Vision Demo Application

Version 2.0
Date: 01/04/2020

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 sensor 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
Axis	Sample task for an axis, needed for nettime handling
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 demo 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 management

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.

These is an example configuration. It is also possible to have different node numbers or multiple cameras of the same type.

3 Parameter structures

The sample supports multiple cameras but only one is displayed at a time. The global structures begin with a “g” for (gVisionSensor, gBlob, gMT, gCodeReader, gMatch, gOCR). The global structures are arrays where the index represents the camera index. The variable “visSelectedSensor” maps one of the global structures to the dynamic local variable in the task “Vi_main”. The variable “visSelectedLight” maps one of the global structures to the dynamic local variable in the task “Vi_light”.

3.1 Vision sensor structure (gVisionSensor)

The vision structure handles all functions and parameters that are sensor related and are independent from the vision function used.

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
	VisionFunction	The type of vision function that is used with the sensor. This information is used to add the correct detail information on the main page.
	PowerlinkNode	The Powerlink node number for this camera. This information is used to generate the correct IP address for the camera.
	DataStructure	Pointer to the vision speccific function. (gBlob, gMT, gCodeReader, gMatch, gOCR)
	…	All other parameters, see camera 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 (gBlob, gMT, gCodeReader, gMatch, gOCR) 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

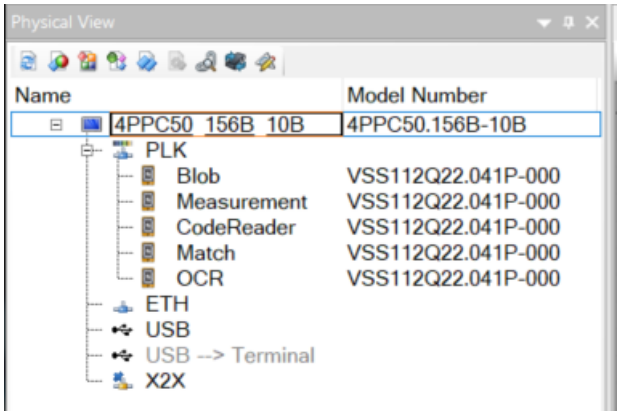
The image structure handles all functions and parameters that are related to the image archive.

CMD		Command structure to trigger an action
	Upload	Upload an image from the sensor 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 sensor
	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 VSS112Q22.041P-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.

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

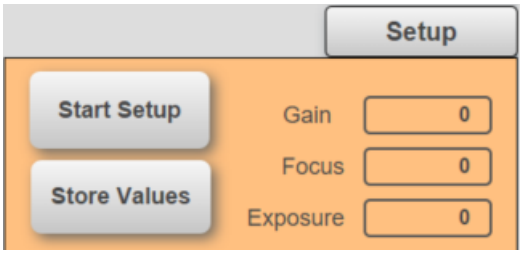
“\ProjectName\Logical\Vision\setRouteToCamera.bat”

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 multiple pages to demonstrate the 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.

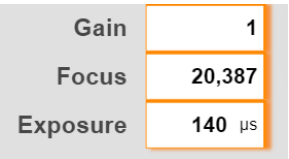
To start the auto setup process click on Setup on the bottom right corner. Click on “Start Setup” to initiate the auto setup that determines the values for gain, focus and exposure.



The sensor light should flash for about 20 seconds. When the process 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 use the Repetitive Mode to make continues images and allign the object.

Click on Trigger to generate a new image. In some cases it may be necessary to adjust the automatically generated values.



Use the crosshair toggle button to show additional information. Images are stored automatically when the checkbox “Auto Archiv Image” is set (see 4.12). Details results can also be viewed on the “Results” page. Google Chrome browser must be used for the mappView project; other browser may not show the correct crosshair position.

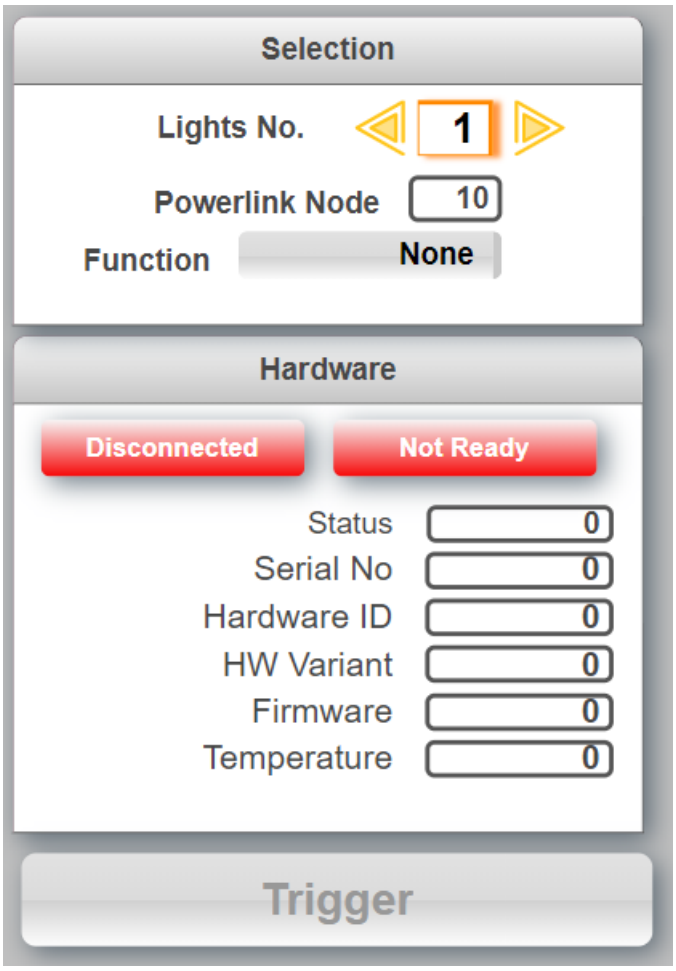
4.3 Changing the demo

The different tasks are designed to easily integrate into an existing application. All interactions are handle by variable structures. To allow easy update of these task they should not be changed. The demo includes a task “YourTask” that can be used to write own code.

The task includes the sample configuration the different vision functions and lights. It also includes a basic state machine to connect a drive and use the nettime functionality.

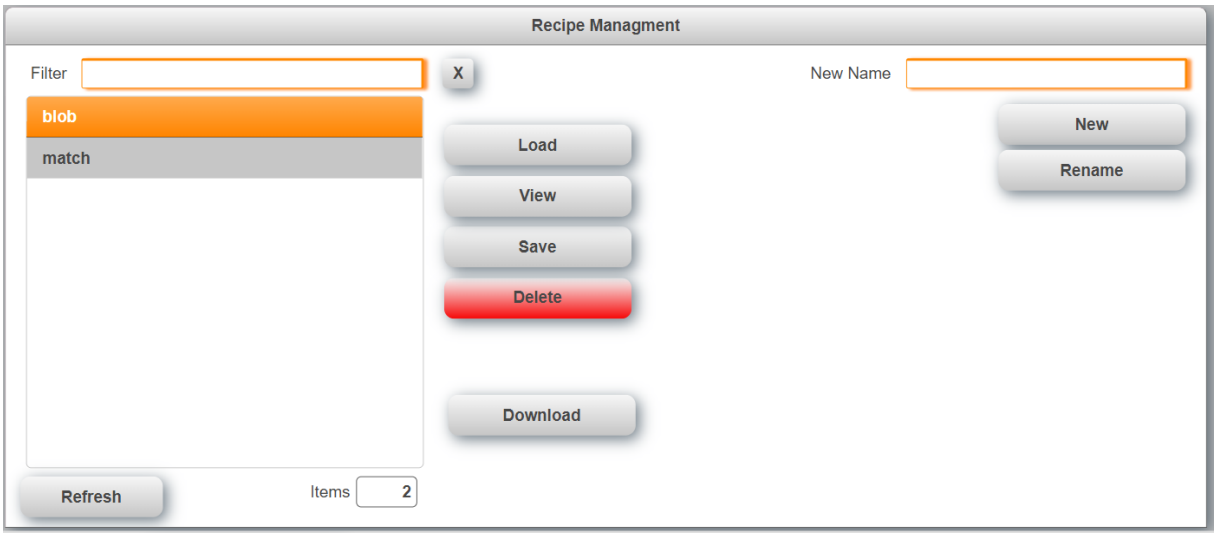
4.4 Lights

The lights page the backlights or barlights can be tested.



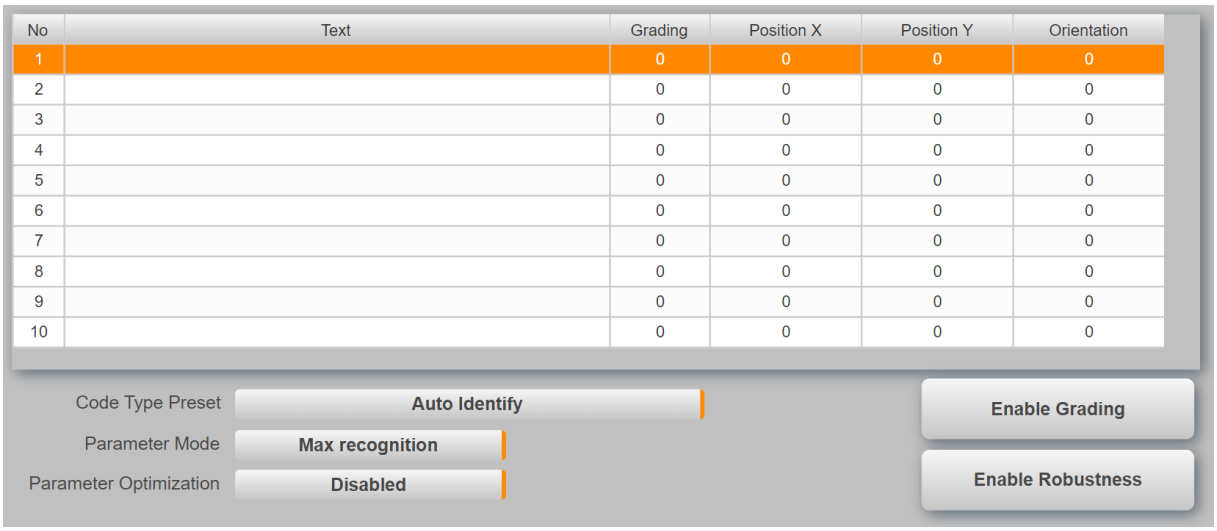
4.5 Recipe

On the recipe page, the camera settings are saved in a CSV file. The data is stored on the user partition. Right now, only the data from the variable “gVisionSensor” is saved.



4.6 Code Reader

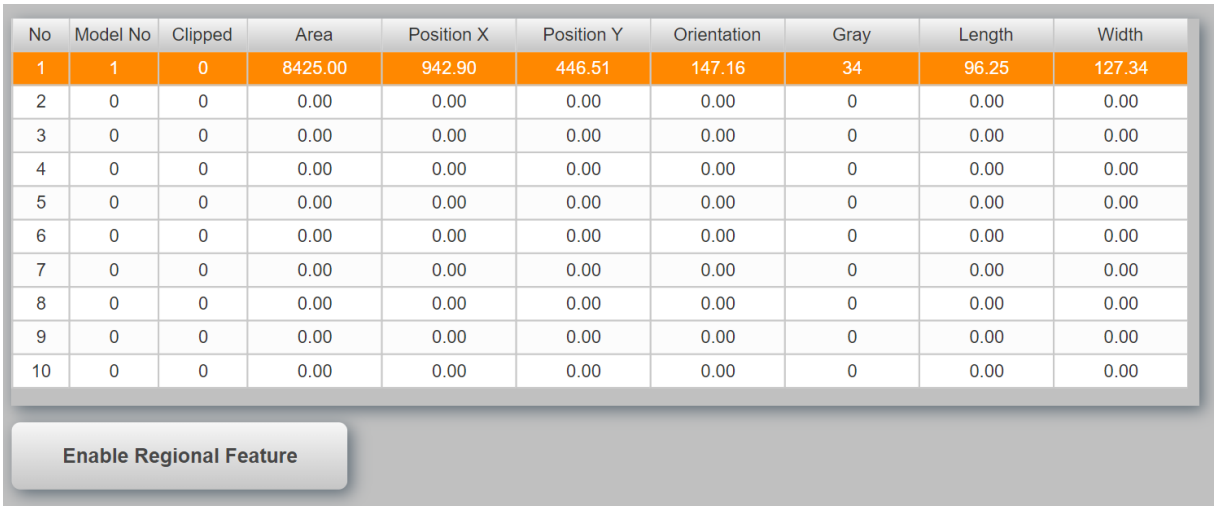
The code reader page provides the information that are specific for the code reader functions. Select the code type from the drop down menu or use “Auto Identify” to start the process that tries to identify the code automatically. The identification process can run for up to 20s.



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

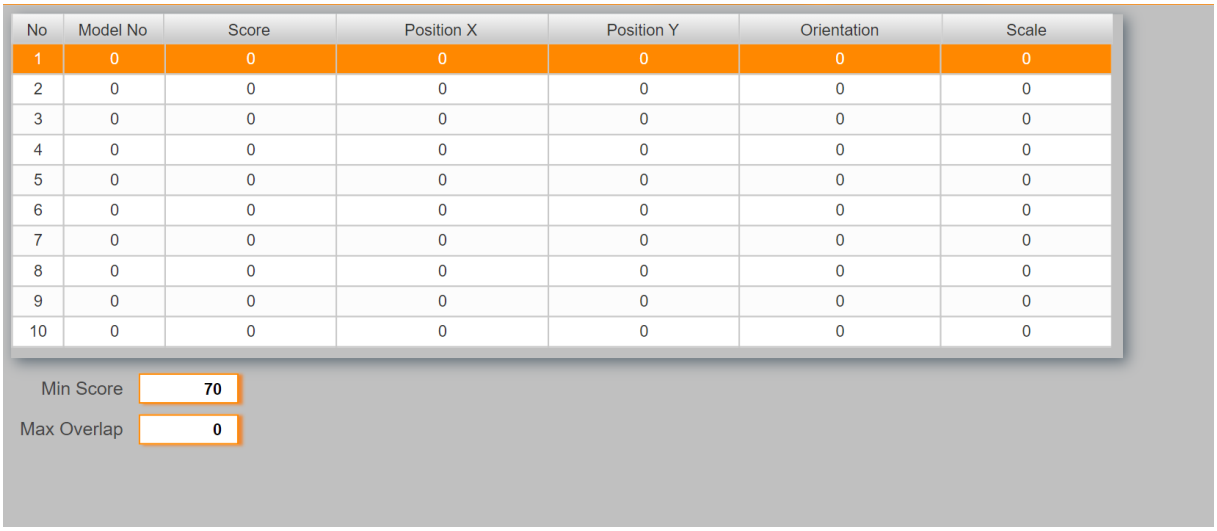
4.7 Blob

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.



4.8 Match

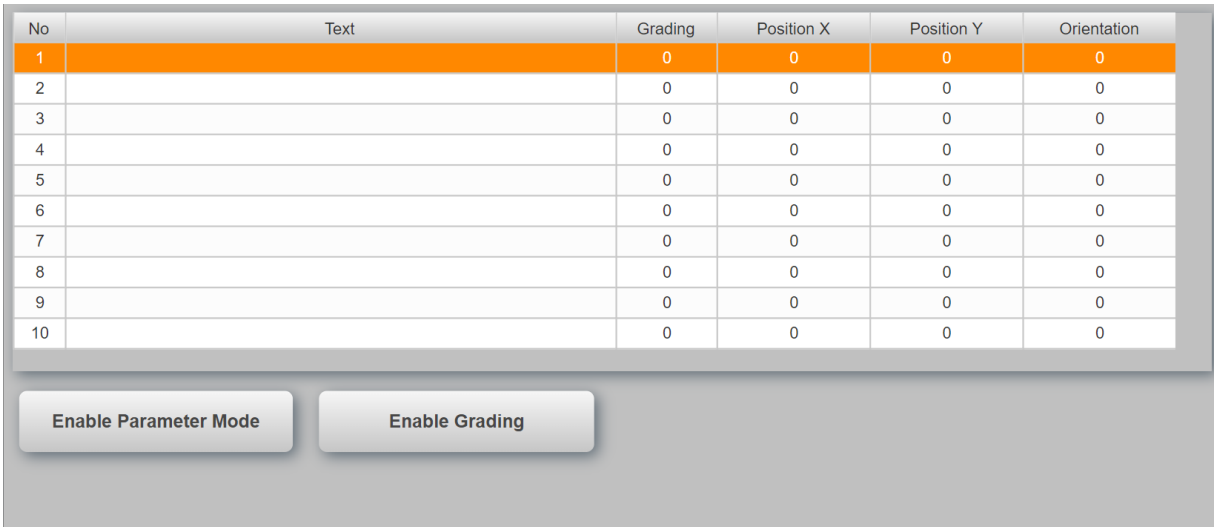
The match page provides the information that are specific for the match functions. The table shows the details for each item that was detected by the sensor. Teaching must be done in the Vision Cockpit.



Use the value “Min Score” to adjust the detection rate. A lower value is more tolerant but can also cause fault detections.

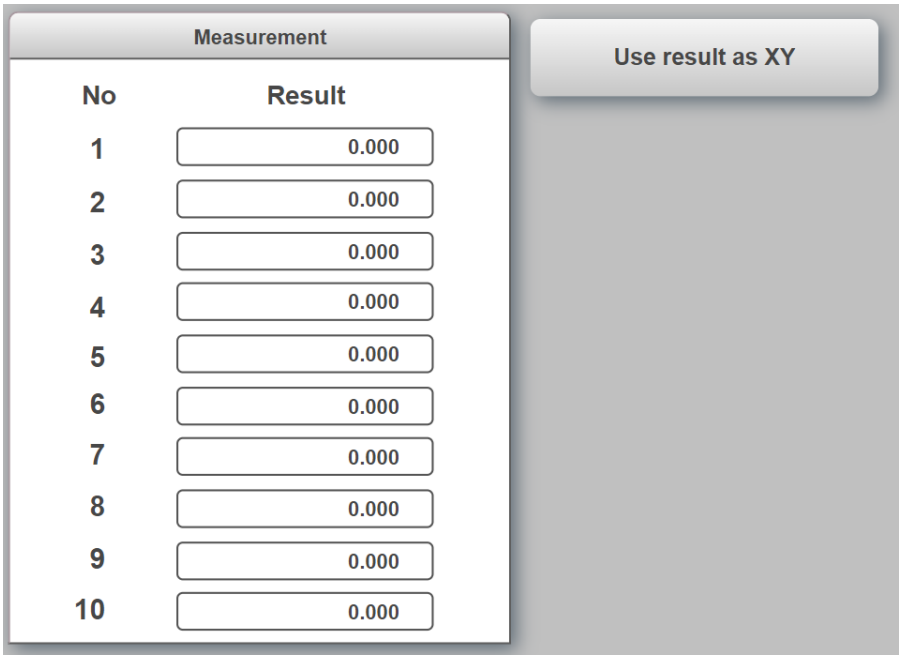
4.9 OCR

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.



4.10 Measurement

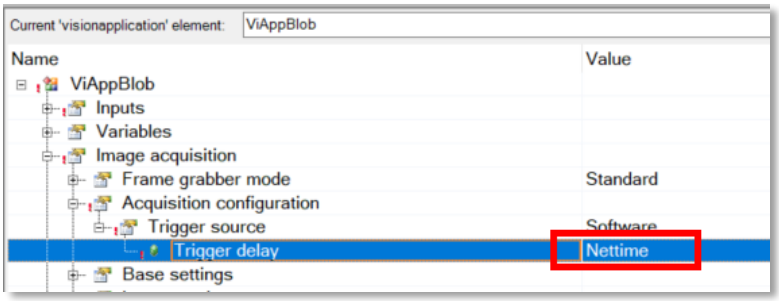
The measurement page provides the information that are specific for the edge measurement functions. This page shows the results for the different measurement functions. What is measured must be configured in the vision cockpit.



For edge detection it 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.

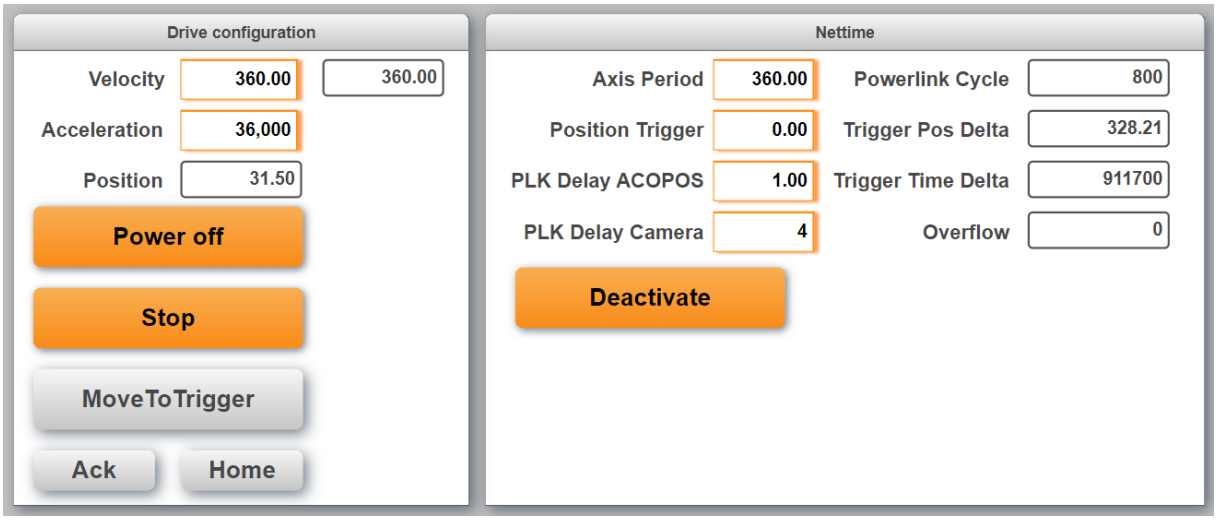
4.11 Using Nettime

In some applications it can be necessary to trigger the sensor periodicly depending on a drive position. This can be accomplished with nettime. To enable nettime the trigger delay has to be changed 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 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

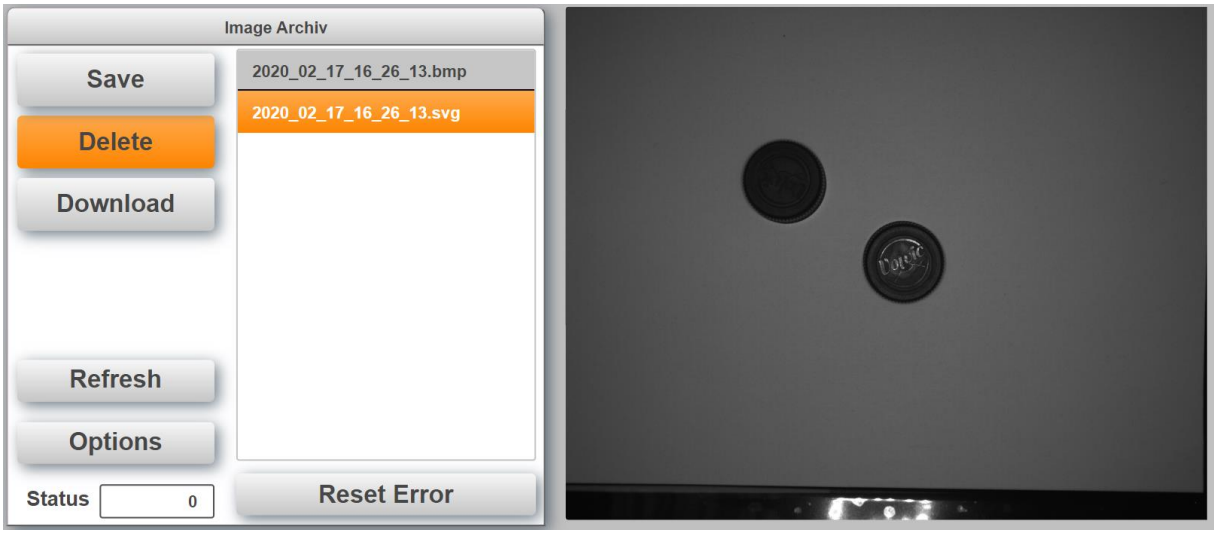
- PLK Delay ACOPOS (number of PLK cycles): When the PLC reads the motor position, this position is some time old, e.g. 1 oder 2 PLK cycles. This delay will be compensated. The value is the number of PLK cycles for the “age” of the motorposition. It is possible to use a value with fraction digits. This 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)
- Overflow: If the nettime handling wants to send the next trigger to the camera, but the camera is not ready, this value will be increased by 1.

4.11.1 Precision

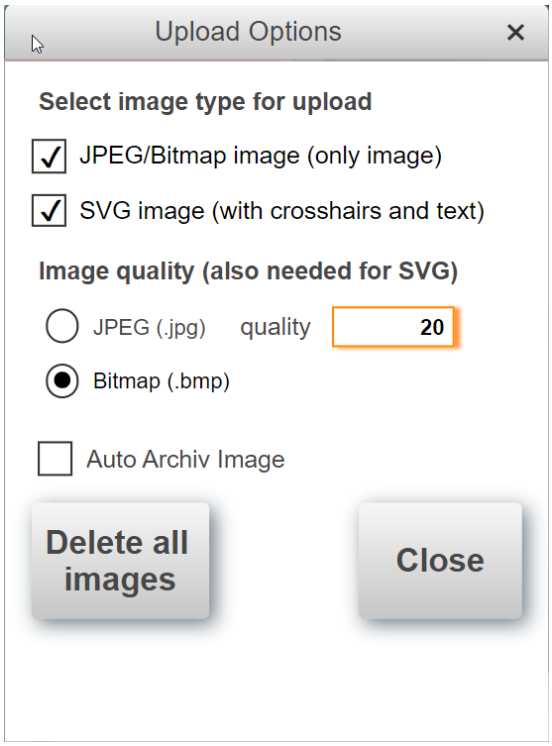
If the nettime seems not to be precise, it makes sense to check the lag error of the axis. To get very good results, a well tuned controller is necessaray.

4.12 Image 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 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 sensor 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

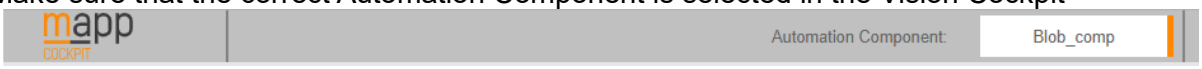
5.1 Sensor is connected and ready but the image on the main page is not refreshed

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 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



5.3 How to setup a T50 to use demo?

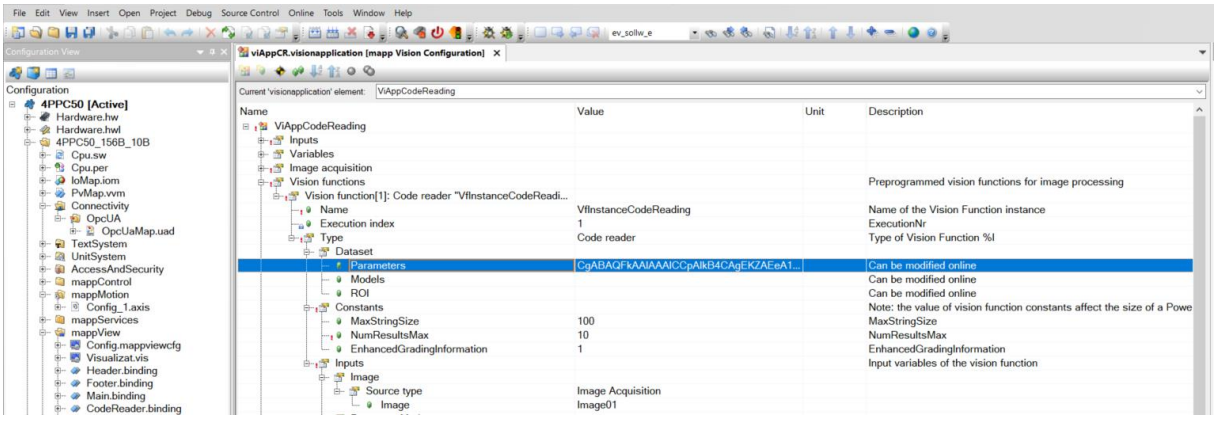
Assuming that the PLC has the IP address: 192.168.1.100. Go into the T50 and change the following settings

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

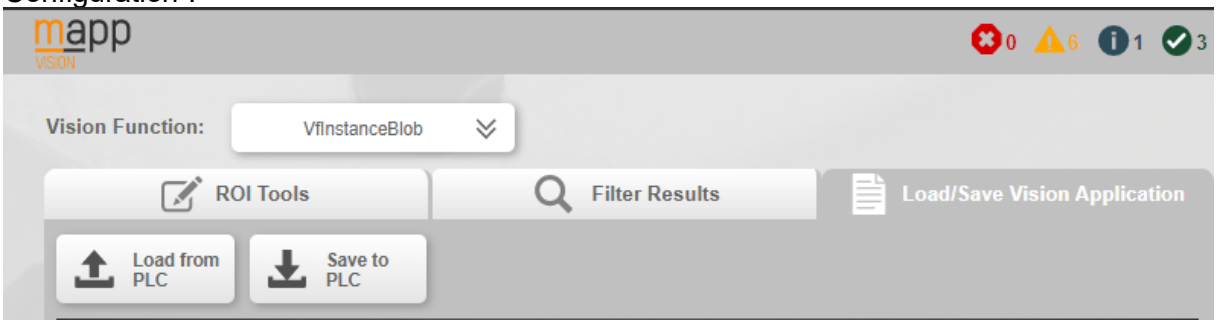
5.4 How is the sensor configuration selected?

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

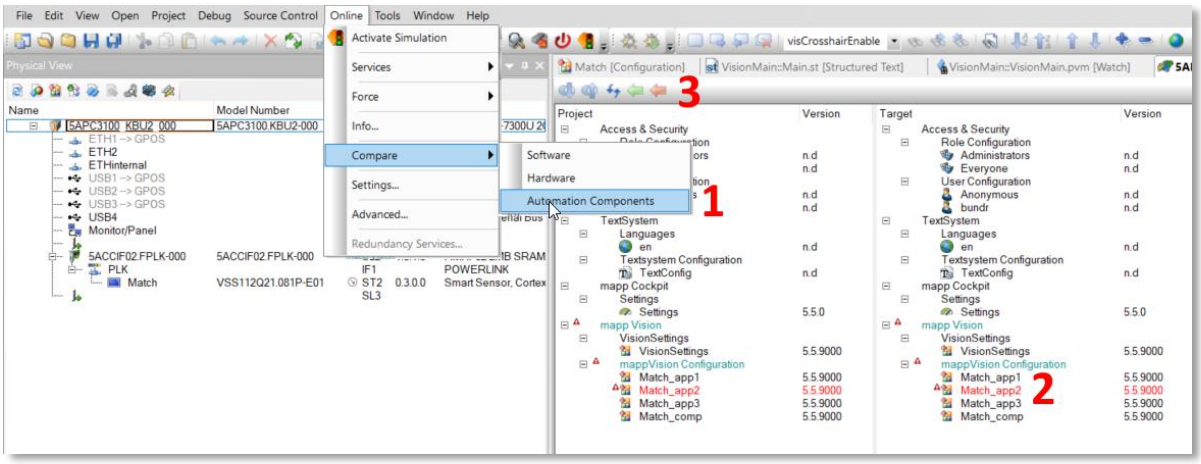


5.5 How to store a configuration taught in the vision Cockpit in the Automation Studio project.

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 sensor 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