

ABB Cognex TCP/IP



Networking Overview

Connect the Cognex camera to a network switch that can be accessed via both a computer and the robot.

- Note. The robot cannot communicate via TCP/IP on LAN 3 if the robot is setup to be an Ethernet/IP slave to a computer.
- This means the LAN ports or WAN port can be used.
- Set the IP address of the camera according to what port you use

Main elements of the COGNEX "In-Sight" vision interface:

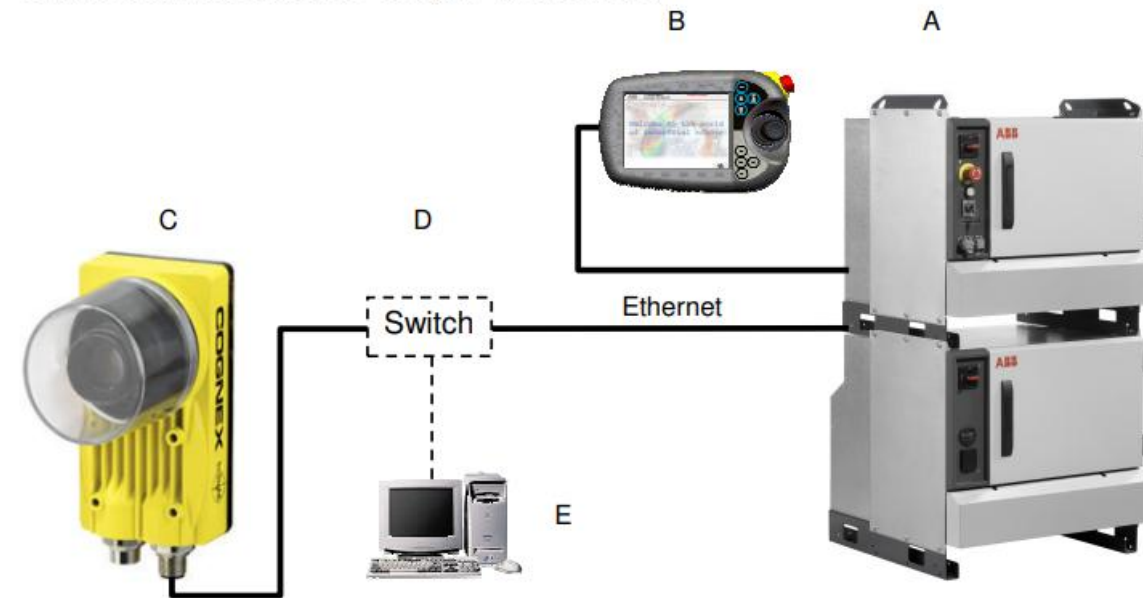


Figure 1. Hardware architecture with an IRC5 control system

A	IRC5 Control System
B	"FlexPendant" unit
C	"In-Sight" smart camera (5000 series)
D	Ethernet hub or "switch"
E	PC compatible computer

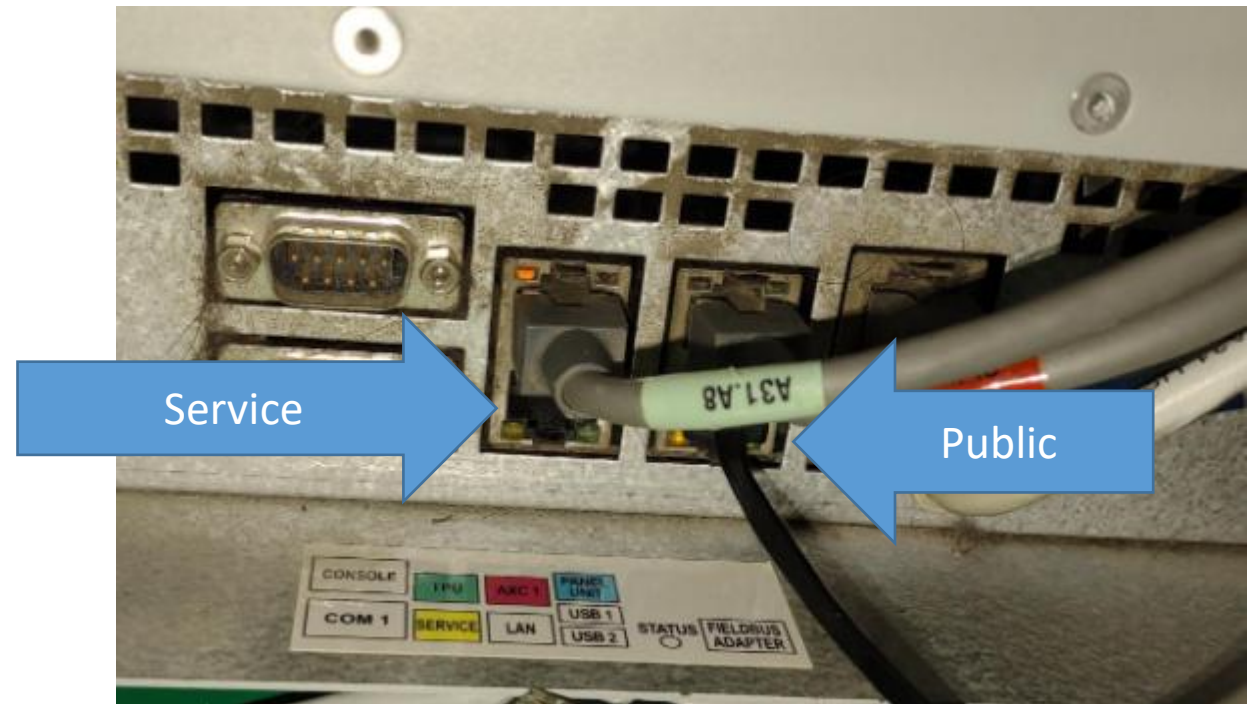
Networking Options

If the switch is connected to the service port, camera and computer must be on the 192.168.125.xxx subnet

The IP address of the Lan port can change based on the network it is connected to/What the controller is set to

Modern ABB's refer to the Lan/public port as a WAN port

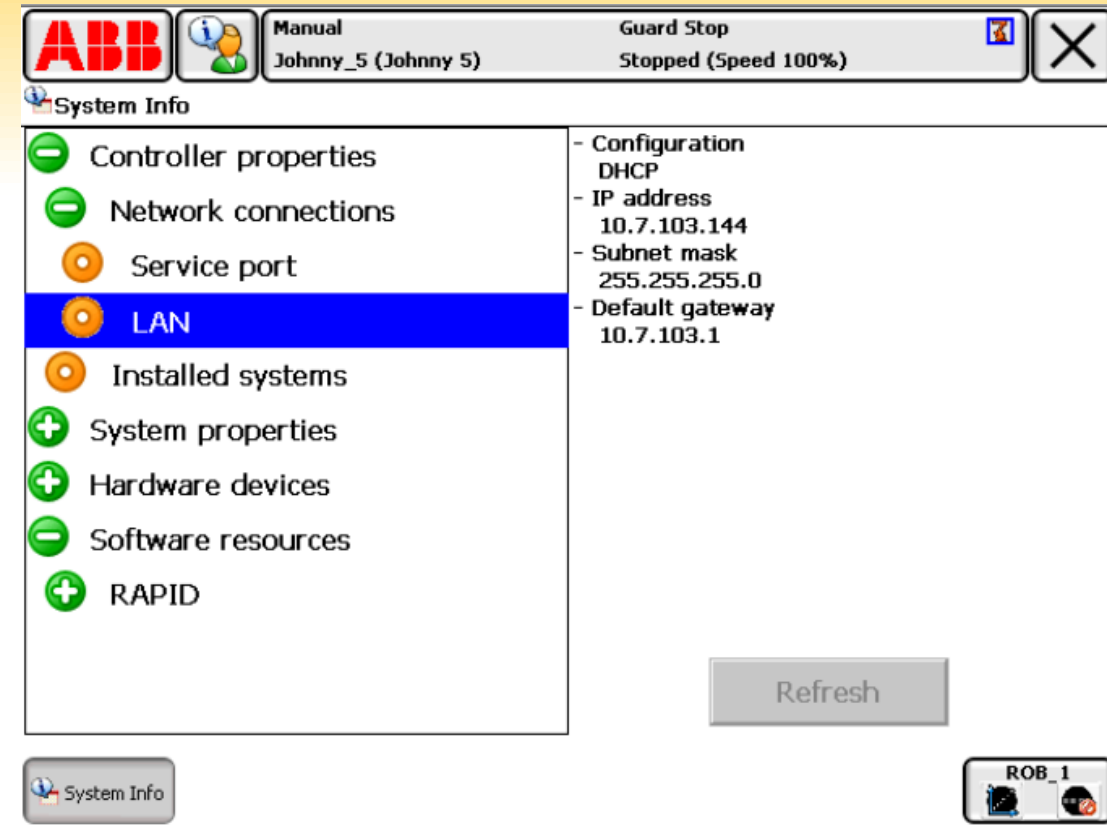
All Lan ports on modern ABB's can be setup for the local network



Networking Public/Wan

If connecting to the robot via the public port, the IP address can be found on the robot under the system information screen similar to the image shown to the right.

When the public port is set to DHCP the robot must be plugged into the network before being turned on, otherwise it will not receive an IP Address

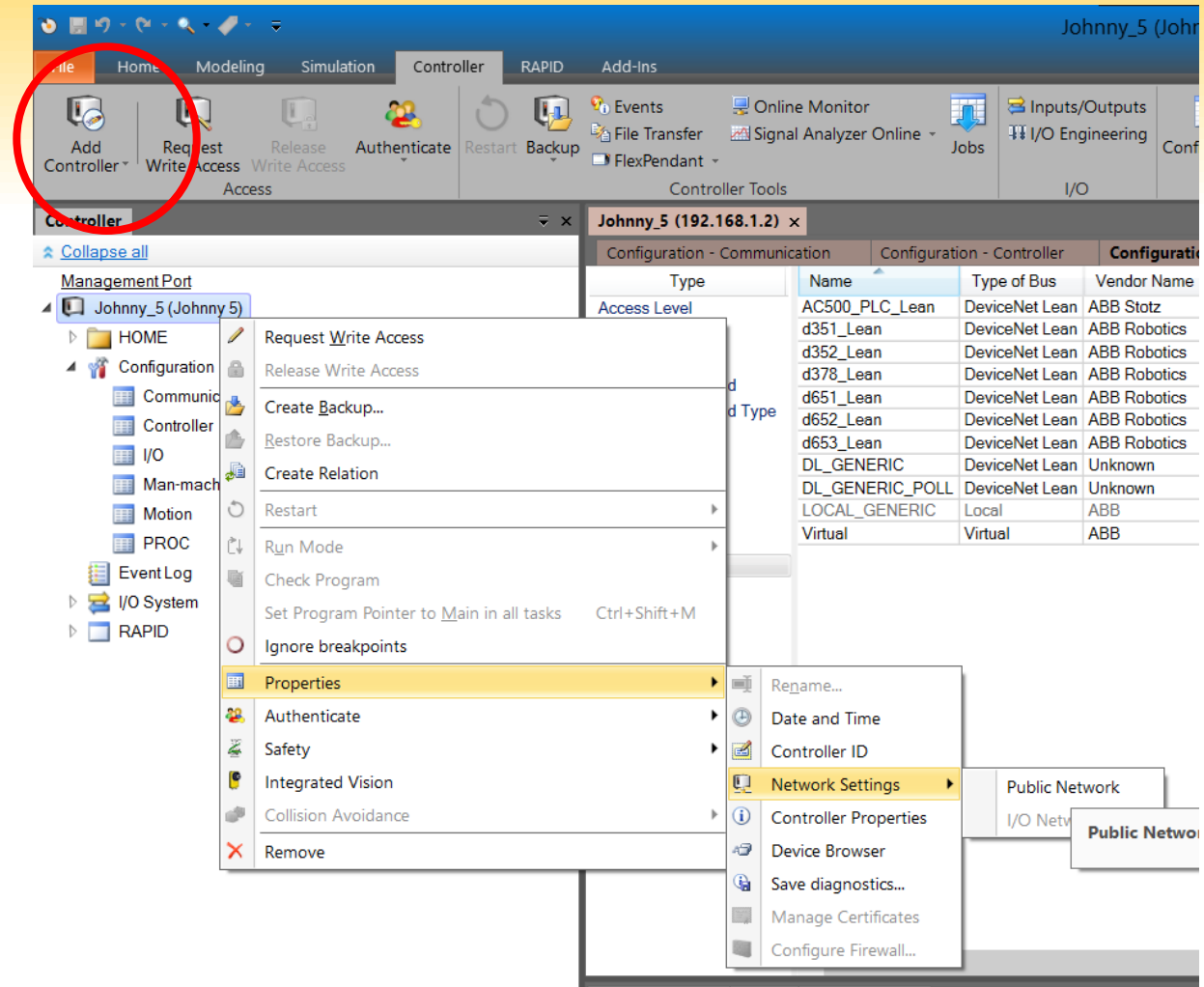


Networking Public/Wan

To change the network from DHCP to static, it is easiest to do so through the service port on the robot

Connect to the robot using the one click connect button by the add controller

Right click on the controller and select properties>Network settings>Public Network

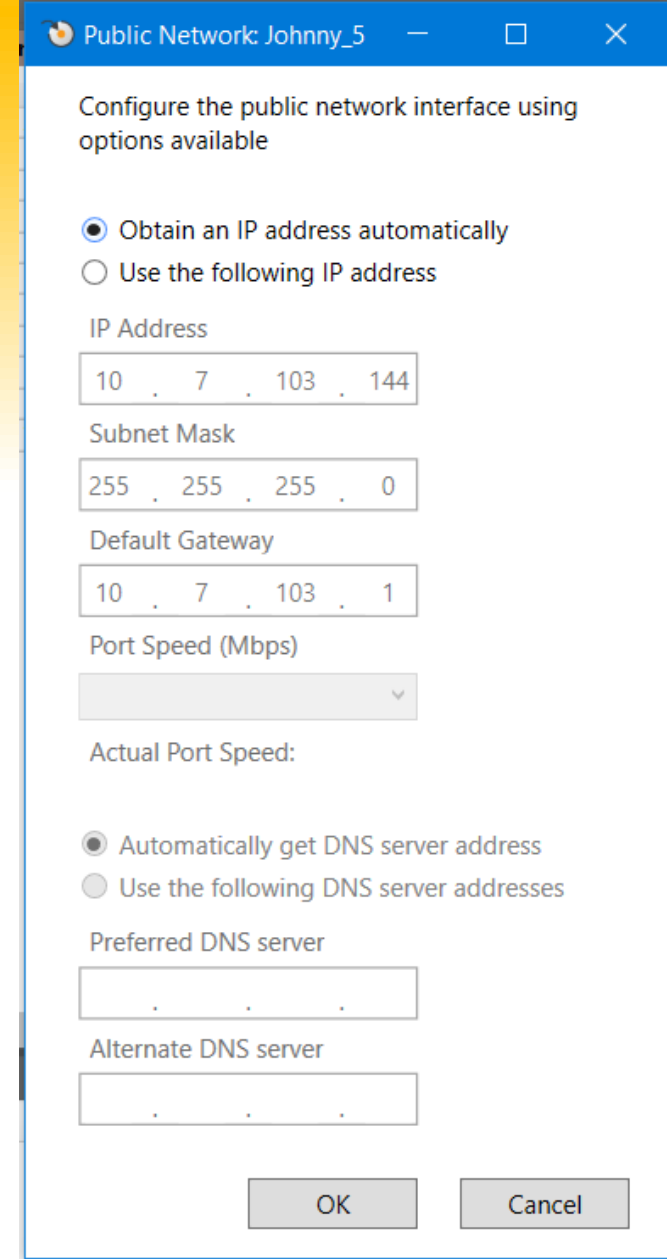


Networking Public/Wan

Change the settings on the screen to operate with your network

A PLC network will generally need a static IP address while an internet network will need a DHCP address

This will be controlled by your IT departement



The screenshot shows the Windows Network Setup window for a public network named 'Johnny_5'. The window title bar includes standard Windows window controls (minimize, maximize, close). The main heading is 'Configure the public network interface using options available'. There are two radio button options: 'Obtain an IP address automatically' (which is selected) and 'Use the following IP address'. Below these are input fields for 'IP Address' (10 . 7 . 103 . 144), 'Subnet Mask' (255 . 255 . 255 . 0), and 'Default Gateway' (10 . 7 . 103 . 1). There is a 'Port Speed (Mbps)' dropdown menu and a label for 'Actual Port Speed:'. Further down, there are two radio button options for DNS: 'Automatically get DNS server address' (selected) and 'Use the following DNS server addresses'. Below these are input fields for 'Preferred DNS server' and 'Alternate DNS server', both showing placeholder dots. At the bottom right are 'OK' and 'Cancel' buttons.

Public Network: Johnny_5

Configure the public network interface using options available

☒ Obtain an IP address automatically
☐ Use the following IP address

IP Address
10 . 7 . 103 . 144

Subnet Mask
255 . 255 . 255 . 0

Default Gateway
10 . 7 . 103 . 1

Port Speed (Mbps)
▼

Actual Port Speed:

☒ Automatically get DNS server address
☐ Use the following DNS server addresses

Preferred DNS server
. . .

Alternate DNS server
. . .

OK Cancel

Robot Modules

Download the ABB Sample code from Cognex's website

- <https://support.cognex.com/en/downloads/detail/in-sight/554/1033>
- Code is also available for ABB, Fanuc, and Denso robots

Extract the downloaded file and load the following modules onto the robot

- INSIGHT.mod, INSIGHT_ENG.mod
- Under the example program, you can also load the TESTVISION.mod
- You can upload the .sys version of the modules and they will be loaded as system modules. System modules are not deleted when a new program is loaded

Loading Modules

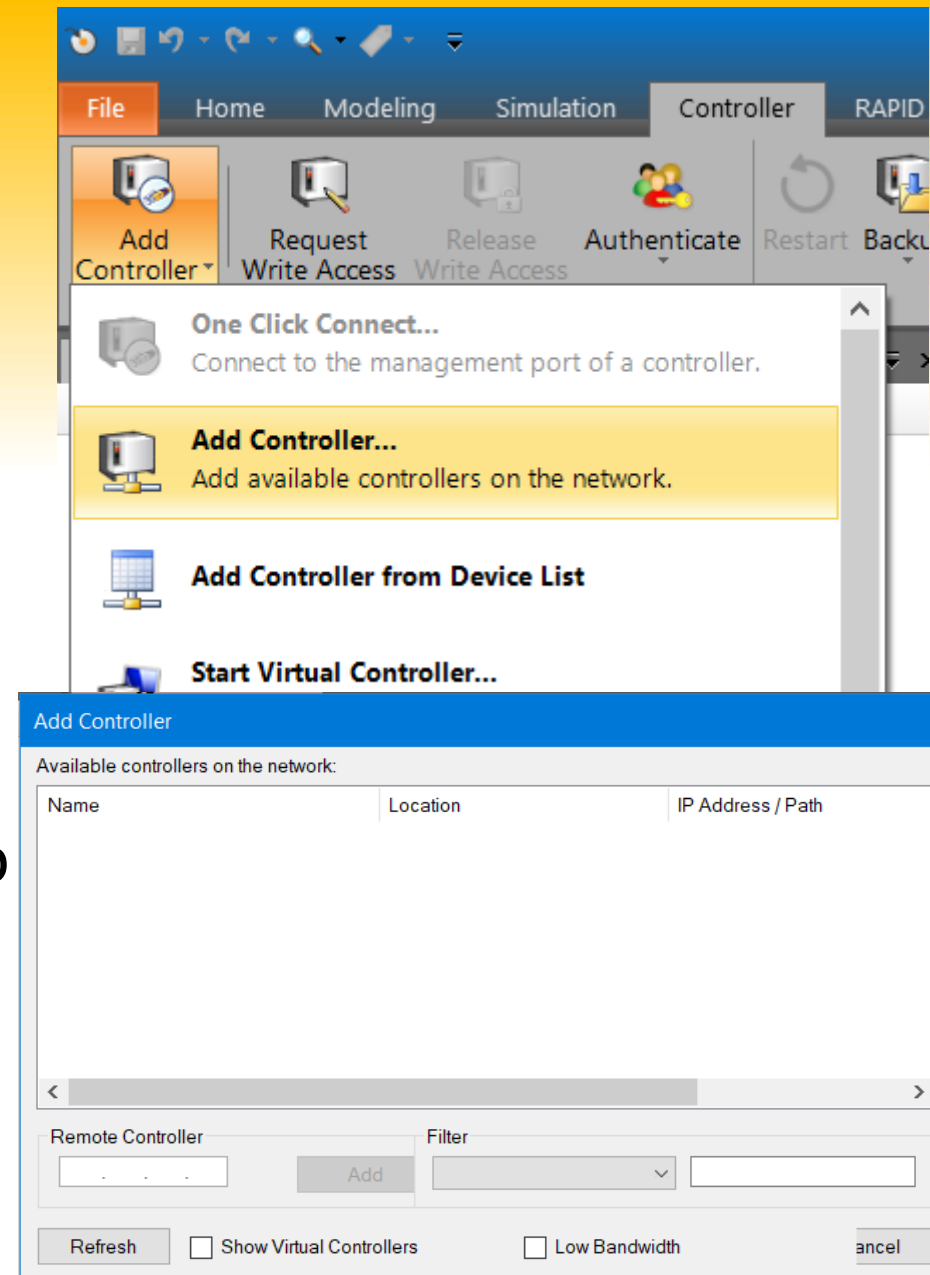
Modules can be loaded to the robot via a USB stick or over the network. The preferred method is over the network.

Connect to the robot using robotstudio

If the switch is connected to the service port, use the ip address 192.168.125.1 to connect to the robot.

- One click connect will also work

If using the public port, use the add controller tab to type in the IP address and connect

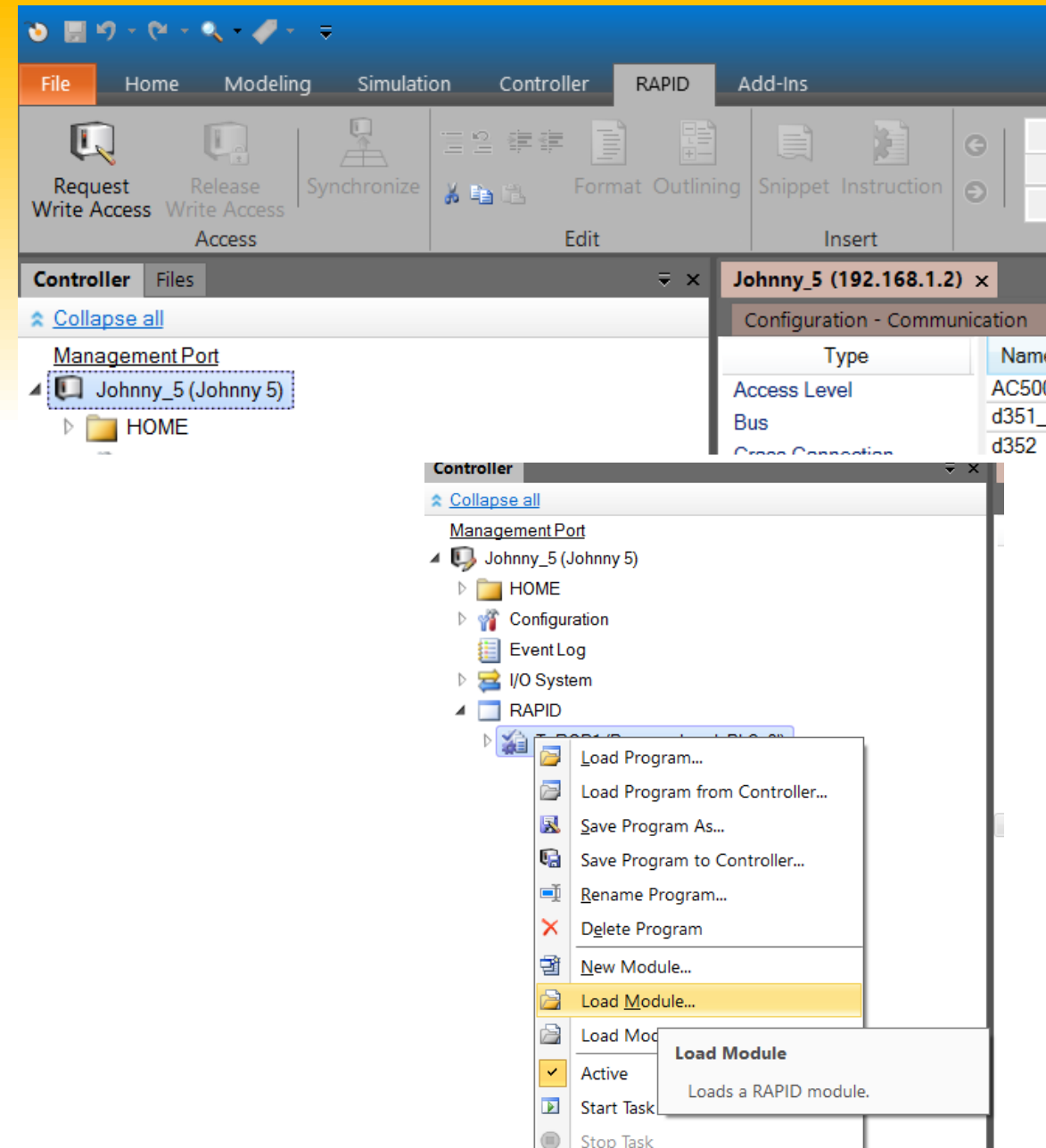


Loading Modules

Request write access to the robot using the rapid tab and hit request write access

- This will prompt the teach pendant, hit Grant
- If the robot is in Auto mode, it may automatically give access depending on the safety setup of the robot

Right click the task under the controller and select load module



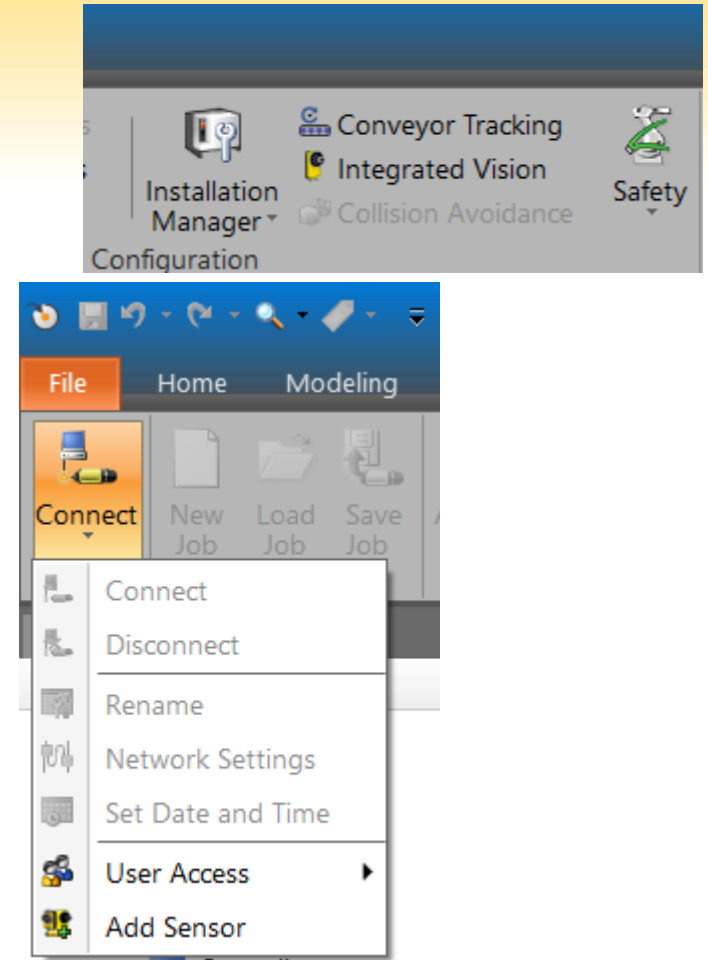
Camera Setup (RobotStudio)

Because ABB is partnered with Cognex, you can connect to Cognex cameras directly through ABB's software.

Under the controller tab, click the integrated vision button

This will open the integrated vision tab where you can add and connect to sensors

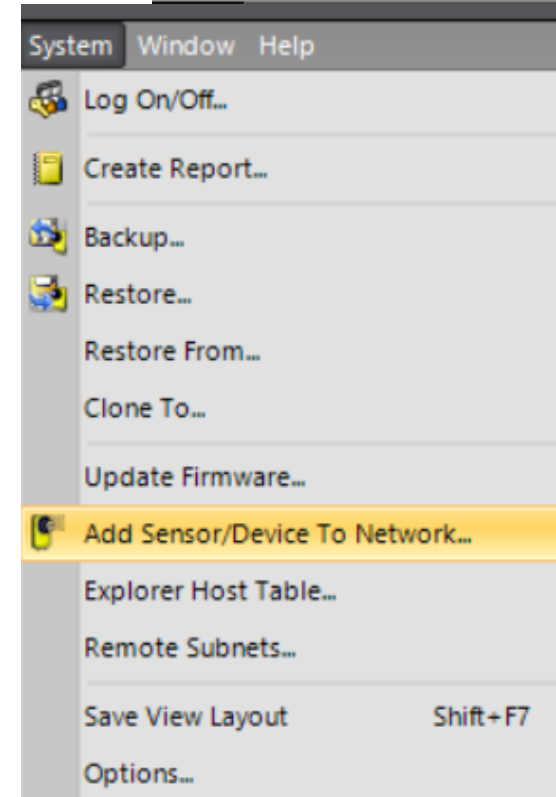
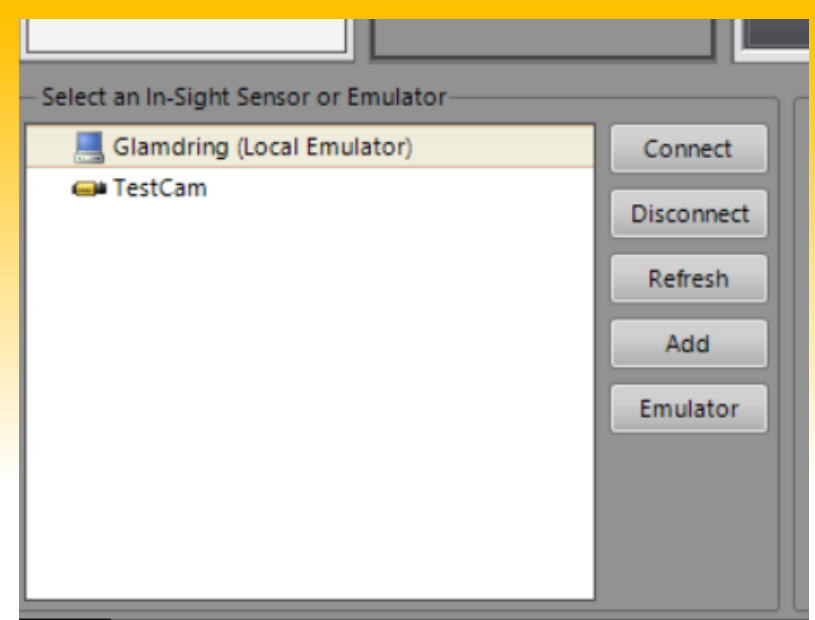
- This will only work with robots that have the integrated vision package



Camera Setup (Insight)

Open Cognex's Insight explorer software

If no cameras show up at the bottom left of the screen, verify that the camera is connected to the network switch properly and proceed to the add sensor button under the system tab

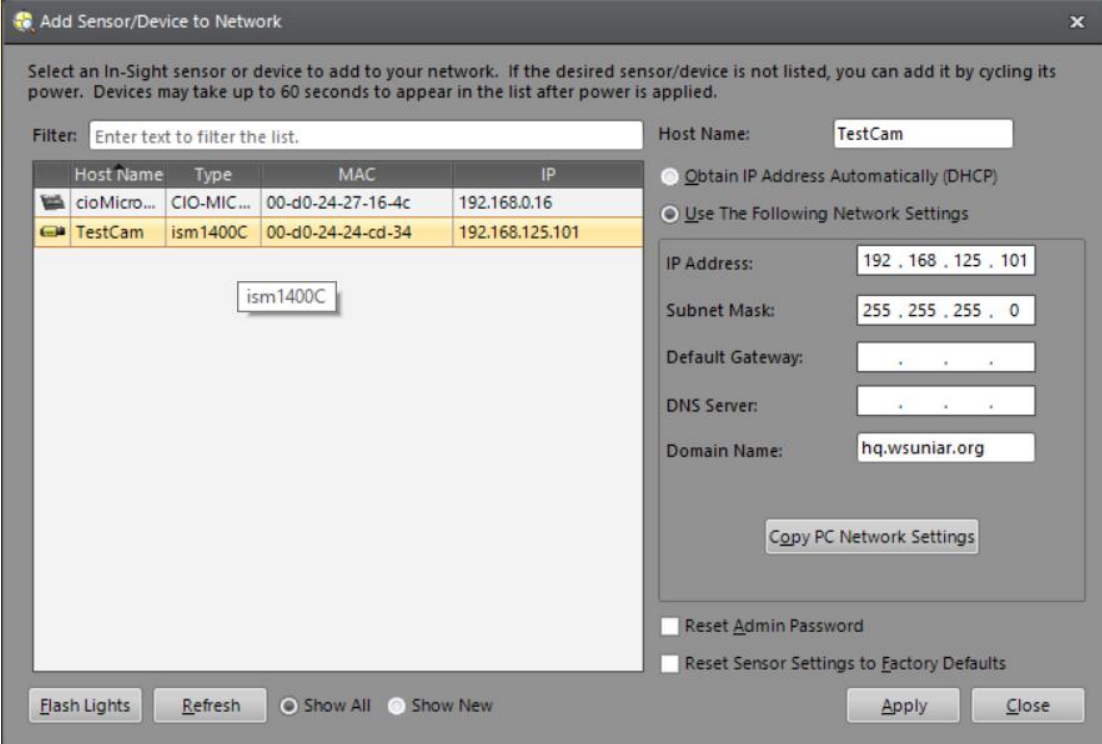


Camera Setup (Insight)

In the add sensor window, select the camera you will be working with and set the IP address to match the subnet of the network you are on

This is also where the name of the camera can be set

- If the camera asks for a password, the username is “admin” and the password is blank
- Ask an administrator if that does not work.



The screenshot shows a window titled "Add Sensor/Device to Network". It contains a table of devices and a form for network configuration.

Host Name	Type	MAC	IP
cioMicro...	CIO-MIC...	00-d0-24-27-16-4c	192.168.0.16
TestCam	ism1400C	00-d0-24-24-cd-34	192.168.125.101

Below the table, a search filter is set to "ism1400C".

On the right, the "Host Name" is set to "TestCam". The "Obtain IP Address Automatically (DHCP)" option is selected. The "Use The Following Network Settings" section is expanded, showing the following values:

- IP Address: 192 , 168 , 125 , 101
- Subnet Mask: 255 , 255 , 255 , 0
- Default Gateway: . , . , .
- DNS Server: . , . , .
- Domain Name: hq.wsuniar.org

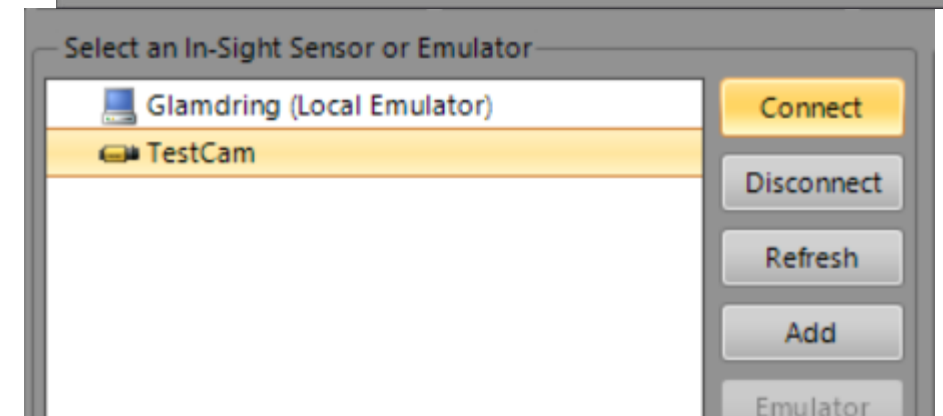
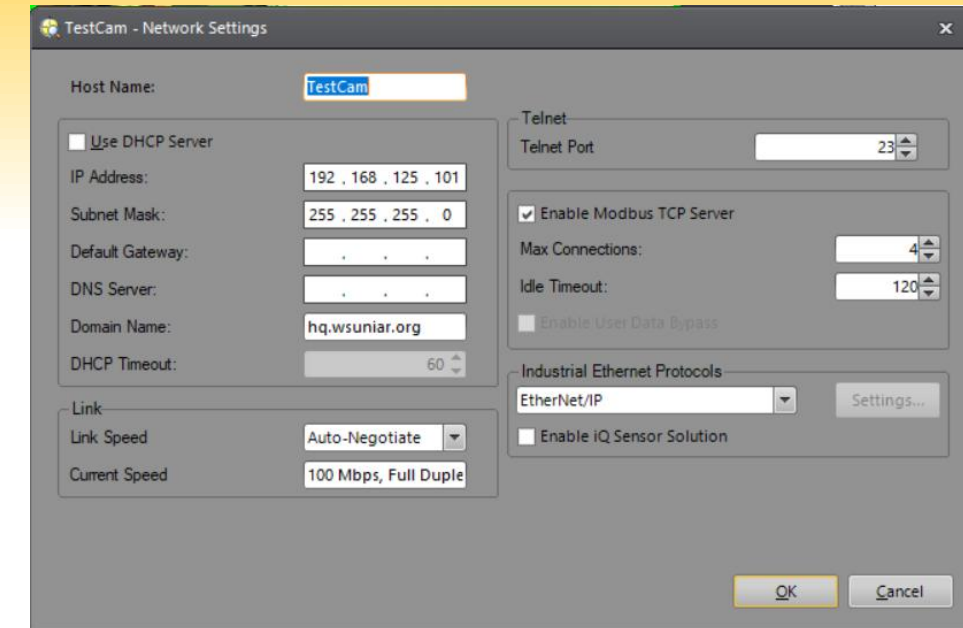
There is a "Copy PC Network Settings" button. At the bottom, there are checkboxes for "Reset Admin Password" and "Reset Sensor Settings to Factory Defaults", and "Apply" and "Close" buttons.

Camera Setup (Insight)

Connect to the camera using the connect option in the bottom left
Verify the camera tcp telnet port is set to 23 under the cameras network settings

- This is the standard for telnet communication

Reference previous lessons covering Cognex machine vision to setup a program that can locate parts



Camera Calibration

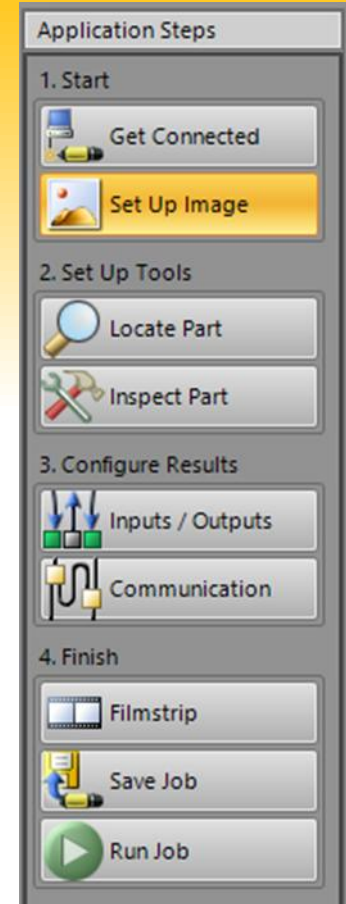
Once the camera has been connected, proceed to the setup tab

Using the Acquisition settings in the bottom left change the following settings

- Trigger: Network
- Exposure: Test with background and controlled lighting
 - Autoexposure may be acceptable

Check the lens on the physical camera to ensure that the aperture and focus are set correctly

Using the live video function can aid in setting the aperture and focus

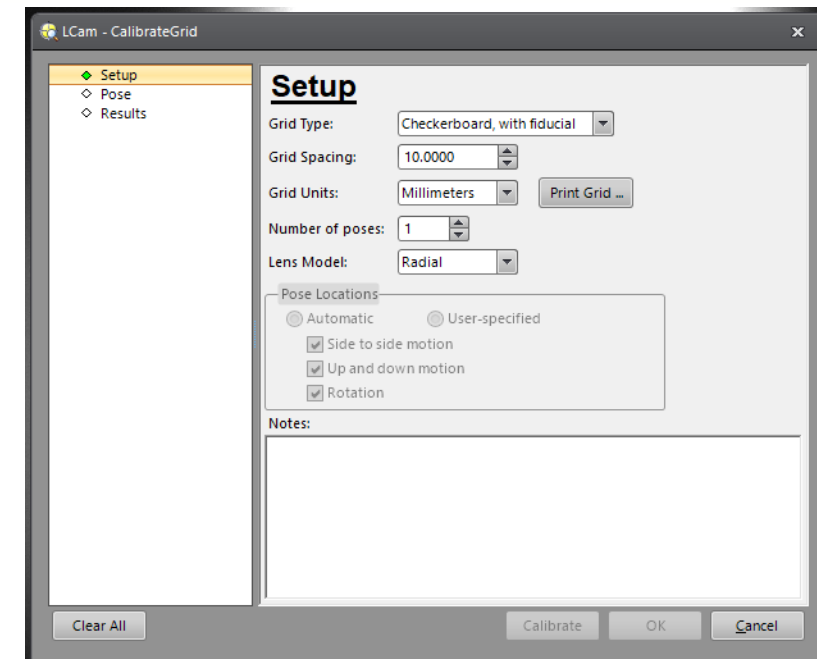
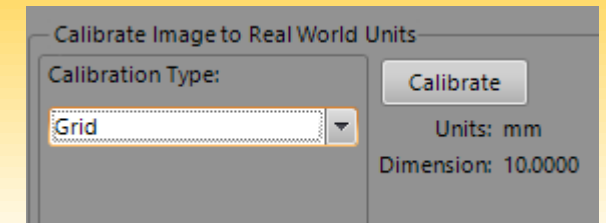


Camera Calibration

On the bottom of the screen will be a drop down requesting calibration type, select grid and hit calibrate

This will bring up the window to the right
Make sure “Checkerboard, with fiducial” is selected

Set the grid spacing to the smallest value that the camera will be able to accurately see. This will also be affected by the accuracy of the printed used to make the sheet.



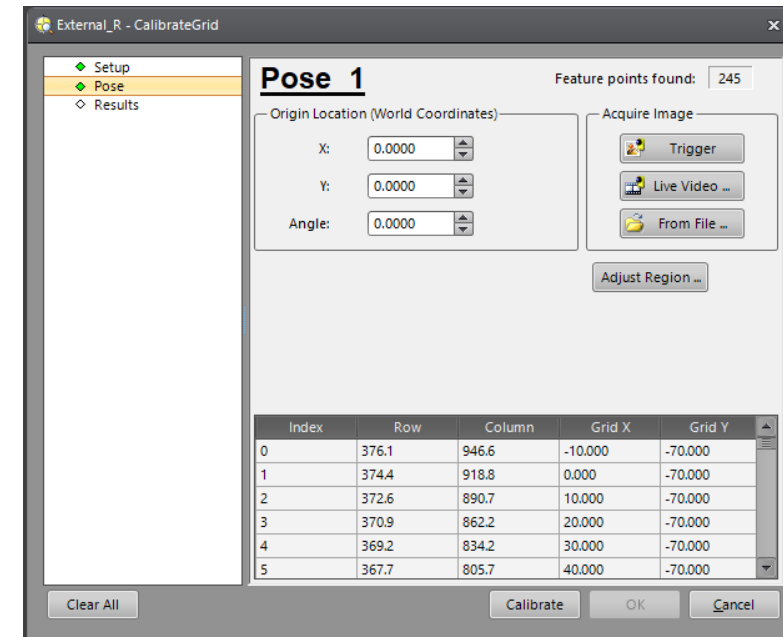
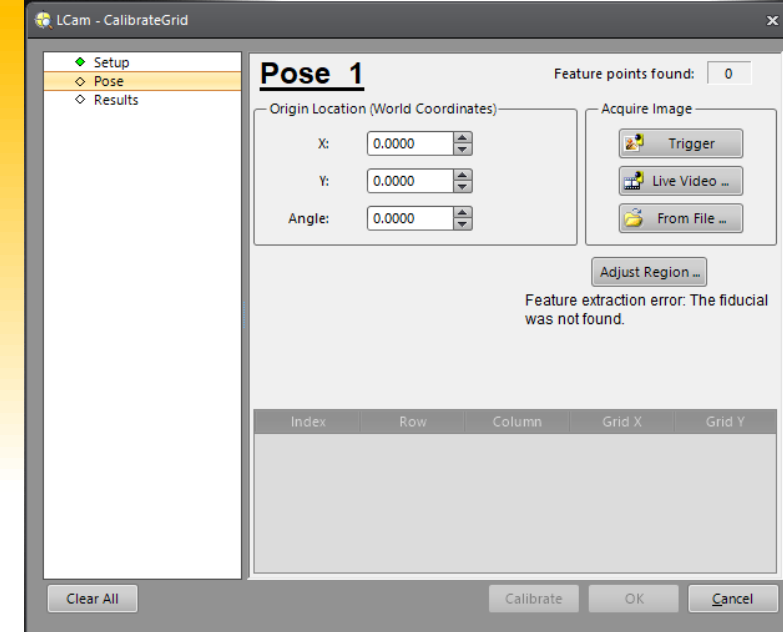
Camera Calibration

Once you have printed the sheet, place the sheet approximately in the middle of the image

- It is important that this paper does not move and is as flat as possible
- Create a point using a non moving work object and tool 0
- This point will determine the nominal z offset of the camera

Acquire a new image

Hit calibrate at the bottom of the window

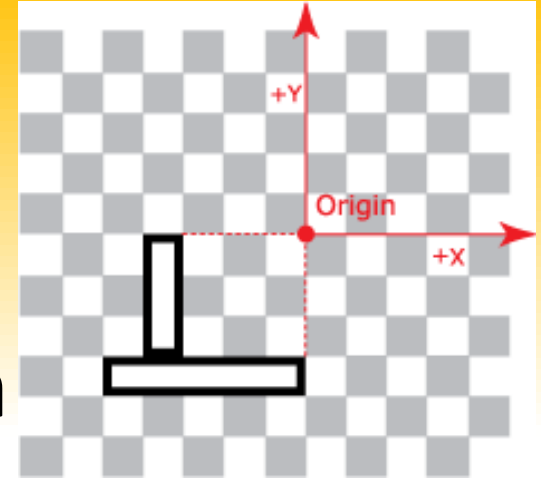


Robot Alignment

Teach a work object on the coordinate system of the frame of the fiducial

- If using an ABB robot make sure to use the Uframe and set the Oframe to identity $[[0,0,0],[1,0,0,0]]$
- Make sure to save the job at this step so the calibration is saved onto the camera

An accurate TCP is required for this step



Robot Commands

The key commands used on the robot are listed below

- CX_SetupCamera
- CX_InitComm
- CX_GetOnline
- CX_SetOnline
- CX_GetFile
- CX_LoadFile
- CX_TriggerImage
- CX_GetVisionData
- CX_ShowErrStatus

Concept

The robot pulls individual cells out of the spread sheet view of the Cognex

This means after running the trigger command on the robot, it will pull from the specified cell in the spreadsheet view

```
! Reading vision data in 1 ABB formatted cell / Lecture des données dans 1 cellule au format ABB
IF CX_TriggImage(nActiveCam,nErrStatus) THEN
    IF NOT CX_GetVisionData(nActiveCam,"C",26,\Read4Cells,tVisionTranfObj1,nVisionScoreObj1,nErrStatus) THEN
        CX_ShowErrStatus nErrStatus;
        RETURN ;
    ENDIF
ELSE
```

Flow

The robot setups a camera to communicate

- Up to 4 cameras can be connected at once

The camera communication is initiated

The camera is set in offline mode so a job can be loaded

The job is loaded

The camera is set in online mode

The camera is triggered

Data is read from the camera to the robot

Robot Setup Camera

Using the CX_SetupCamera command the robot give the camera

- An Identity number that is used to refer to the camera
- IP address of the camera
- port number communication is done on
- User name for logging onto the camera
- Camera password
- Minimum and maximum values excepted for coordinates
- Optionally a timeout value for if the robot fails to connect

```
PROC CX_SetupCamera(num Identity, string SocketIP,  
    num SocketPort, string UserName,  
    string Password, num MinCoordX,  
    num MaxCoordX, num MinCoordY,  
    num MaxCoordY, num MinCoordRZ,  
    num MaxCoordRZ, num MinScore  
    [\Timeout:=num])  
Module: INSIGHT
```

```
nActiveCam:=1;  
CX_SetupCamera nActiveCam, strIP_Camera1, 23, "admin", "", -600, 600, -600, 600, -180, 180, 50 \Timeout:=1;
```

Robot Com Init

CX_InitComm verifies the robot can talk to the camera and opens up socket communication

- Uses the camera id made in the camera setup command
- Returns if the action was a success
- An INOUT parameter is used to determine what errors may have occurred

```
FUNC bool CX_InitComm(num CameraId, INOUT num ErrStatus)  
Module: INSIGHT
```

Robot Camera Online

```
FUNC bool CX_SetOnline(num CameraId [\On] | [\Off], INOUT num ErrStatus)  
Module: INSIGHT
```

CX_SetOnline is used to both set the camera to online mode as well as offline mode

- Returns if the action was a success
- This takes the camera identity number
- If the camera should be put in online or offline mode
- An INOUT parameter is used to determine what error occurred

Jobs can only be loaded in offline mode while it can only be ran in online mode. This is the primary use of this function

Robot Load Job

```
FUNC bool CX_GetFile(num CameraiD, INOUT string FileName,  
    INOUT num ErrStatus)  
Module: INSIGHT  
FUNC bool CX_LoadFile(num CameraiD, string FileName,  
    INOUT num ErrStatus)  
Module: INSIGHT
```

CX_LoadFile and CX_GetFile are often used side by side.

Rather than reloading an already loaded file, CX_GetFile is used to determine the active job.

- This takes in the camera ID, a variable to store the active file name in, and a variable to store any errors in

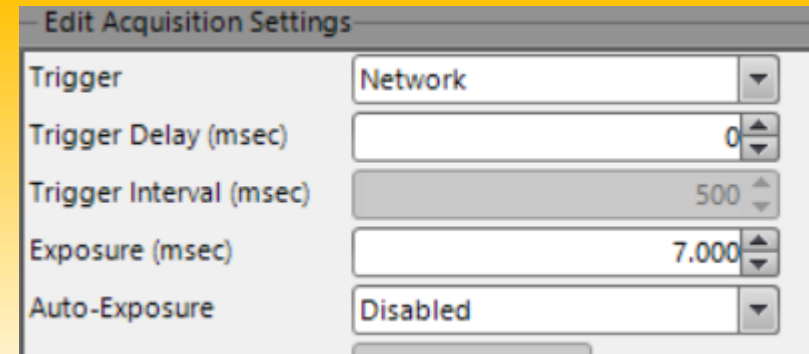
If the job that is already loaded is the same as what you were going to run you do not need to run the CX_LoadFile command

- CX_LoadFile takes the camera identity number as well as the job filename to be loaded.
- The INOUT variable is used to return any errors that might occur

Robot Trigger Image

CX_TriggerImage is used to trigger the camera to take an image

- The camera does need to be set to have a network trigger for this to work correctly
- The camera id provides what camera the robot is communicating with
- The ErrStatus is used to return any errors that may have occurred



```
FUNC bool CX_TriggerImage(num CameraId, INOUT num ErrStatus)
Module: INSIGHT
```

Robot Get Data

```
FUNC bool CX_GetVisionData(num CameralId, string CellColumn,  
    num CellRow [\Read1Cell] | [\Read4Cells],  
    INOUT pose VisionTranform,  
    INOUT num VisionScore,  
    INOUT num ErrStatus)  
Module: INSIGHT
```

CX_GetVisionData is used to pull data from the camera

- This function uses the camera id from the setup camera
- The column and the row refers to the position in the Cognex spreadsheet view
- The next 3 items return the data from the Cognex camera

CX_GetValue can also be used to pull a single point of data

	A	B	C	D	E	F
0	Image					
1	480.000	99.000	0.000	0.000		
2	Trigger	Trigger Del	Trigger Inte	Exposure (i	Auto-Expos	Max Exposu
3	Network	0	500	7.000	Disable	1000.00
4		1.000	0.000	1.000	1.000	0.000
5	Start Row	Number Of	Light Contr	Light Contr	Light Enabl	Light Enable
6	0	480	On (Exp	None	0.000	0.000
7	1.000		0.000	0.000		0
8	ism1400c	1.000	1.000	0.000	0.000	0
9	Exposure	80.000	100.000	320.000	440.000	0
10	Focus	80.000	100.000	320.000	440.000	0.000
11	White Ba	0.000	0.000	480.000	640.000	
12	White Ba	1.000	WhiteBal	32.000	0.000	

Example Code

```
PROC TestProduction()  
  VAR num nResponse;  
  TPErase;  
  MoveAbsJ [[0,0,0,0,90,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v1000,fine,tool0\WObj:=wobj0;  
  nActiveCam:=1;  
  !Setup Camera Communication  
  CX_SetupCamera nActiveCam,strIP_Camera1,23,"admin","", -600,600, -600,600, -180,180,50\Timeout:=1;  
  
  IF NOT CX_InitComm(nActiveCam,nErrStatus) THEN  
    CX_ShowErrStatus nErrStatus;  
    RETURN ;  
  ENDIF  
  IF CX_GetOnline(nActiveCam,nErrStatus) THEN  
    TPErase;  
    TPWrite "The vision is already on line";  
  ELSE  
    IF CX_SetOnline(nActiveCam,\On,nErrStatus) THEN  
      TPWrite "The vision has been set to on line mode";  
    ELSE  
      CX_ShowErrStatus nErrStatus;  
      RETURN ;  
    ENDIF  
  ENDIF  
ENDPROC
```

```
} IF CX_GetFile(nActiveCam,strFileName,nErrStatus) THEN  
  TPWrite "Name of vision program: "+strFileName;  
ELSE  
  IF strFileName=strNULL THEN  
    bStatus:=CX_SetOnline(nActiveCam,\Off,nErrStatus);  
    IF NOT CX_LoadFile(nActiveCam,strJobVision,nErrStatus) THEN  
      CX_ShowErrStatus nErrStatus;  
      RETURN ;  
    ENDIF  
    bStatus:=CX_SetOnline(nActiveCam,\On,nErrStatus);  
  ELSE  
    CX_ShowErrStatus nErrStatus;  
    RETURN ;  
  ENDIF  
ENDIF  
  
IF nResponse=1 RETURN ;  
! Reading vision data in 1 ABB formatted cell / Lecture des données dans 1 cellule au format ABB  
IF CX_TriggerImage(nActiveCam,nErrStatus) THEN  
  IF NOT CX_GetVisionData(nActiveCam,"C",26,\Read4Cells,tVisionTranfObj1,nVisionScoreObj1,nErrStatus) THEN  
    CX_ShowErrStatus nErrStatus;  
    RETURN ;  
  ENDIF  
ELSE  
  CX_ShowErrStatus nErrStatus;  
  RETURN ;  
ENDIF  
DisplayResults;  
ENDPROC
```

Work Objects

The pose data from the camera can be loaded directly to the robot Oframe in the trained work object

This will update all motions programmed for the part that utilize the trained camera work object

This data can also be fed directly into a target