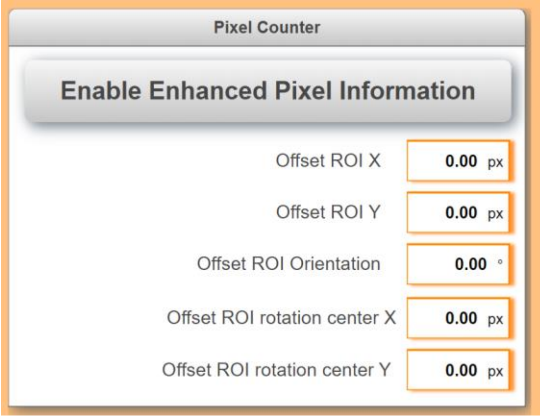
4.3.2.2.3 Matching



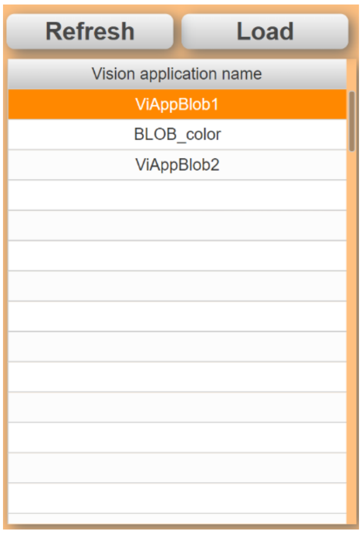
4.3.2.2.4 OCR



4.3.2.2.5 Pixel counter



4.3.2.3 Flyout: Application

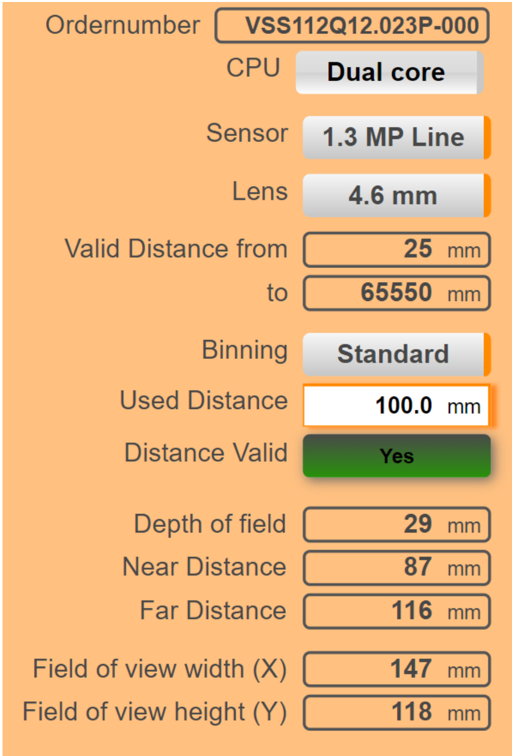


This flyout show all the vision application that exist on the camera. The first vision application is the one currently loaded on the camera. To change vision application select another application on the list and click load.

4.3.2.4 Flyout: Optics

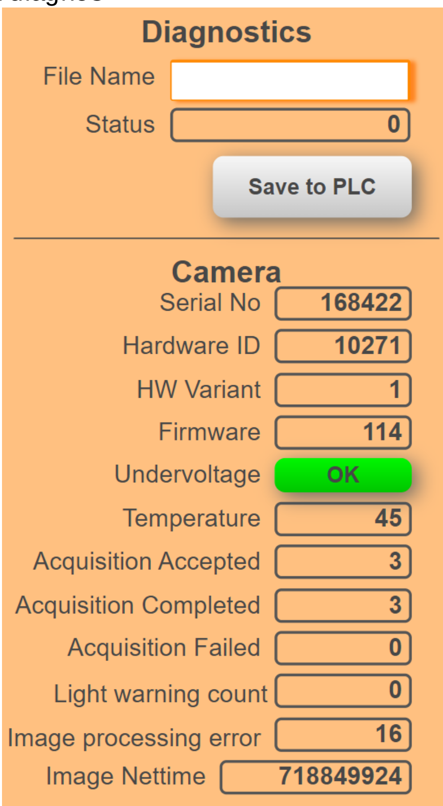
This flyout contain optic relevant information which is retrieved from the used camera product number.

The data can be changed in this flyout, and the optic information is recalculated based on the modified information. This can be used to check FOV at different distances, optics or sensor resolutions. When the flyout is closed, the data is reset to the actual camera again.



4.3.2.5 Flyout: Info

This flyout contains camera information based on the IO points of the camera. For more information about each of these IO points see the Automation Studio help file. It is also possible to save the mappVision diagnostic files to the file device on the user partition.



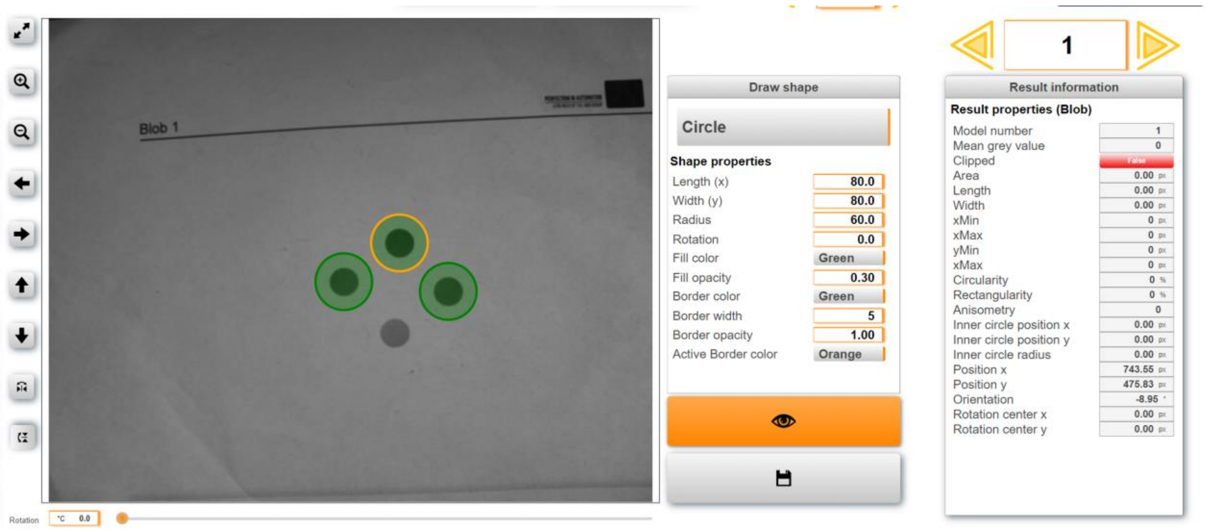
4.3.2.6 Flyout: Setup

On this flyout its possible to setup the flash and status light on the camera. Each of the four falsh segments can be enabled/disabled along with activation of the IR filter. The used Flash color and Status LED color can be configured.

The start Setup button will trigger a search of acquisition settings on the camera, which will try to find the most optimal focus and exposure time. This is done by optimizing contrast in the middle of the image, therefore its important to have something with sharp contrast edges like a printed paper with text on it. The setup procedure can take around 20 seconds.

4.3.3 Main page

The main page is used to show the image and draw an overlay based on the found results on top of the image. Its also possible to see the individual vision function resutl data either by clicking on the overlay or toggle thrug the result information.



If the object is not aligned correctly use the Repetitive Mode to make continues images and allign the object.

The draw shape is used to configure what kind of shape should be drawn for the result. In the image above a green circle with a solid border and an tranparent fill. The selected/active result is show with an orange border.

Its possible to reset view, pan, zoom, flip by using the buttons on the left side of the image.

4.3.4 Results

4.3.4.1 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.

BLOB

No	Model No	Clipped	Area	Position X	Position Y	Orientation	Gray	Length	Width
0	1	0	0.00	743.55	475.83	-8.95	0	0.00	0.00
0	1	0	0.00	847.08	578.41	1.61	0	0.00	0.00
0	1	0	0.00	625.60	557.87	4.67	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00
0	0	0	0.00	0.00	0.00	0.00	0	0.00	0.00

The blob sends only basic information like Position, Model No by default even if more data is configured. To receive all data the parameter "**Enhanced Information**" must be enabled in the **VF settings** flyout

4.3.4.2 Code Reader

The code reader page provides the information that are specific for the code reader functions.

In the **VF Setting** flyout the used code can be selected in the **Code Type Preset** or use "Identify code" to start the process that tries to identify the code automatically. The identification process can run for up to 20s.

It is possible to read multiple codes at the same time but all codes must be of the same code type.

Code Reader

No	Text	Grading	Position X	Position Y	Orientation	Code Type
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0
0		0	0.00	0.00	0.00	0

4.3.4.3 Matching

The matching result page provides the information that are specific for the matching function. The table shows the details for each item that was detected by the camera. Teaching must be done in the Vision Cockpit.

Matching

No	Model No	Score	Position X	Position Y	Orientation	Scale
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0
0	0	0	0.00	0.00	0.00	0

In the **VF** Setting the **Min Score** and **Max Overlap** can adjusted to tune the detection rate. A lower **Min Score** value is more tolerant but can also cause fault detections.

4.3.4.4 Measurement

The measurement result 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.

Measurement

No	Result
1	0.000
2	0.000
3	0.000
4	0.000
5	0.000
6	0.000
7	0.000
8	0.000
9	0.000
10	0.000

4.3.4.5 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.

OCR

No	Text	Grading	Position X	Position Y	Orientation
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00
0		0	0.00	0.00	0.00

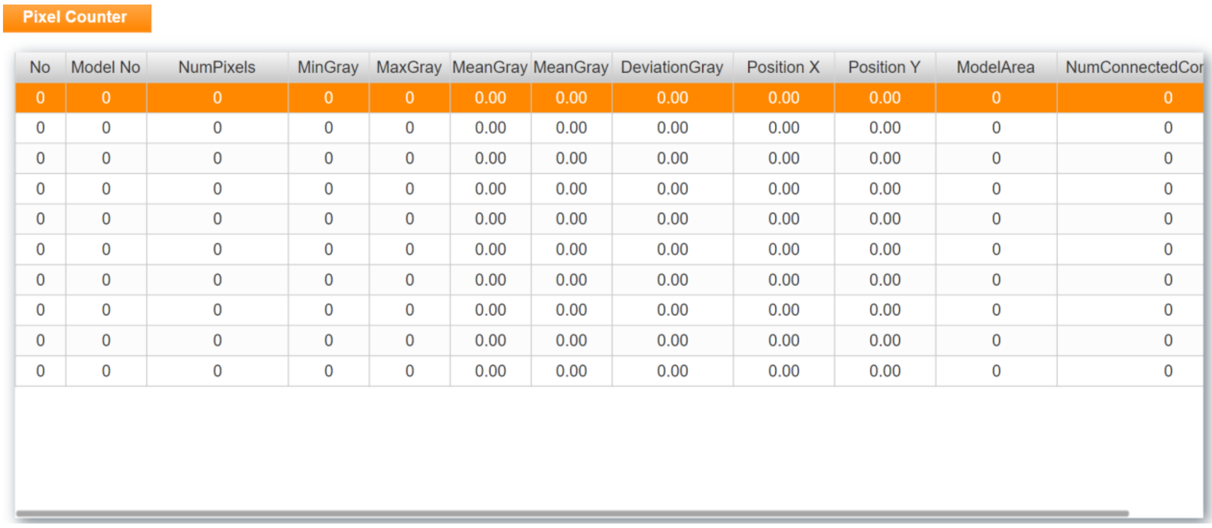
4.3.4.6 Pixel Counter

Vision function Model Based PixelCounter is a function for counting pixels and extracting features from them. Model Based Pixel Counter makes it possible to define regions within which the pixels corresponding to a predefined grayscale value interval (ThresholdMin/ThresholdMax) are counted.

No	Model No	NumPixels	MinGray	MaxGray	MeanGray	DeviationGray	Position X	Position Y	ModelArea	NumConnectedComponents
1	0	0	0	0	0.00	0.00	0.00	0.00	0	0
2	0	0	0	0	0.00	0.00	0.00	0.00	0	0
3	0	0	0	0	0.00	0.00	0.00	0.00	0	0
4	0	0	0	0	0.00	0.00	0.00	0.00	0	0
5	0	0	0	0	0.00	0.00	0.00	0.00	0	0
6	0	0	0	0	0.00	0.00	0.00	0.00	0	0
7	0	0	0	0	0.00	0.00	0.00	0.00	0	0
8	0	0	0	0	0.00	0.00	0.00	0.00	0	0
9	0	0	0	0	0.00	0.00	0.00	0.00	0	0
10	0	0	0	0	0.00	0.00	0.00	0.00	0	0

Enable Enhanced Pixel Information

If **Enable Enhanced Pixel Information** is disabled in the **VF settings** flyout, only ModelNumber, NumPixels and ModelArea are cyclically filled with data. If outputs are defined that do not correspond to those described above, they will be filled with 0.

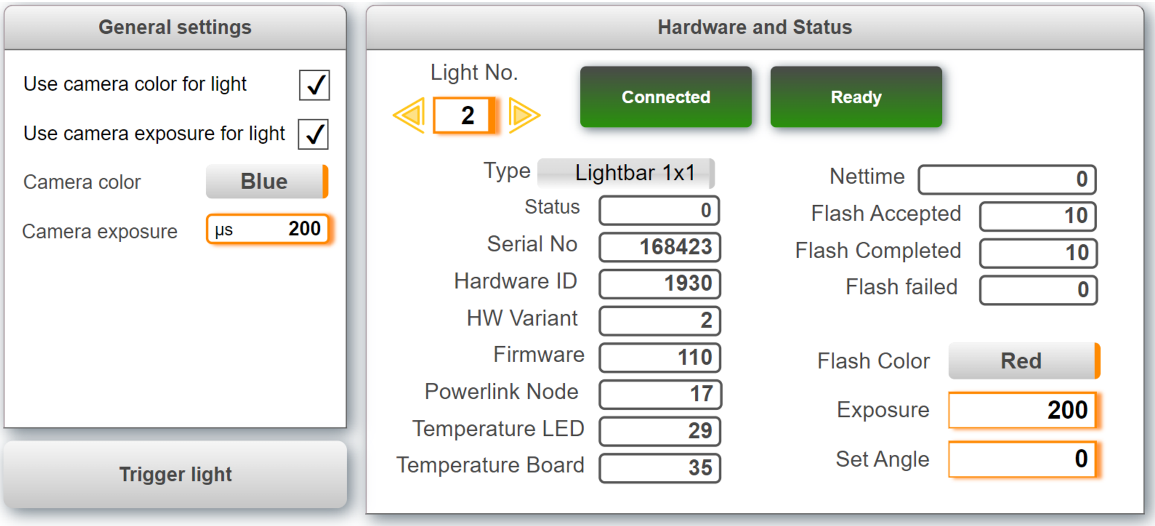

Pixel Counter

No	Model No	NumPixels	MinGray	MaxGray	MeanGray	MeanGray	DeviationGray	Position X	Position Y	ModelArea	NumConnectedCor
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0
0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0	0

Enabling this parameter increases the size of the POWERLINK frame since the remaining cyclically reading parameters are additionally transferred.

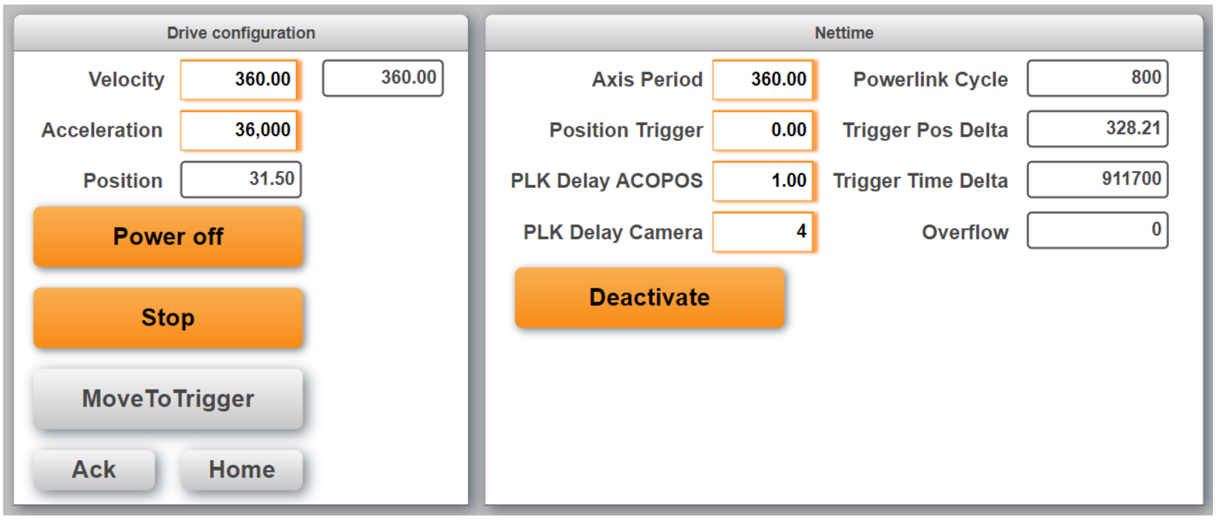
4.3.5 Lights

On the lights page the backlights or lightbars can be tested. On the left side there are some general settings for all the lights connected to the selected camera. If use camera color for light is checked then the camera color for LED is used on the lights instead of the individual selected color on the light. The same applies for the exposure time. In the general settings the camera LED color and exposure time can also be altered.



4.3.6 Motion axis

In some applications it can be necessary to trigger the sensor periodicly depending on a drive position. This is accomplished with the nettime trigger.
The task **axis** provides the necessary calculation for a motion application. 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.

- Power: Switches the axis on and off. In the task "Axis" all the Axis-Handling is done. By default the setting is to use the encoder reference pulse. So when the axis is switched on and not homed it automatically searches the reference pulse
- Run: A continous movement will start with the set velocity and acceleration
- MoveToTrigger: Moves the axis to the "Position Trigger" (Nettime-settings)
- Ack: Acknowledges errors, if there are any errors
- Home: Makes again a homing, also if it was already done automatically while powering on.

On the right hand side are the nettime settings:

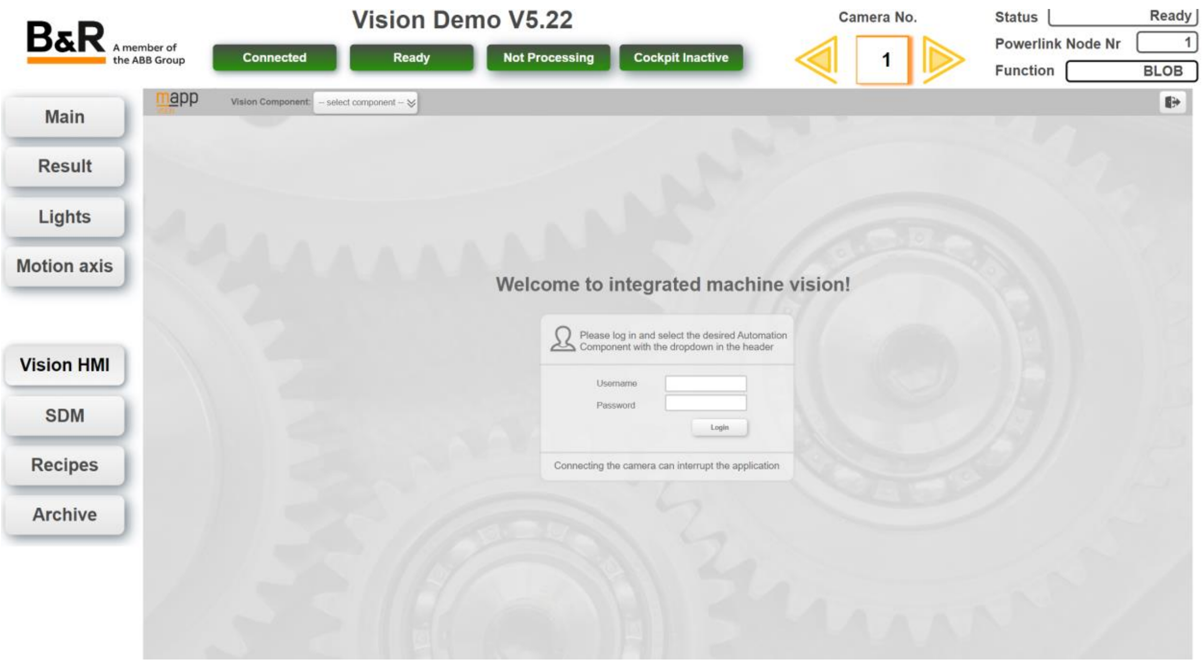
- Axis period: This is the number of units for one cycle (360 for a rotating axis). It should be the same value as in the axis settings
- Position Trigger: This is the position where the image will be made. Should be in the period
- PLK Delay ACOPOS (number of PLK cycles): When the PLC reads the motor position, this position is some time old, e.g. 1 oder 2 PLK cycles. This delay will be compensated. The value is the number of PLK cycles for the "age" of the motorposition. It is possible to use a value with fraction digits. This makes sense because of not only the Powerlink has a delay. E.g. also the encoder could have a small delay. So it is possible to adjust the value very precise
- PLK Delay Camera (number of PLK cycles): This is used to calculate the time when the nettime value must be set at the latest to make it to the sensor in time. The camera needs to get the nettime for the trigger some time before the trigger. A good value is 4. If the value is too small, the camera gets the nettime too late and can't make the image any more. If the value is too high, the camera gets the nettime earlier. A speed-change will then no more be calculated.
- Powerlink Cycle: Powerlink cycletime (in microseconds)
- Trigger Pos Delta: This is the remaining position delta to the next trigger (in units)
- Trigger Time Delta: This is the remaining time delta to the next trigger (in microseconds)

4.3.6.1 Precision

If the nettime seems not to be precise, it makes sense to check the lag error of the axis. To get very good results, a well tuned controller is necessaray.

4.3.7 Vision HMI

This page shows the mappVision HMI embedded into the vision demo HMI.

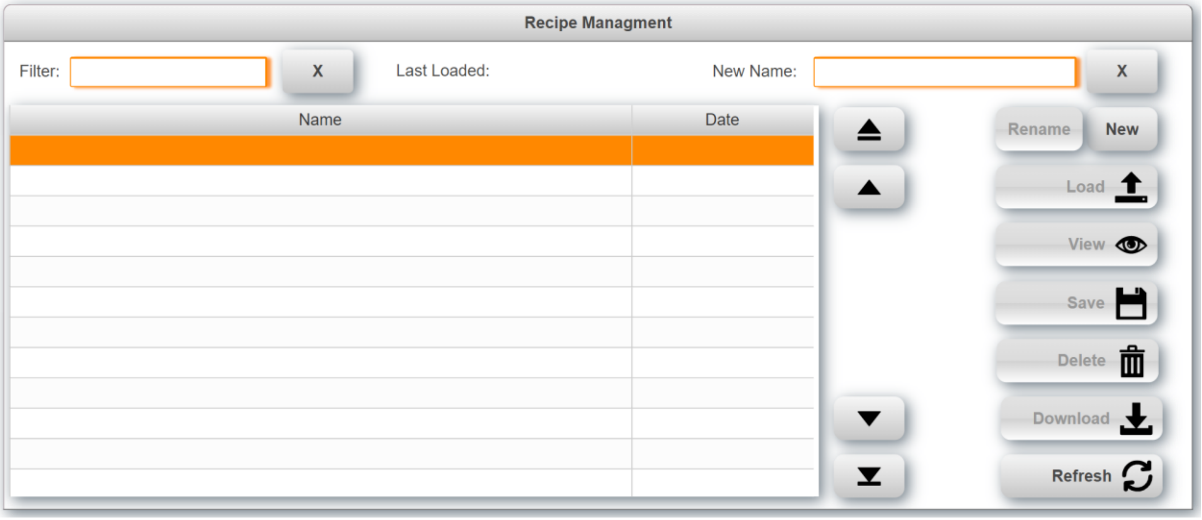


4.3.8 SDM

This page shows the System Diagnostic Manager page. On this page it is possible to check the connected hardware.

4.3.9 Recipes

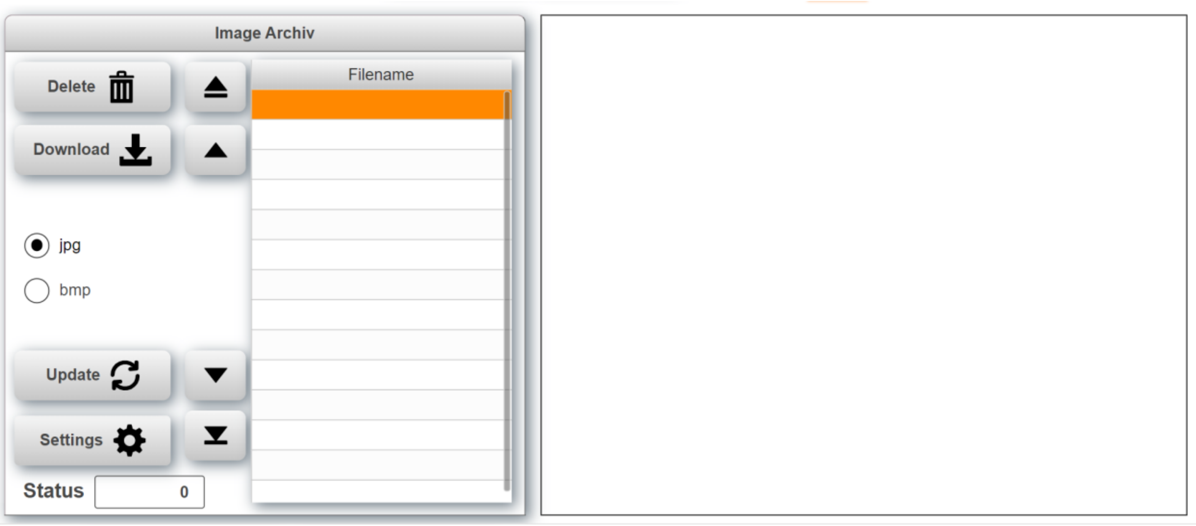
On the recipe page, each camera settings are saved in a CSV file. The data is stored on the user partition.



The data that is stored are all the recipe data structures in the programs.
Its possible to create a new recipe with a given name **New Name.** It can also be renamed, loaded, view on in a flyout, saved, deleted, download (if running from a PC) and the list can be refreshed. Its also possible to filter the recipes.

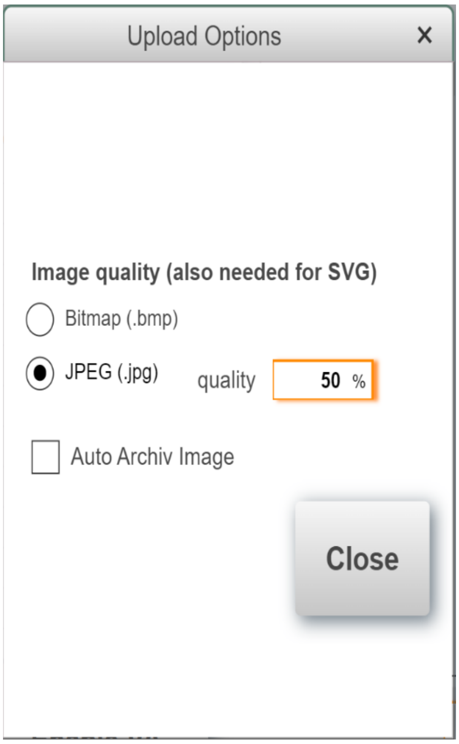
4.3.10 Archive

The image archive is used to store sensor images on the PLC flash card. This can be necessary to inspect 'bad' products later in the process. The image archive is controlled by the **Files** task but the saving of image is done in the **Vision1** task.



When an Image is load the overlay string which are saved in svg is also loaded and shown just like on the main page.

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 sensor creates a BMP or JPEG image. For JPEG images the quality can be selected. Also 100% is possible. The selected image type is both the one shown on the main page and the one which is saved. The overlay string is also saved as an SVG.

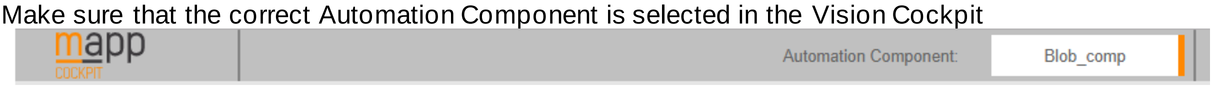
The PLC has the FTP server enabled to check the images remotely. The user name and password is “bundr”.

5 Tips and Hints

5.1 The Vision Cockpit does not work correct and/or does not show the sensor image when the sensor 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



5.2 How to setup a T50 and C50 to use demo?

The T50/C50 must use the PLC as Gateway for the camera image on the main page to work. Assuming that the PLC has the IP address: 192.168.1.100.

T50
Go into the T50 and change the following settings

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

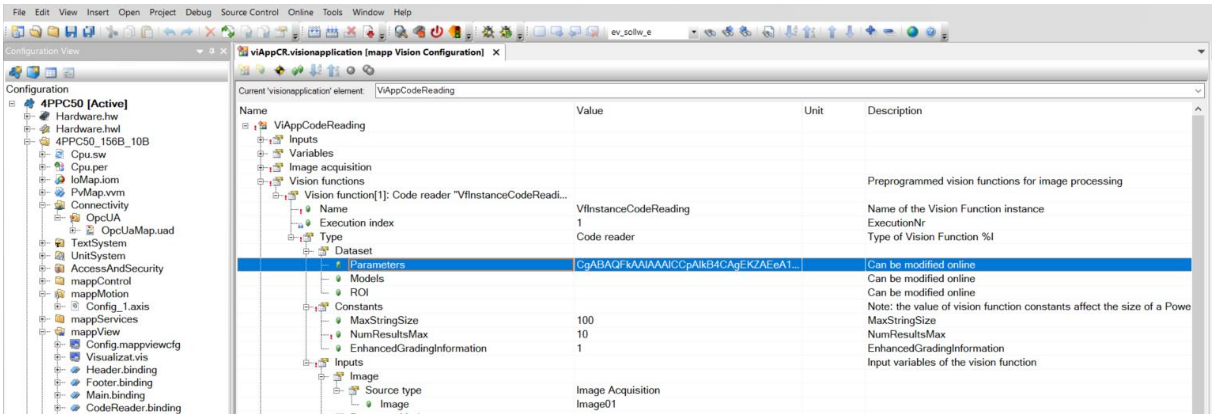
Network:
IP address: 192.168.1.98
Subnet mask: 255.255.255.0

C50
Go into the CPU configuration under Terminal configuration change the network settings

Network:
IP address: 192.168.1.98
Subnet mask: 255.255.255.0

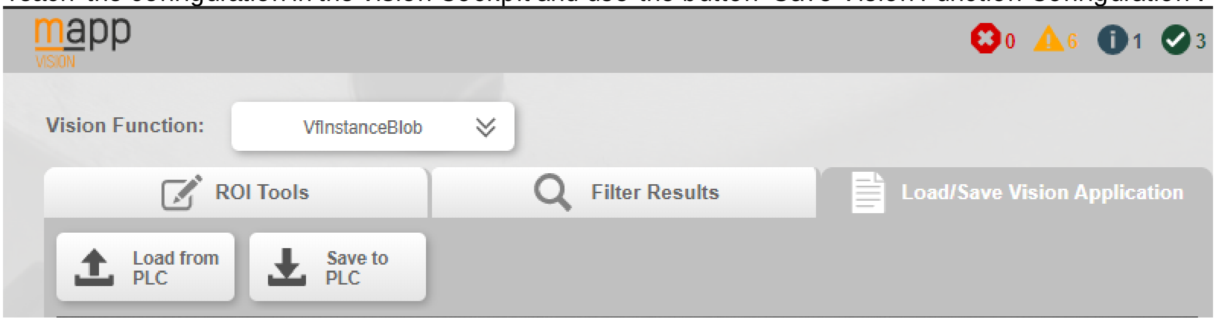
5.3 How is the sensor configuration selected?

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

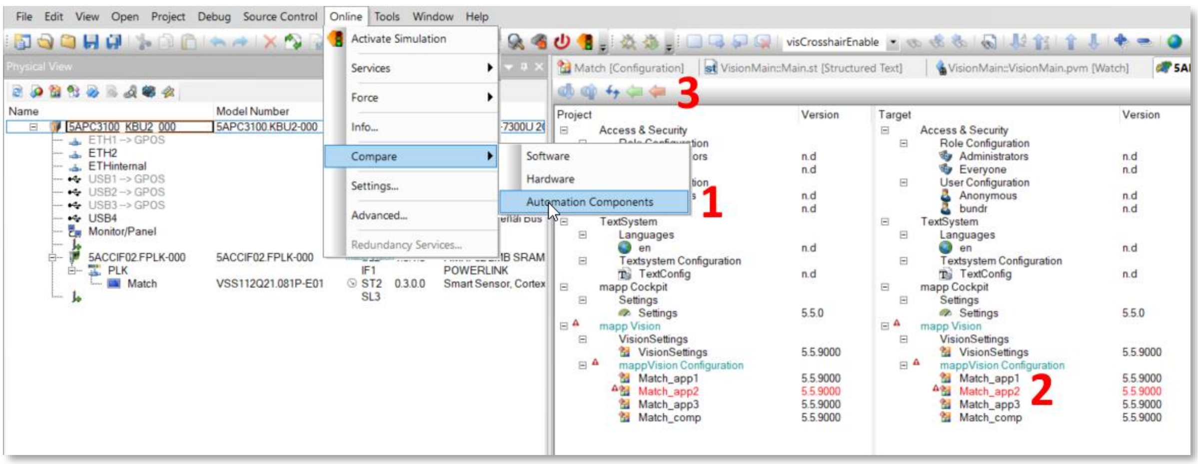


5.4 How to store a configuration taught in the vision Cockpit in the Automation Studio project.

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 the orange arrow (3) at the top to transfer the sensor configuration back to Automation Studio.

6 Revision History

➔ You can find the revision history on GitHub, https://github.com/br-automation-com/mappVision-Demo

Table Index

7 Index