



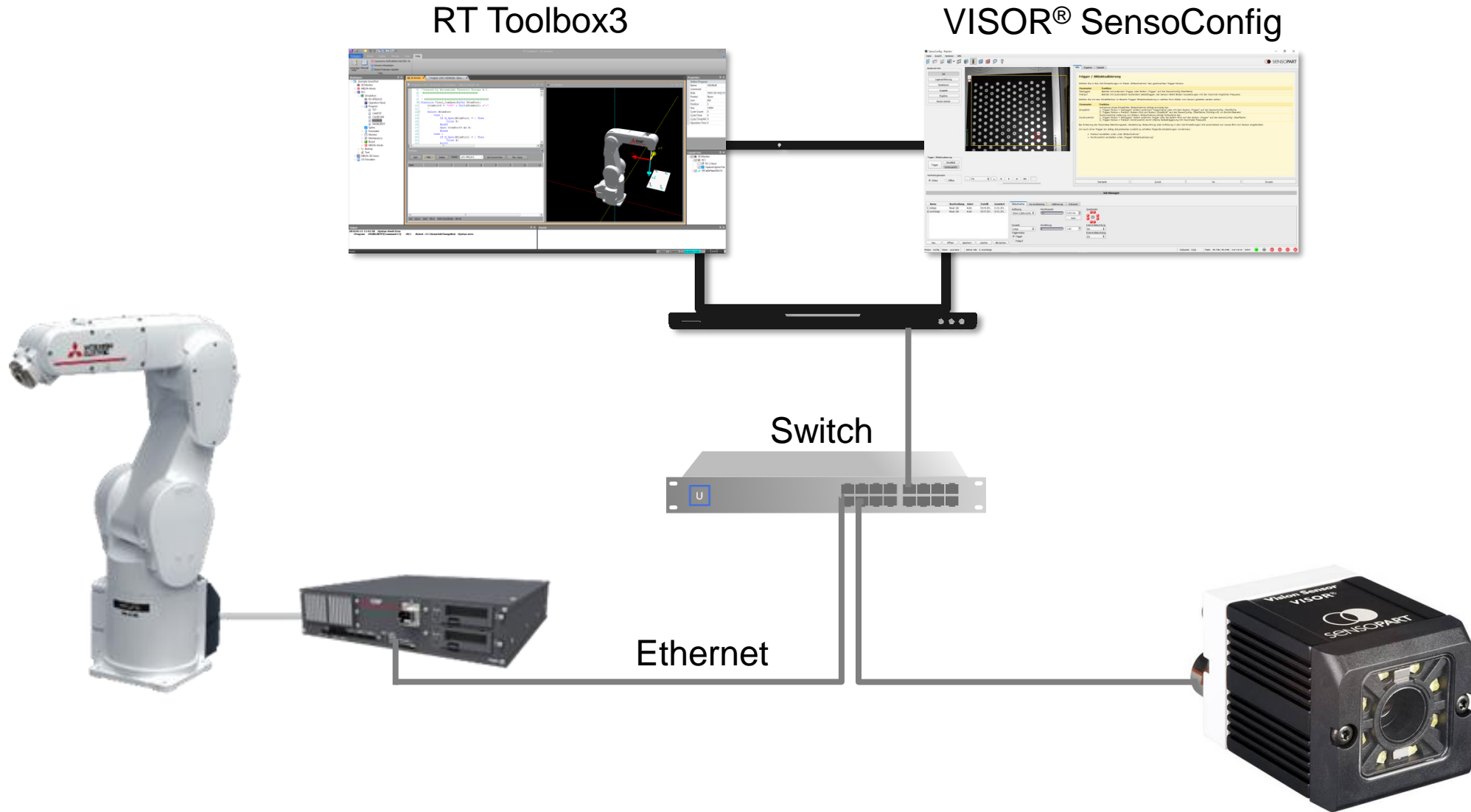
Connecting...

Sensopart VISOR® and MELFA Roboter

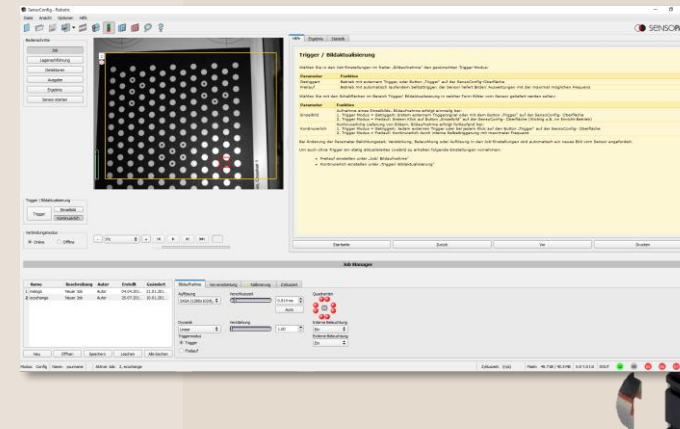
VISOR Robotic Mitsubishi Library V1.0
May, 2nd 2019



Setup

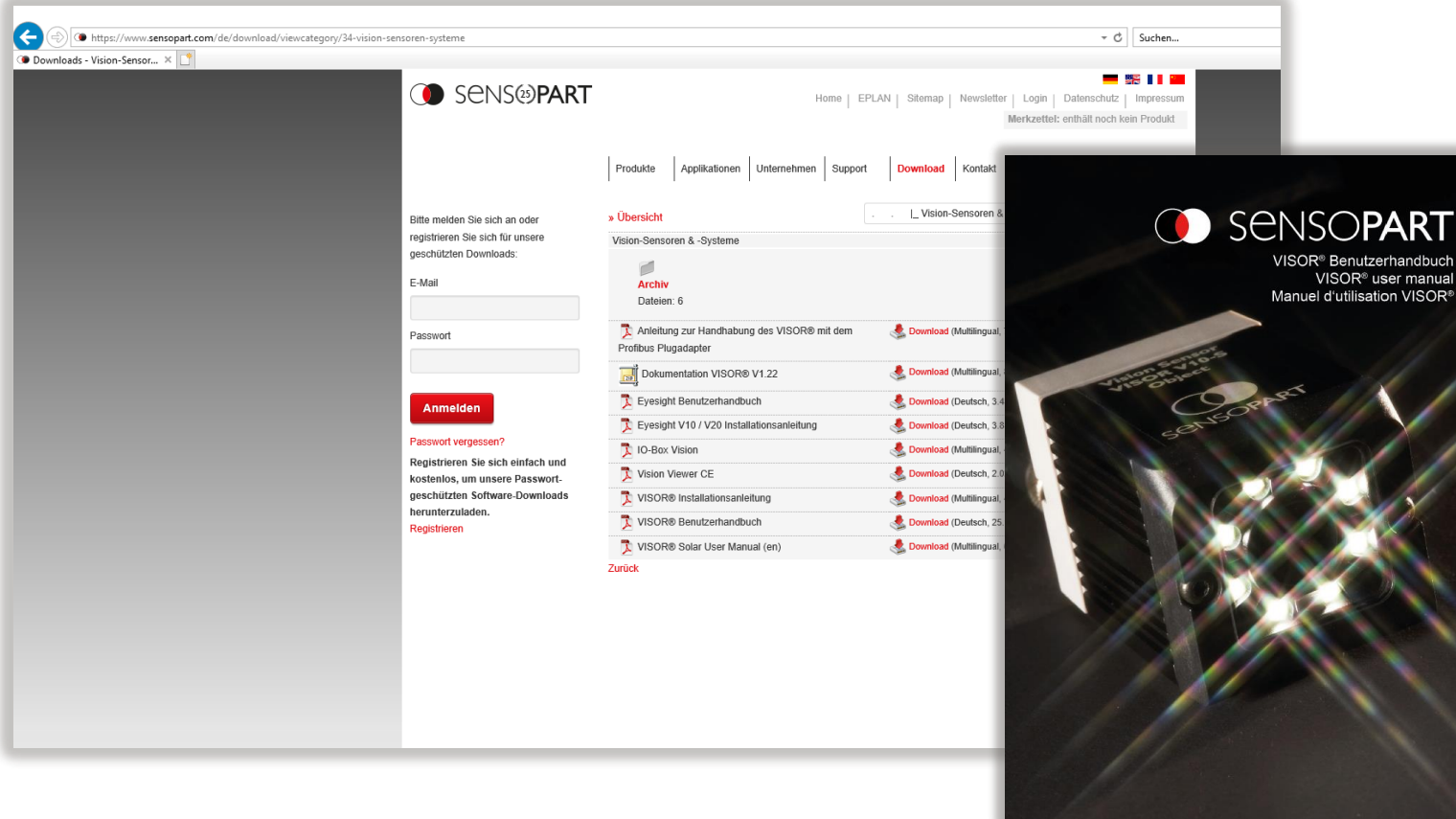


Setup VISOR® Robotic with VISOR® PC software



Setup VISOR® Robotic

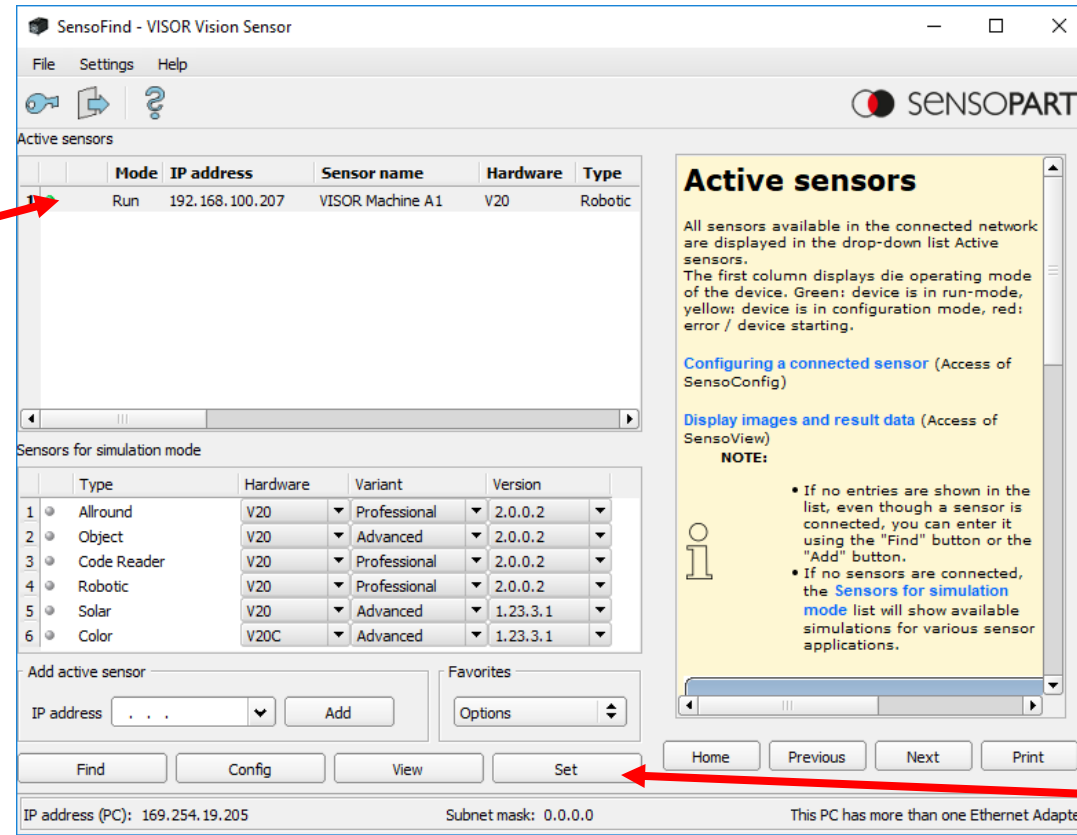
- Download all necessary tools and information from SensoPart homepage



<https://www.sensopart.com/de/download/finish/34-vision-sensoren-systeme/3911-visor-benutzerhandbuch>

Setup VISOR® Robotic

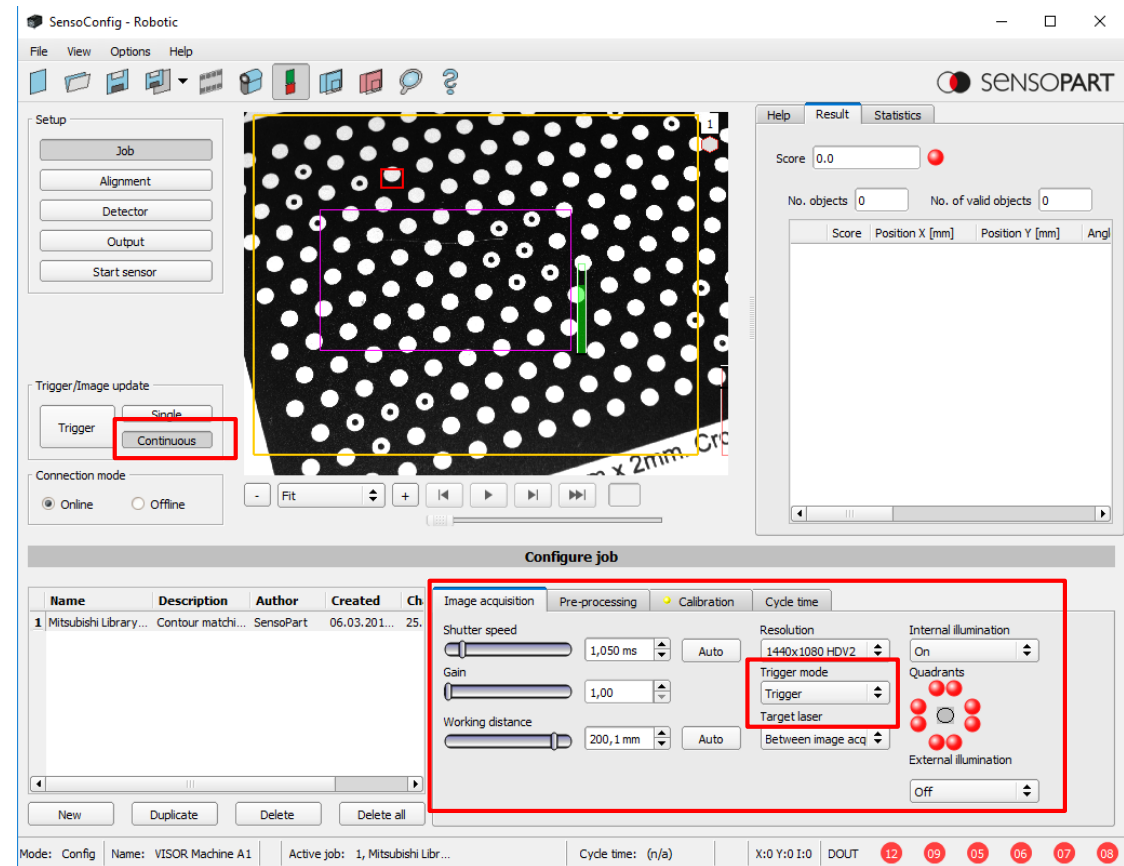
- Start VISOR® PC software and connect with the vision sensor



Setup IP-address of the VISOR® if necessary.

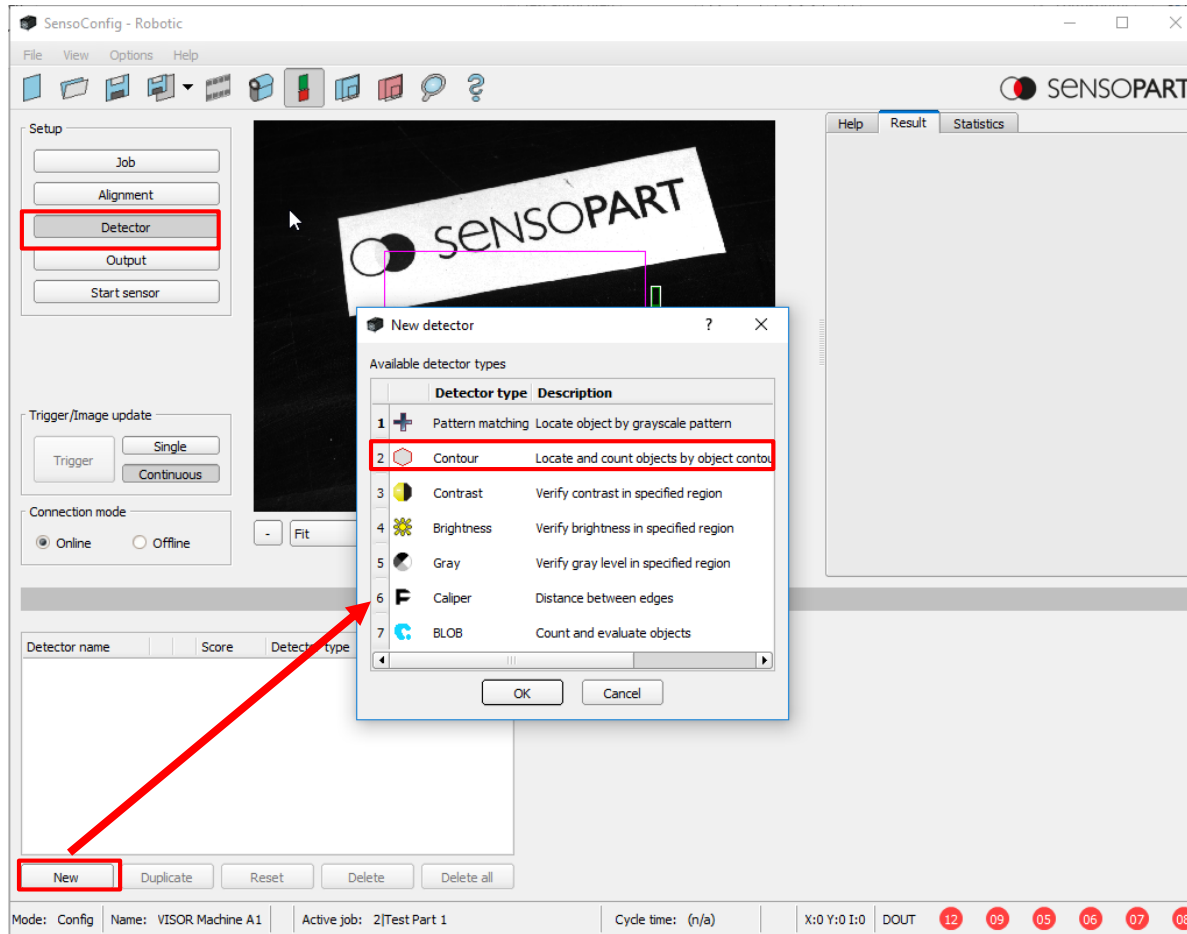
Setup VISOR[®] Robotic

- Adjust image acquisition parameters (shutter, working distance etc.)
- Set trigger mode to “trigger”



Setup VISOR[®] Robotic

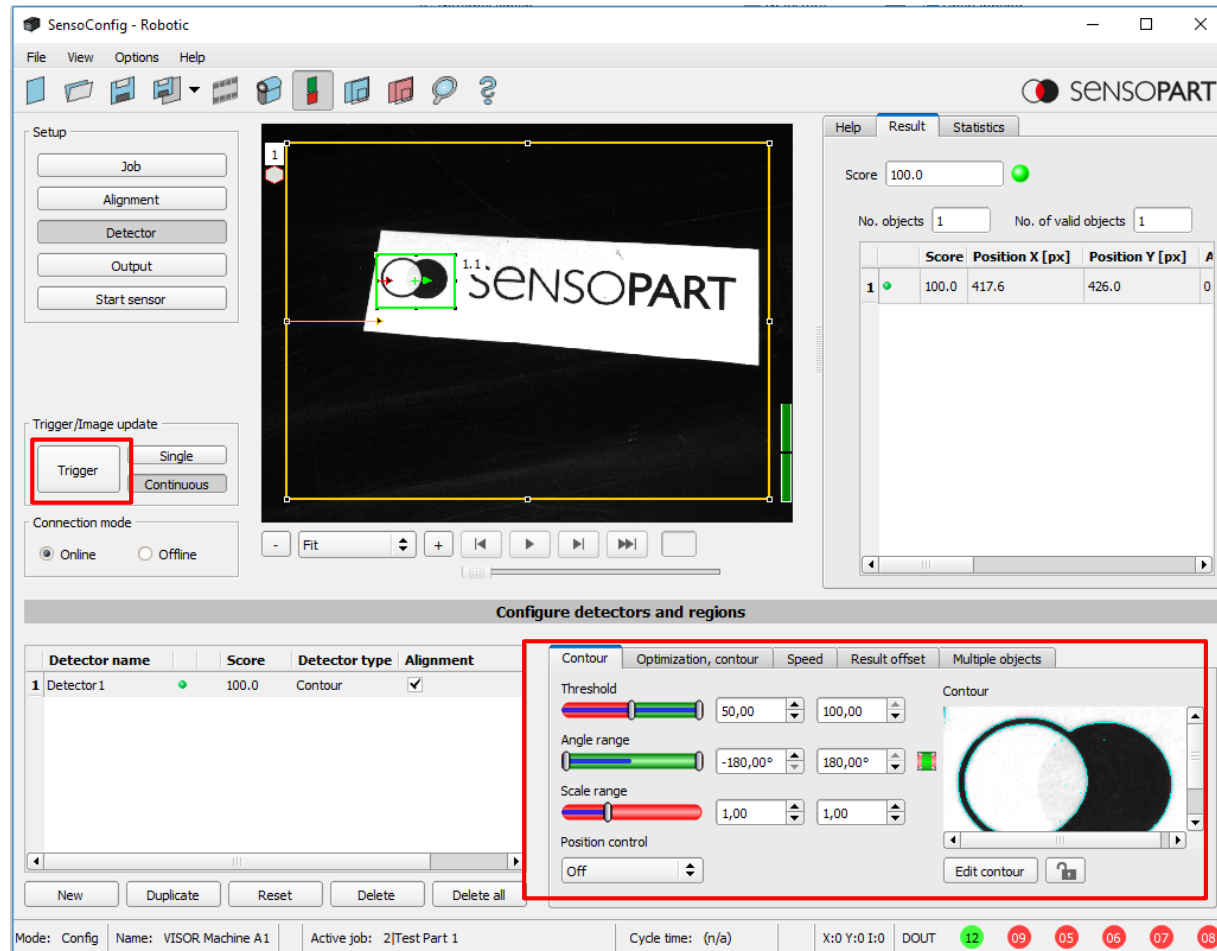
- Create new detector (e.g. contour detector)



Setup VISOR® Robotic

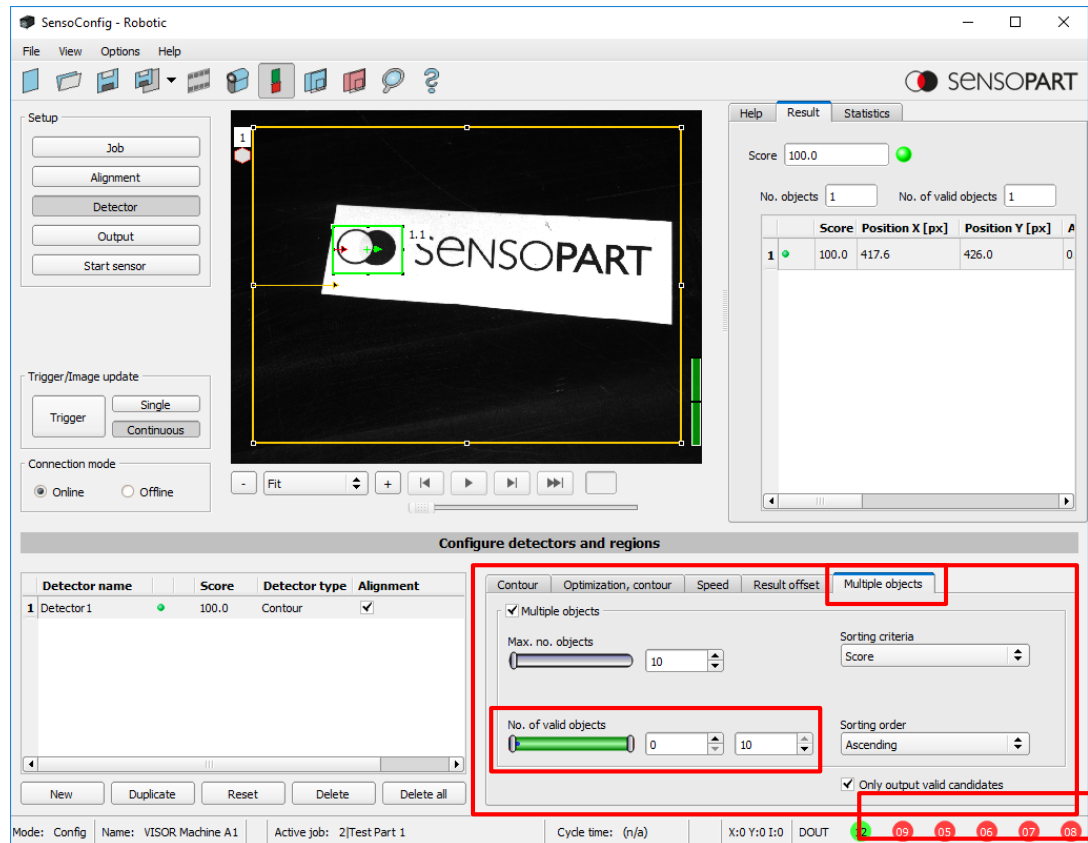
- Place the part the VISOR® should search for in the field of view
- Teach the detector

- With Trigger you can update the image



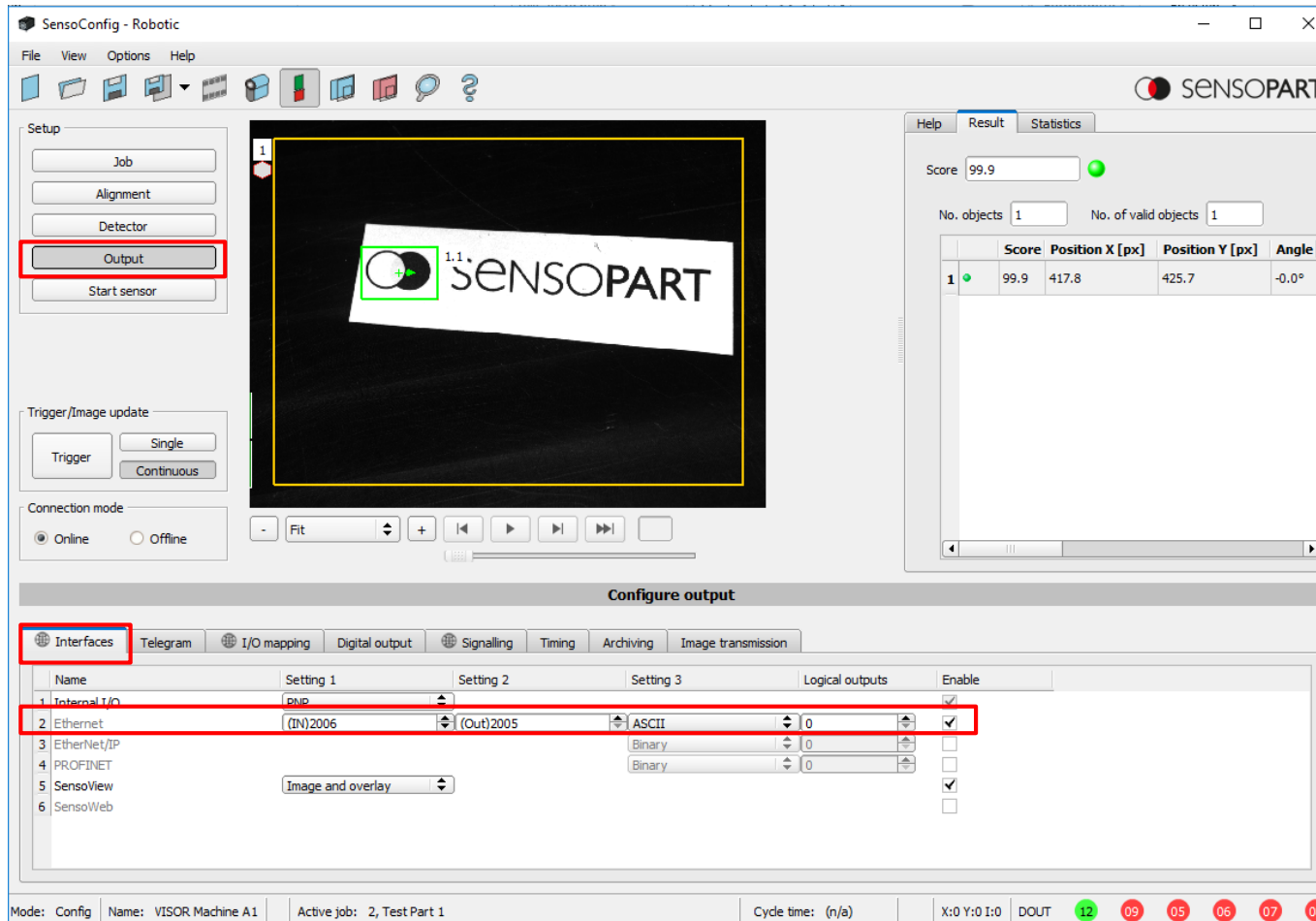
Setup VISOR® Robotic

- If necessary the VISOR® Robotic can detect up to 200 parts with contour detector.
- Activate the function multiple part detection and select only valid candidates.



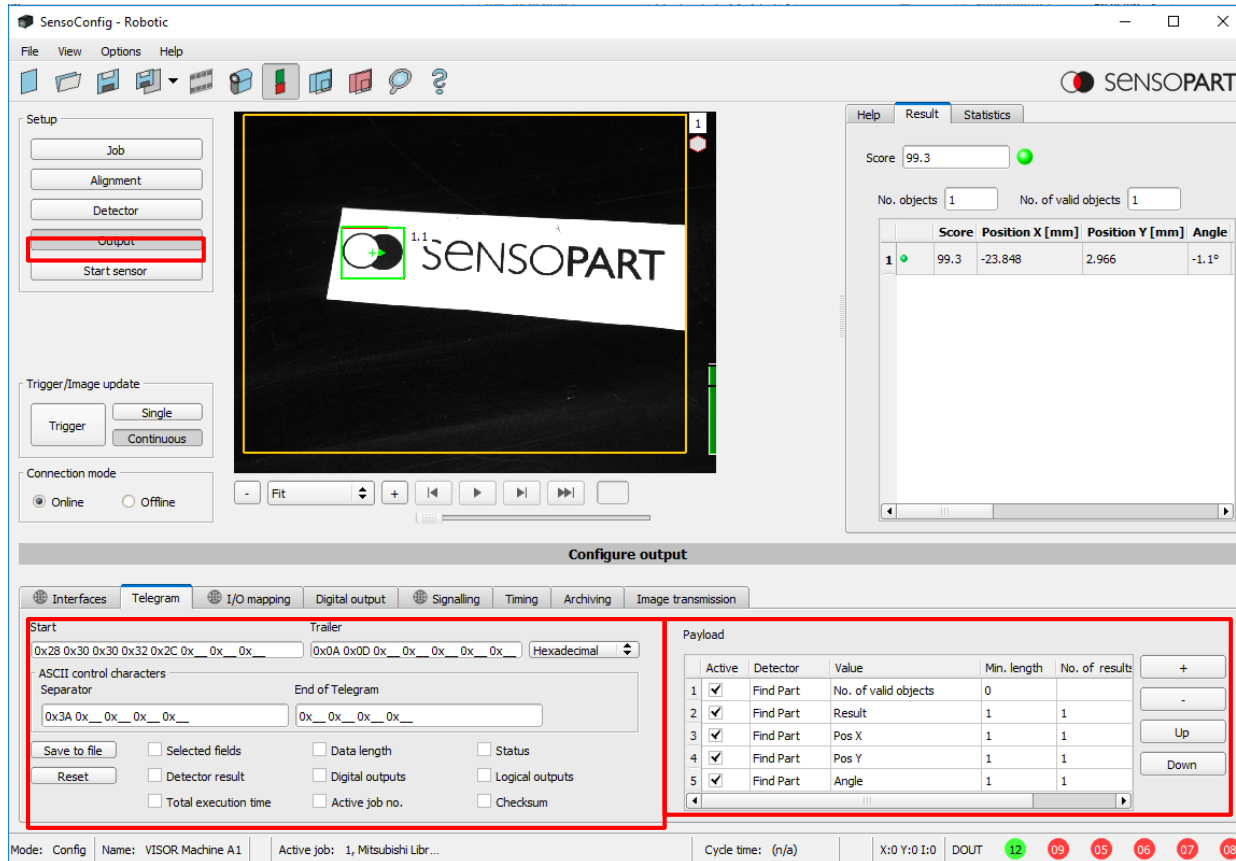
Setup VISOR[®] Robotic

- Activate the Ethernet interface for communication between VISOR[®] and robot controller



Setup VISOR[®] Robotic

- Define the values send to robot in step output.
- The robot controller calculates every thing else needed!



The screenshot displays the SensoConfig - Robotic software interface. The main window is divided into several sections:

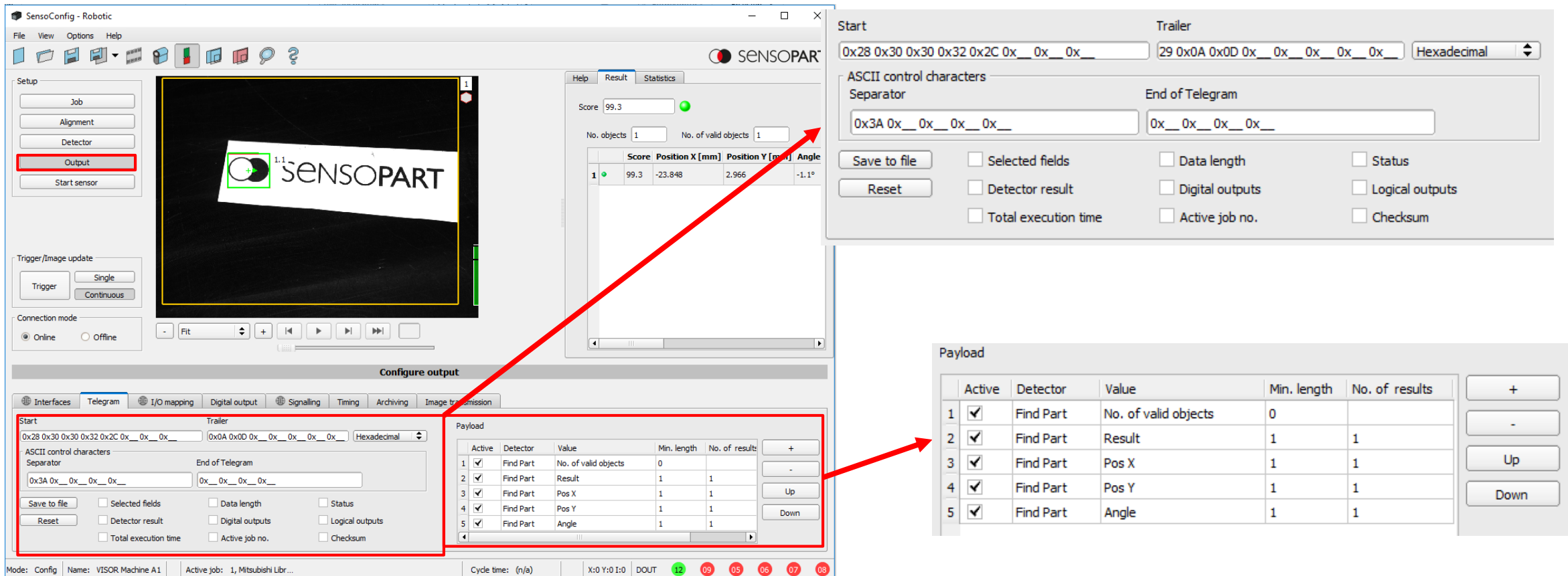
- Setup:** Includes buttons for Job, Alignment, Detector, Output (highlighted with a red box), and Start sensor.
- Trigger/Image update:** Includes a Trigger button and radio buttons for Single and Continuous.
- Connection mode:** Includes radio buttons for Online and Offline.
- Configure output:** This section is highlighted with a red box and contains:
 - Telegram:** Fields for Start (0x28 0x30 0x30 0x32 0x2C 0x__ 0x__ 0x__ 0x__), Trailer (0x0A 0x0D 0x__ 0x__ 0x__ 0x__ 0x__ 0x__), and Separator (0x3A 0x__ 0x__ 0x__ 0x__ 0x__). It also has checkboxes for Selected fields, Data length, Status, Detector result, Digital outputs, Logical outputs, Total execution time, Active job no., and Checksum.
 - Payload:** A table showing the output data structure. The table has columns: Active, Detector, Value, Min. length, and No. of result. The data is as follows:

Active	Detector	Value	Min. length	No. of result
1	Find Part	No. of valid objects	0	1
2	Find Part	Result	1	1
3	Find Part	Pos X	1	1
4	Find Part	Pos Y	1	1
5	Find Part	Angle	1	1

The bottom status bar shows: Mode: Config, Name: VISOR Machine A1, Active job: 1, Mitsubishi Libr..., Cycle time: (n/a), X:0 Y:0 I:0, DOUT, and a series of colored indicator lights.

Setup VISOR® Robotic

- It is important that Start / Trailer / Separator and End of Telegram are set like described here
- Make sure the first 5 elements of the payload are defined like described here



SensoConfig - Robotic

File View Options Help

Setup

Job

Alignment

Detector

Output

Start sensor

Trigger/Image update

Trigger

Single

Continuous

Connection mode

Online Offline

Fit

Help Result Statistics

Score: 99.3

No. objects: 1 No. of valid objects: 1

Score: Position X [mm] Position Y [mm] Angle

1 99.3 -23.848 2.966 -1.1°

Start Trailer

0x28 0x30 0x30 0x32 0x2C 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ Hexadecimal

29 0x0A 0x0D 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__

ASCII control characters

Separator: 0x3A 0x__ 0x__ 0x__ 0x__ 0x__

End of Telegram: 0x__ 0x__ 0x__ 0x__

Save to file

Reset

☐ Selected fields

☐ Detector result

☐ Total execution time

☐ Data length

☐ Digital outputs

☐ Active job no.

☐ Status

☐ Logical outputs

☐ Checksum

Configure output

Interfaces Telegram I/O mapping Digital output Signalling Timing Archiving Image transmission

Start Trailer

0x28 0x30 0x30 0x32 0x2C 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ Hexadecimal

0x0A 0x0D 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__ 0x__

ASCII control characters

Separator: 0x3A 0x__ 0x__ 0x__ 0x__ 0x__

End of Telegram: 0x__ 0x__ 0x__ 0x__

Save to file

Reset

☐ Selected fields

☐ Detector result

☐ Total execution time

☐ Data length

☐ Digital outputs

☐ Active job no.

☐ Status

☐ Logical outputs

☐ Checksum

Payload

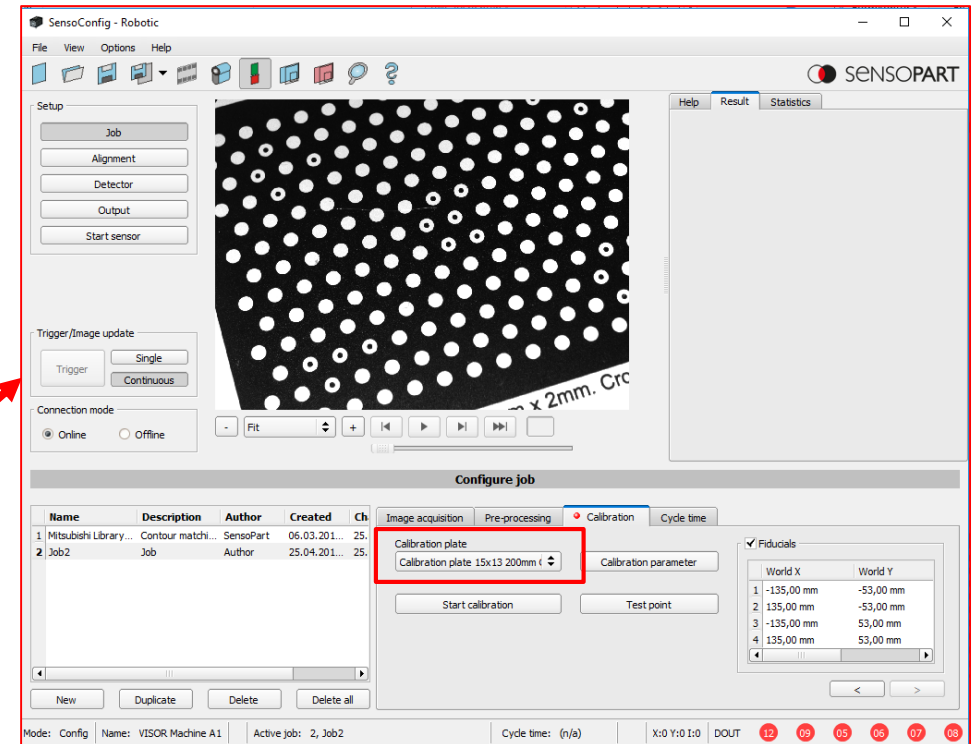
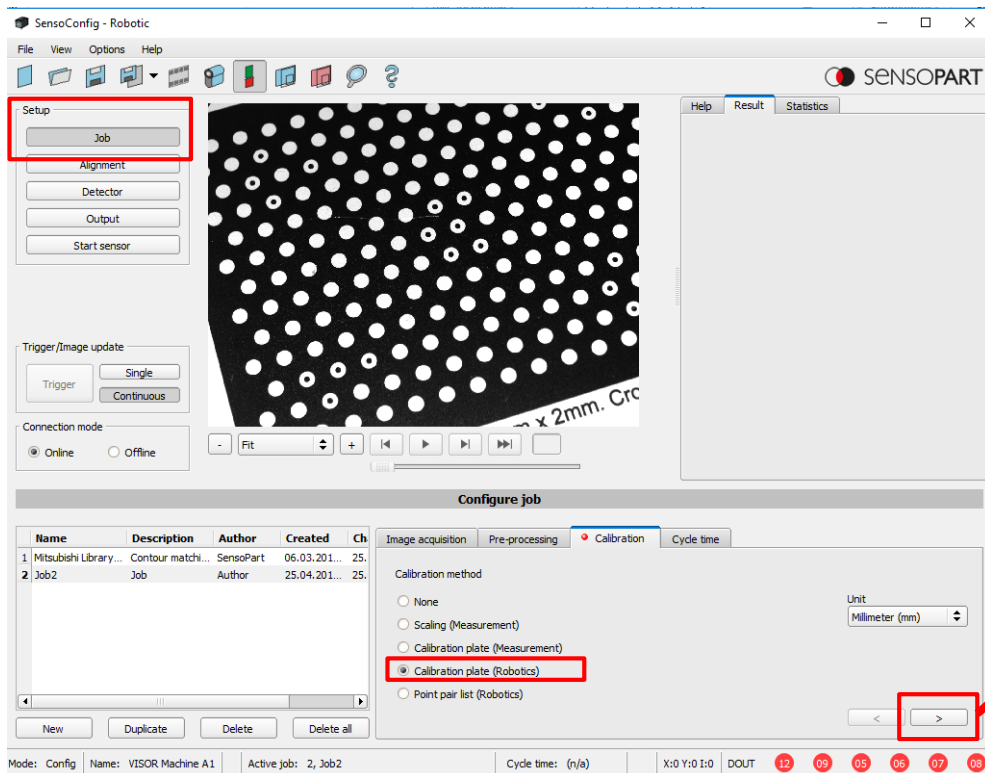
	Active	Detector	Value	Min. length	No. of results
1	<input checked="" type="checkbox"/>	Find Part	No. of valid objects	0	
2	<input checked="" type="checkbox"/>	Find Part	Result	1	1
3	<input checked="" type="checkbox"/>	Find Part	Pos X	1	1
4	<input checked="" type="checkbox"/>	Find Part	Pos Y	1	1
5	<input checked="" type="checkbox"/>	Find Part	Angle	1	1

Mode: Config Name: VISOR Machine A1 Active job: 1, Mitsubishi Libr... Cycle time: (n/a) X:0 Y:0 I:0 DOUT 12 09 05 06 07 08

Setup VISOR® Robotic

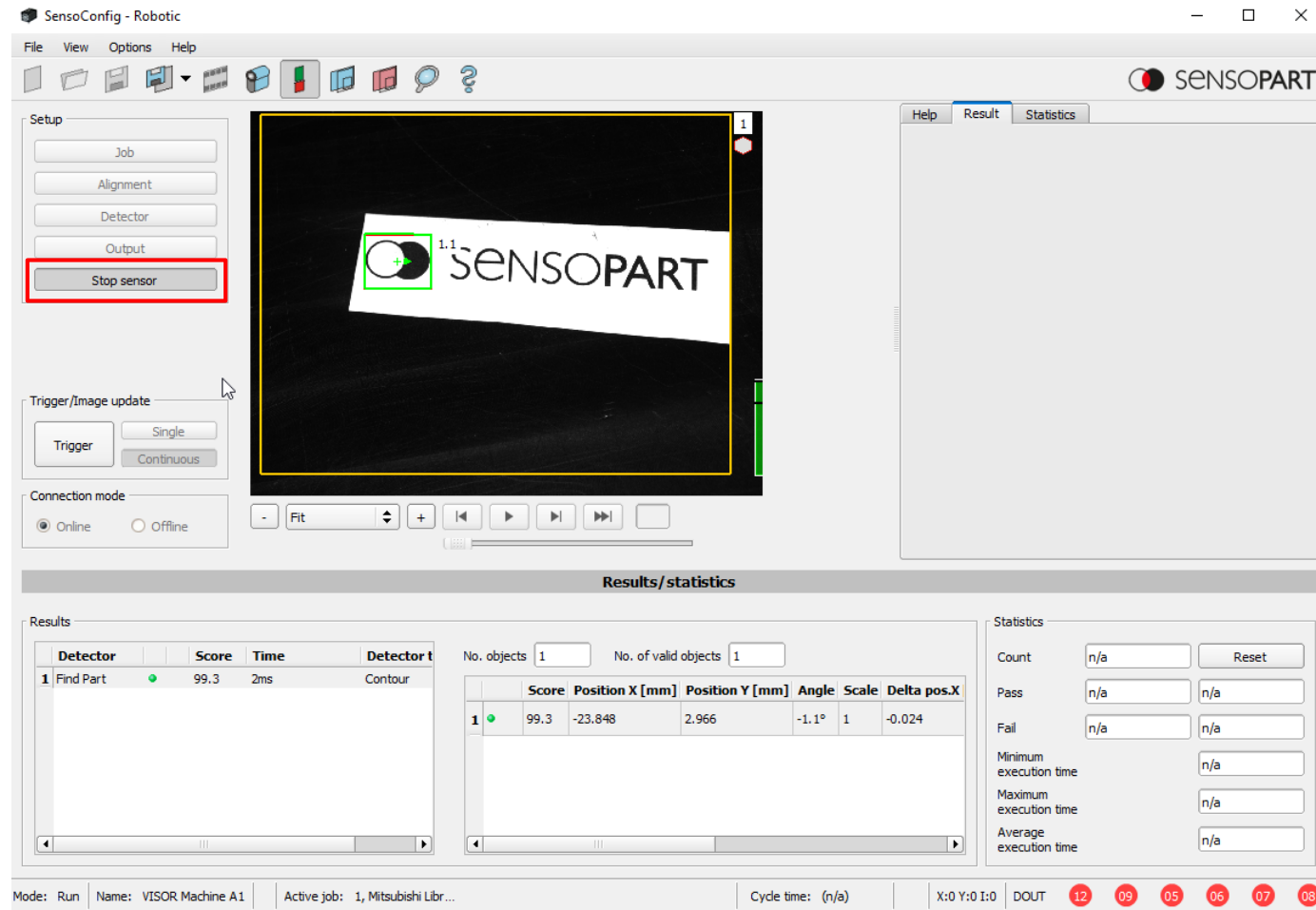
Go back to Job -> Calibration

- Select calibration method „Calibration plate (Robotic)”
- Select the right calibration plate model / size
- The calibration process itself is started by the robot program

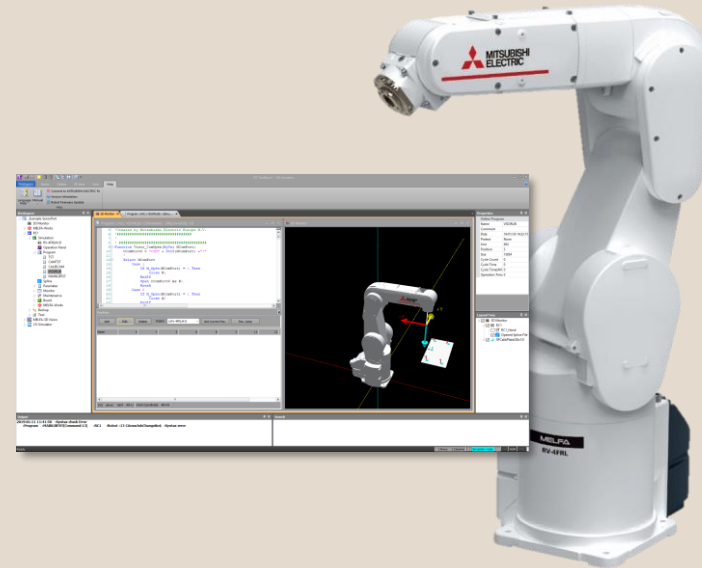


Setup VISOR[®] Robotic

- Start the VISOR[®] to switch to „run mode“ and download the configuration file to the device.

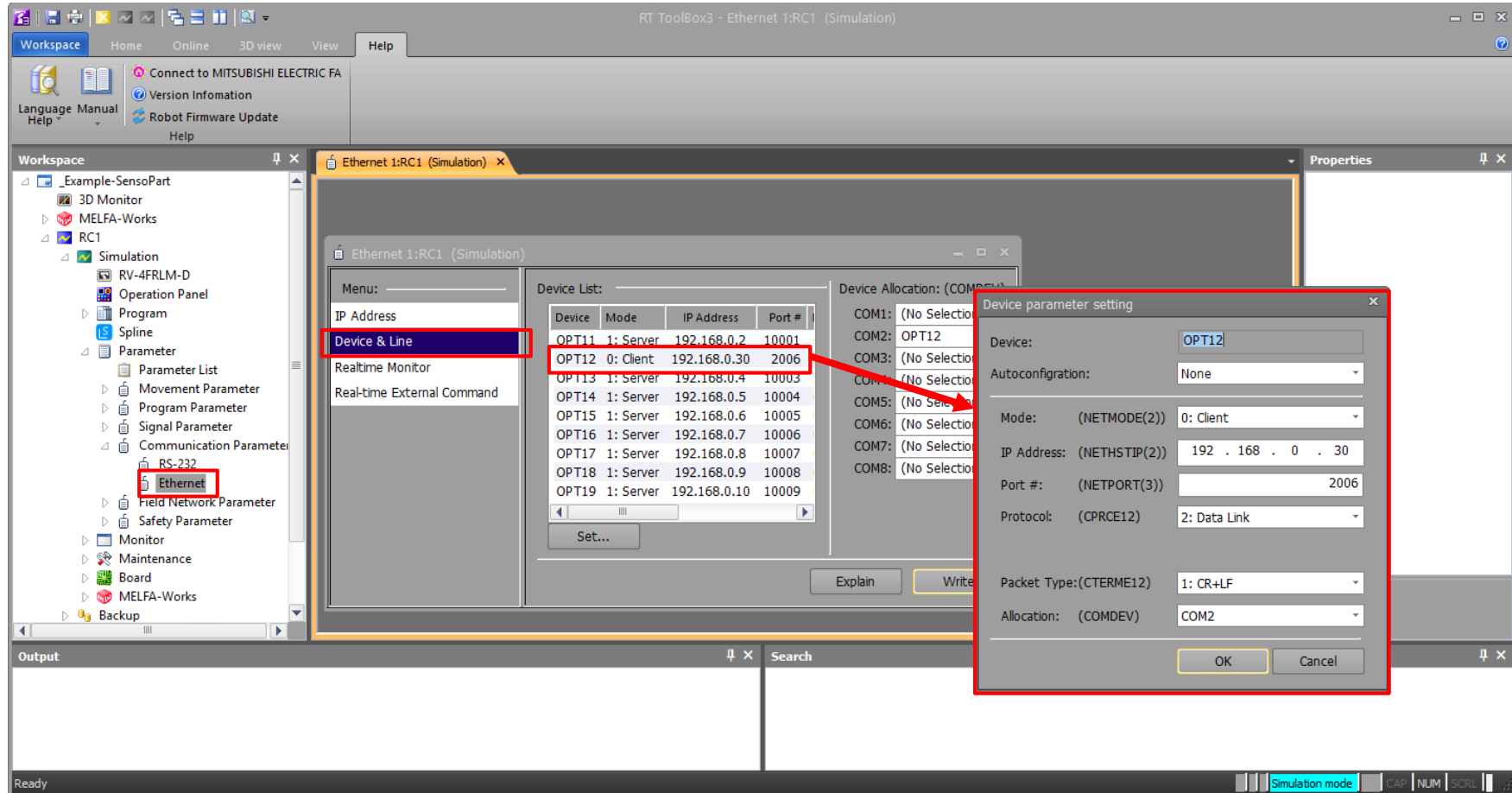


Setup robot



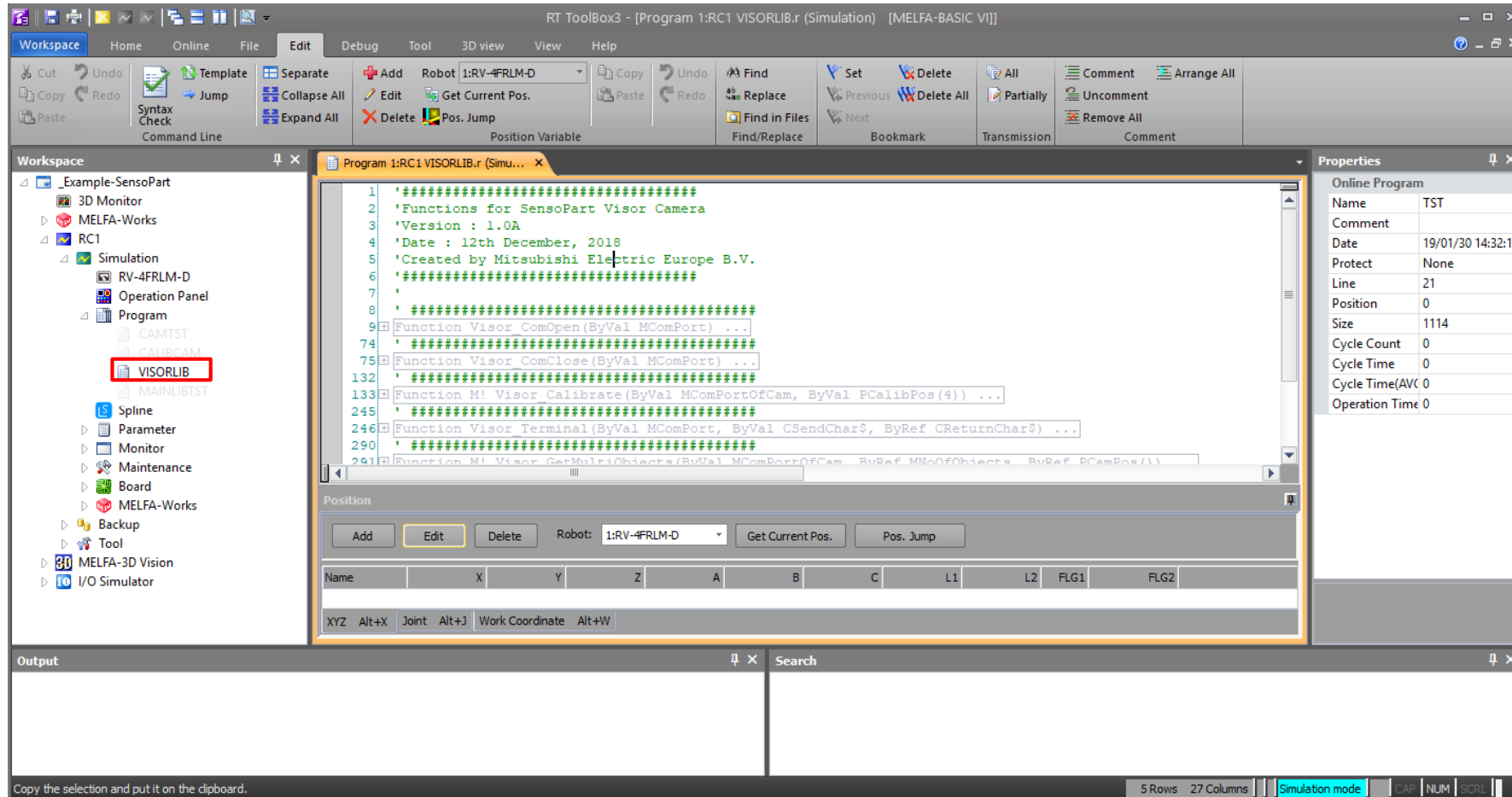
Setup robot

- Set up the communication interface



Setup robot

- Add the VISOR.lib to the libraries in the robot controller



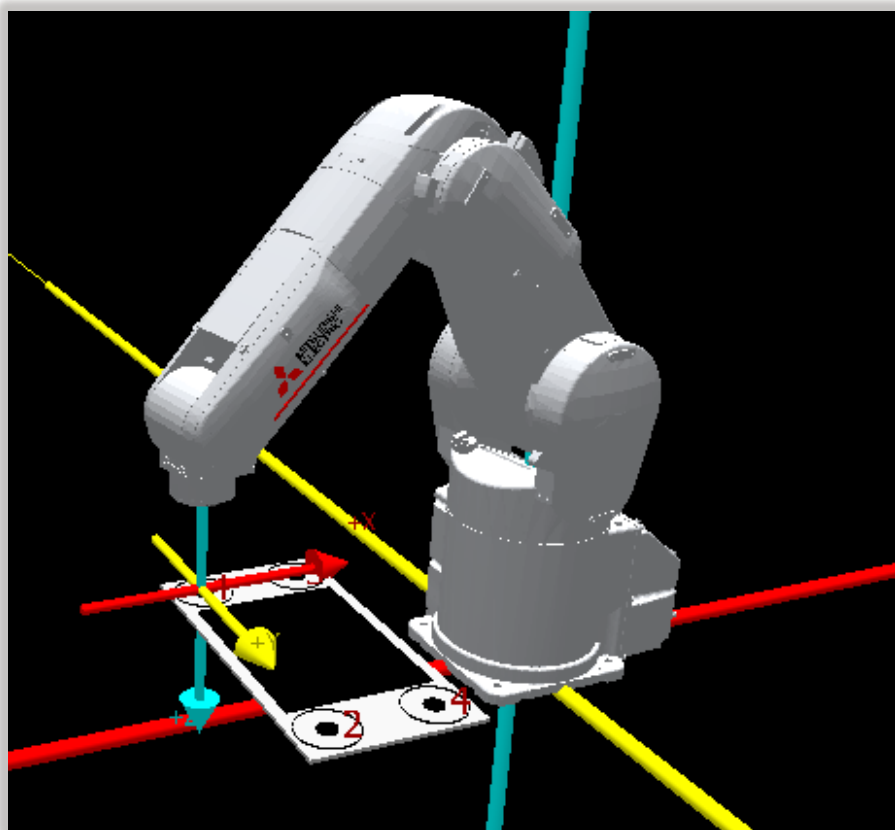
Program robot

- Function *Visor_Calibrate(<Com Port>, <Field with the calibration positions>)*
- Program example for calibration.

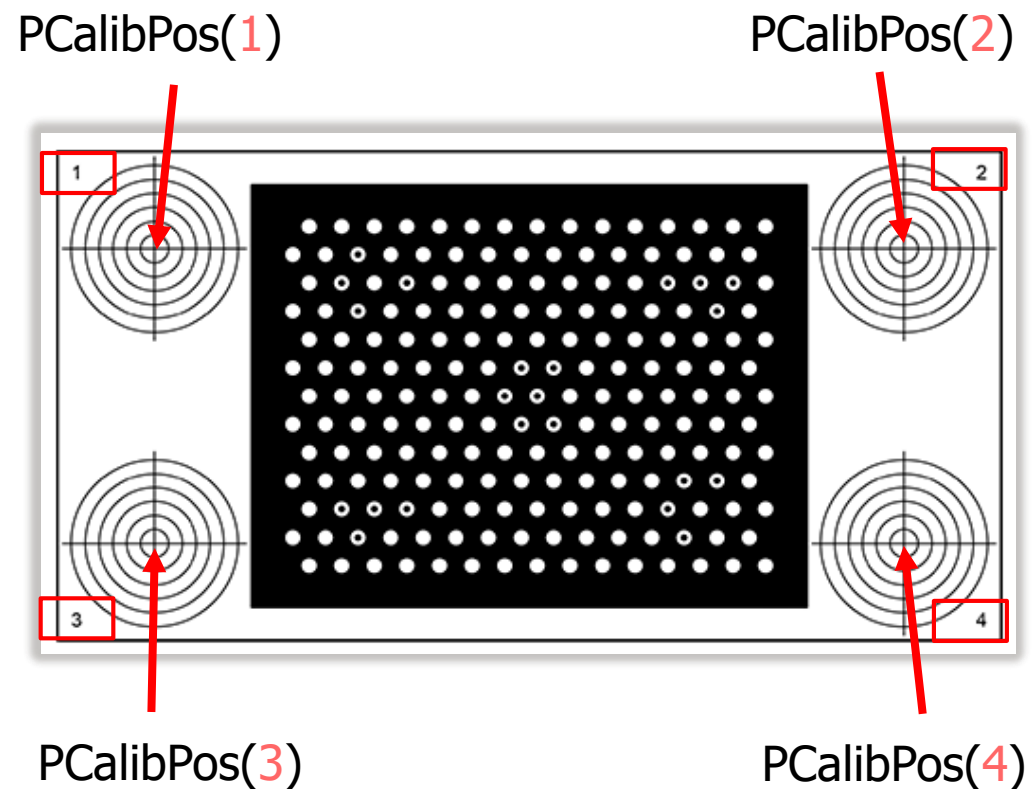
```
#Include "VISORLIB"  
Dim PCalibPos(4) ' Calibration position  
'  
  
MTemp = Visor_Calibrate(2, PCalibPos) ' start calibration  
If MTemp <> 1 Then ' calibration successful= 1  
    Error 9200  
    Hlt  
EndIf
```

Calibration

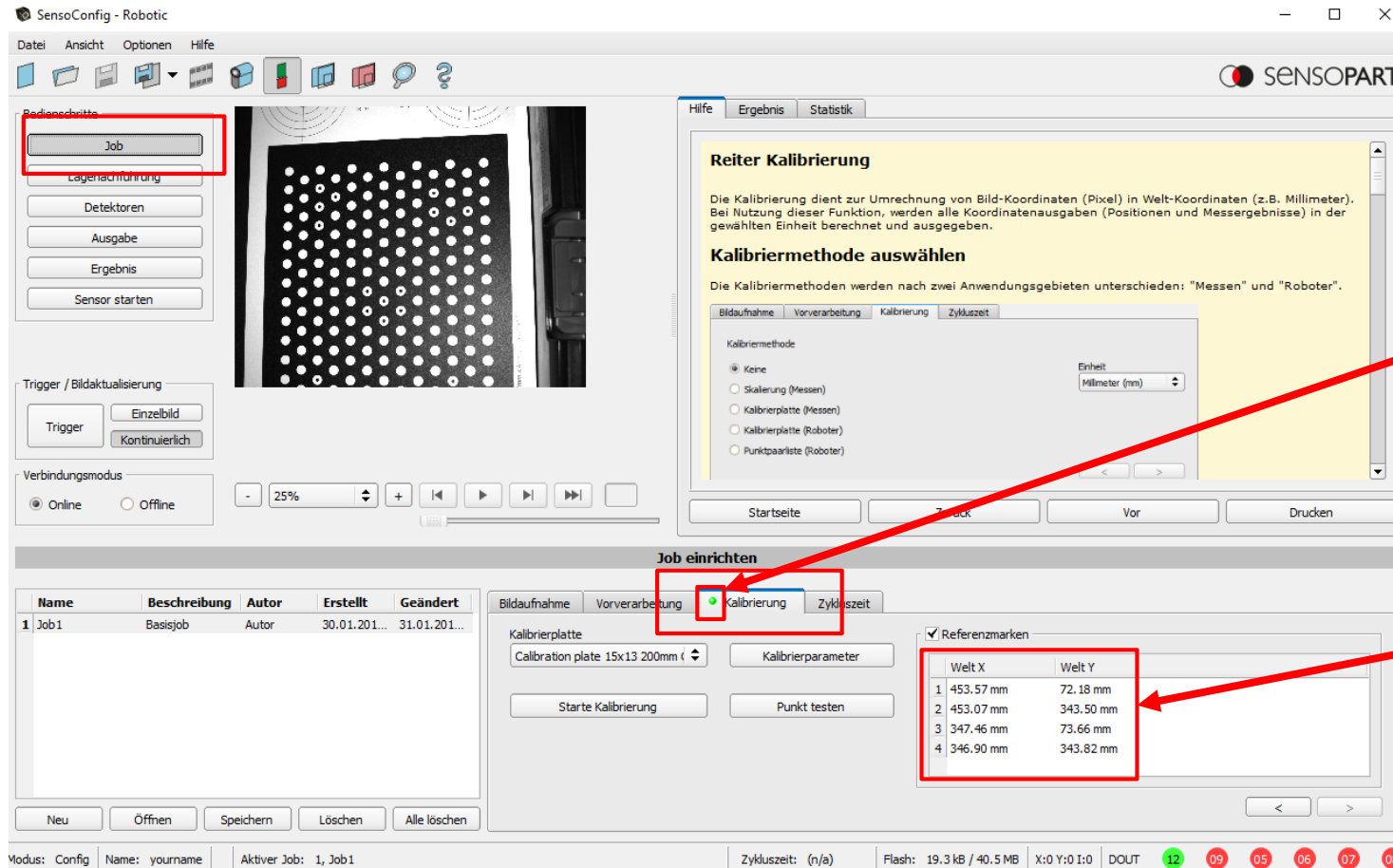
- Teach the calibration positions by moving the TCP over the four fiducials



e.g. Robot at teach position 1 (PCalibPos(1))



- Start calibration in the robot
- To check the calibration you can optionally use the VISOR[®] PC Software



LED coloring:

green : calibration valid : results are really precise

yellow : calibration valid with less accuracy

red : no valid calibration

These positions must exact be the Same like in the field of calibration positions

Object detection / pick an object

- Function

Visor_GetMultiObjects (<Com Port>, <number of objects>, <field with positions of the objects>)

- Example locate object – move robot there:

```
#Include "VISORLIB"
Visor_ComOpen(2)           ' open communication port
MTemp = Visor_GetMultiObjects(2, MPartsFound, PCamPos)
If MPartsFound > 0 Then    ' are valid objects available?
  For MCnt=1 To MPartsFound
    PPick = PBase          ' take over basis position (Z-height, orientation)
    PPick.X = PCamPos(MCnt).X
    PPick.Y = PCamPos(MCnt).Y
    PPick.C = PCamPos(MCnt).C
    Mov PPick
  '
  Mov PPlace
Next
EndIf
Visor_ComClose(2)         ' close communication port
```

Additional functions

- Function change job by **name**

Visor_JobChangeByName(<Com Port>, <Job name>)

e.g.:

MTemp = Visor_JobChangeByName(2, „Job1“)

- Function change job by **number**

Visor_JobChangeByNumber(<Com Port>, <Job number>)

e.g.:

MTemp = Visor_JobChangeByNumber(2, MJobNumber)

