



## SCS1 SERIES

*Smart Camera Sensor*



**USEasy™**

## INSTRUCTION MANUAL



We

**DATASENSOR S.p.A.**  
**Via Lavino, 265**  
**40050 Monte San Pietro**  
**Bologna - Italy**

*declare under our sole responsibility that the product(s)*

**SCS1-XX-XXXX-XXX, SMART CAMERA SENSOR AND ALL ITS MODELS**

*To which this declaration relates in conformity with the following standard(s) or other normative document(s)*

**CEI EN 55011, MAY 1999:**

*INDUSTRIAL, SCIENTIFIC AND MEDICAL (ISM) RADIO-FREQUENCY EQUIPMENT  
RADIO DISTURBANCE CHARACTERISTICS  
LIMITS AND METHODS OF MEASUREMENT*

**CEI EN 50082-2, JUNE 1997:**

*ELECTROMAGNETIC COMPATIBILITY. GENERIC IMMUNITY STANDARD.  
PART 2: INDUSTRIAL ENVIRONMENT*

**CEI IEC 60947-5-2, MARCH 1999:**

*LOW-VOLTAGE SWITCHGEAR AND CONTROLGEAR. PART 5: CONTROL CIRCUIT  
DEVICES AND SWITCHING. SECTION 2: PROXIMITY SWITCHES*

*Following the provision of the Directive(s):*

<b>89/336 CEE AND SUCCESSIVE AMENDMENTS, 92/31 CEE, 93/68 CEE;</b>
--

Monte San Pietro, 12/10/2004

Gianni Stradiotti



Quality Assurance Manager





## 1. GENERAL INFORMATION

### 1.1. General description of SCS1 Smart Camera Sensor



SCS1 is an user-friendly vision sensor equipped with advanced inspection capability that assures very precise quality controls on industrial manufacturing and automatic materials handling lines. Quality control is based on vision techniques that consist in acquiring a reference image of the target-object and configure the sensor in order to compare the template with the target-object images.

SCS1 can be configured either using the sensor integrated user interface, based on display and push-buttons, or using the SCS1 graphic interface with PC connection.

The SCS1 sensor has been developed to function in a stand-alone mode. Even with PC configuration, the digital outputs are activated only when the inspection is performed by the sensor.

SCS1 mounting and positioning are easy and immediate, thanks to a fixing bracket set that guarantees precise regulation.

Interchangeable optics are supplied as accessories, together with integrated and external illuminators that make SCS1 extremely flexible and suitable to many different applications.

The sensor integrated interface guarantees quick and easy setting of the main basic sensor functions.

The PC interface allows the user to fully exploit all the more advanced functions of the SCS1 sensor.

### 1.2. Available models

Six SCS1 models are available with the following characteristics:

#### **SCS1 with integrated illuminator**

- with RS232 serial protocol
- with RS485 serial protocol
- without serial protocol, with two supplementary digital inputs for inspection selection

#### **SCS1 without integrated illuminator**

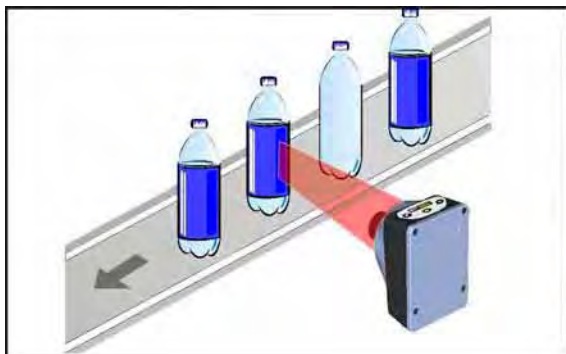
- with RS232 serial protocol
- with RS485 serial protocol
- without serial protocol, with two supplementary digital inputs for inspection selection



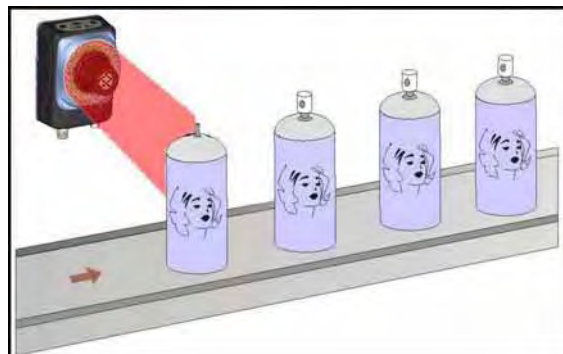
### 1.3. Typical applications

Typical SCS1 applications consist in quality and integrity controls of assembled products on an industrial manufacturing line.

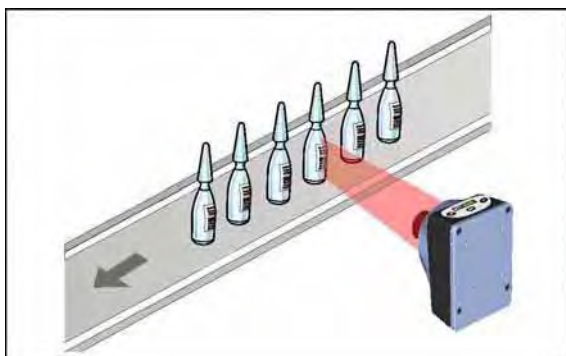
Some examples:



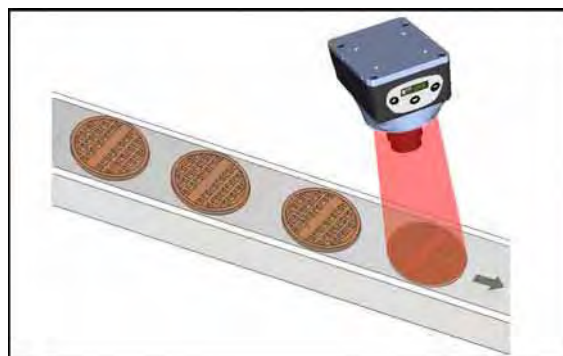
*Label presence on phials / bottles*



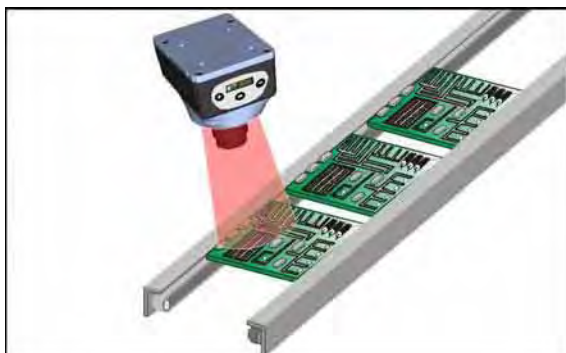
*Presence of mechanical parts*



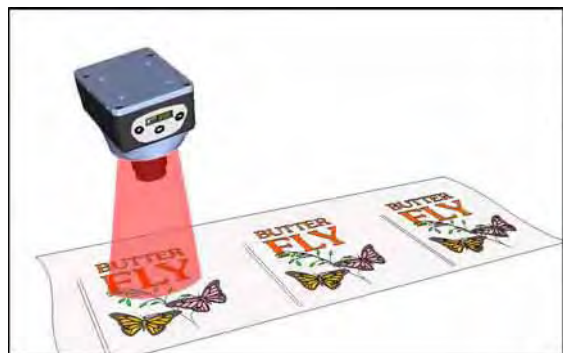
*Marking control*



*Contour control (contour match)*



*Component control on PCBs*



*Similarity check with a template*



## 1.4. Components

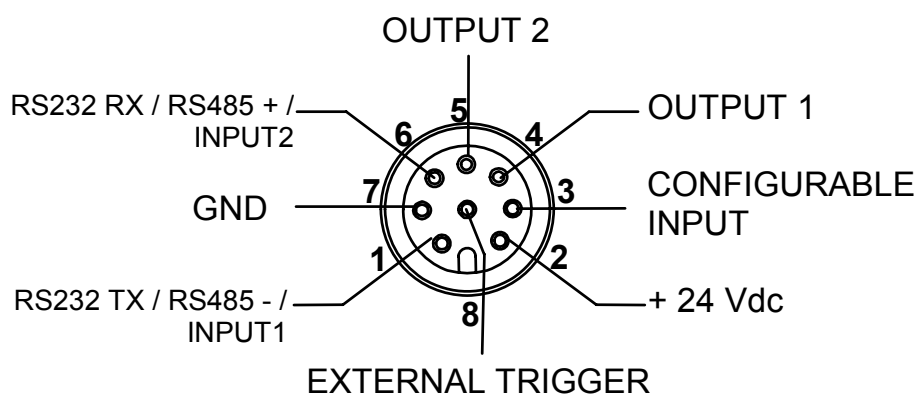
The system is composed of:

- compact sensor housing with reduced dimensions and weight;
- standard industrial connectors (M8, M12);
- 16 mm C-mount lens with 5 mm adapter ring to make it compatible with the sensor's CS-mount;
- RJ45 connector with Ethernet PC connection;
- LED function indicators and 4-digit display;
- fixing bracket set to facilitate sensor positioning;
- integrated *ring light* illuminator with deactivation option: composed of coloured LEDs able to generate both continuous as well as pulsed light (with adjustable interval) to obtain maximum efficiency;
- external illuminators powered through an M8 connector are available to generate pulsed light: ring light, spot light, back light and laser light. Illuminators dedicated to particular applications can be developed upon request.



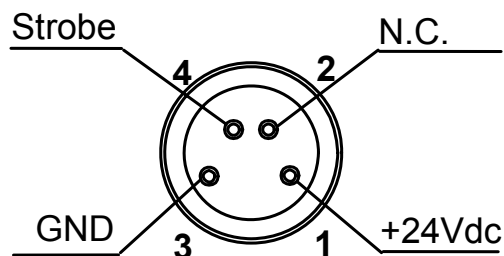
## 2. ELECTRICAL CONNECTION

### M12 connector



- |            |                                 |
|------------|---------------------------------|
| 1 = white  | = RS232 TX / RS485 - / Input1   |
| 2 = brown  | = 24 Vdc power supply           |
| 3 = green  | = Configurable input            |
| 4 = yellow | = Output 1                      |
| 5 = grey   | = Output 2                      |
| 6 = pink   | = RS232 RX / RS485 + / Input2   |
| 7 = blue   | = GND                           |
| 8 = red    | = External trigger signal input |

### M8 connector



- |           |                     |
|-----------|---------------------|
| 1 = brown | = 24 Vdc            |
| 2 = white | = Not connected     |
| 3 = blue  | = GND               |
| 4 = black | = Strobe TTL signal |

### RJ45 connector

This connector is used for sensor and PC communication using Ethernet 10/100 Mbps network.



### 3. INSTALLATION

#### 3.1. Hardware connection

Check the following necessary components:

- SCS1 sensor;
- 16mm C-mount lens and adapter compatible with the sensor's CS-mount. Alternatively, any optics with CS mounting or C mounting with adapter;
- cable with M12 8-pole connector;
- 24 Vdc power supply.

And if necessary:

- cable with M8 4-pole connector (if an external illuminator is used);
- Ethernet 10/100 Mbps cable (if the SCS1 graphic interface is used).
- Thread optics onto the sensor.



- Power the cable with the M12 8-pole connector.
- Assure that the brown wire (pin 2) is connected to the positive 24 Vdc pole and the blue wire (pin 7) to the negative pole (GND).

5) Connect the cable with M12 8-pole connector to the sensor.





- 6) *Optional:* connect external illuminator to the sensor using the M8 4-pole connector.



- *Optional:* connect the sensor to the PC using the Ethernet 10/100 Mbps cable.





### 3.2. Installation and activation of SCS1 graphic interface

#### 3.2.1. Minimum system requirements:

- Celeron® 700 Mhz processor
- 128 MB RAM
- super VGA monitor (800x600 pixel)
- Ethernet 10/100 Mbps card
- 20 MB of available space on hard disc
- CD-ROM drive

#### Recommended system:

- Pentium® 4 2 Ghz processor
- 256 MB RAM
- monitor with 1024x768 pixel resolution
- Ethernet 10/100 Mbps card
- 20 MB of available space on hard disc
- CD-ROM drive

#### 3.2.2. SCS1 graphic interface installation

To install the SCS1 graphic interface, insert the CD-ROM supplied with the sensor. The following window will appear automatically:



Figure 1

*Note:* if the CD-ROM does not automatically run, choose “Run” from the Start menu and type D:\setup.exe (substitute the appropriate drive letter of your CD-ROM drive for D) and press “Enter”.

The guided installation of the software will begin selecting “Install SCS1 Graphic User Interface”. Follow then the installation instructions provided on the window.



### 3.3. Communicating with the sensor

The PC and sensor communicate using the Ethernet network and require compatible IP addresses.

*Note:* an IP address is equivalent to a telephone number. The PC connects itself to the sensor only when the sensor's IP address is known, just like the phone number of the person to contact. Differently from a telephone number, not all IP addresses are compatible among themselves and a few rules, illustrated successively, must be respected.

All SCS1 present the following default standard communication parameters:

IP address    172.    27.    101. 209  
Subnet mask   255. 255. 0    0

#### 3.3.1. PC configuration

The following steps have to be followed to configure correctly the PC:

1. Open the "Network Neighbourhood" clicking the desktop icon (Figure 2) and select "Properties".

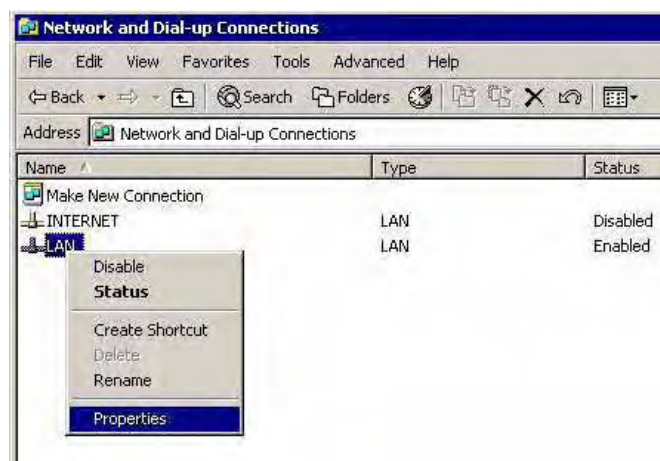


**Figure 2**

or, alternatively, use the following route:

Start → settings → control panel

Double-click "Networks and Dial-up connections" and the following window will show up:



**Figure 3**

Right-click the LAN icon and then select Properties



2. Select "TCP/IP protocol" in the visualised window and click "Properties" (Figure 4).

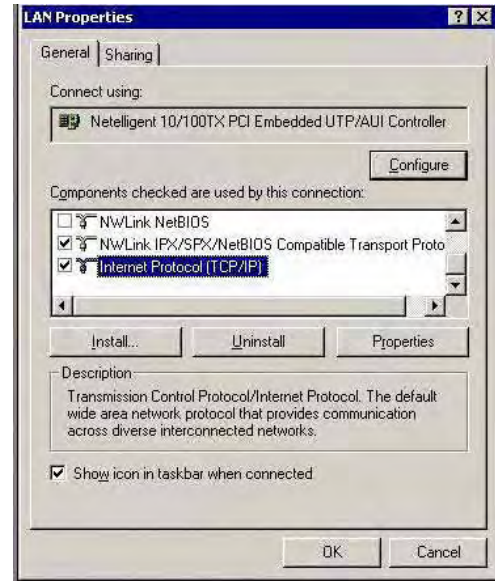


Figure 4

3. Configure the requested fields as follows (Figure 5):

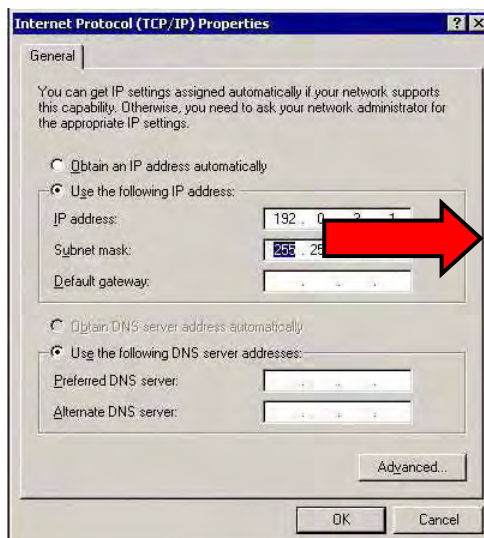


Figure 5

IP address:  
172.27.101.1

Subnet mask:  
255.255.0.0

4. Click "OK" to confirm.



### 3.3.2. Sensor configuration



The integrated push-button interface has to be used to configure the sensor's IP address.

The display options can be visualised using the + and – push-buttons, while SET is used to confirm the selected option.

1. Press SET to access the sensor configuration menu. The display visualises the **SETP** text, press SET again to access the Setup menu.
2. Scroll down using + or – until the **IPES** text is visualised.
3. Press SET to access the communication parameters.
4. Scroll down using + or – until the **IP\_A** text is visualised, then press SET.
5. The display visualises a 3-digit number which represents the first field of the sensor IP address. The 4 fields of the IP address can be visualised by using + and -. The field currently visualised can be selected by pressing SET. Pressing again + and –, the numeric value of the selected field can be changed. A continuous pressure on the push-button allows to proceed more rapidly. Press SET to confirm changes (Figure 6).
6. Once checked all 4 fields of the sensor's IP address, scroll down the menu until the **ES0** text is visualised on the display. Press SET to return to communication parameter configuration menu.
7. Scroll down menu using + or – until the **SAVE** text is visualised. Press SET to return to general menu. Alternatively, if the new configuration does not have to be saved, continue to scroll the down the menu until reaching **ES0** and press SET to exit without saving.

**Note: the changes made are effective only after sensor re-booting.**

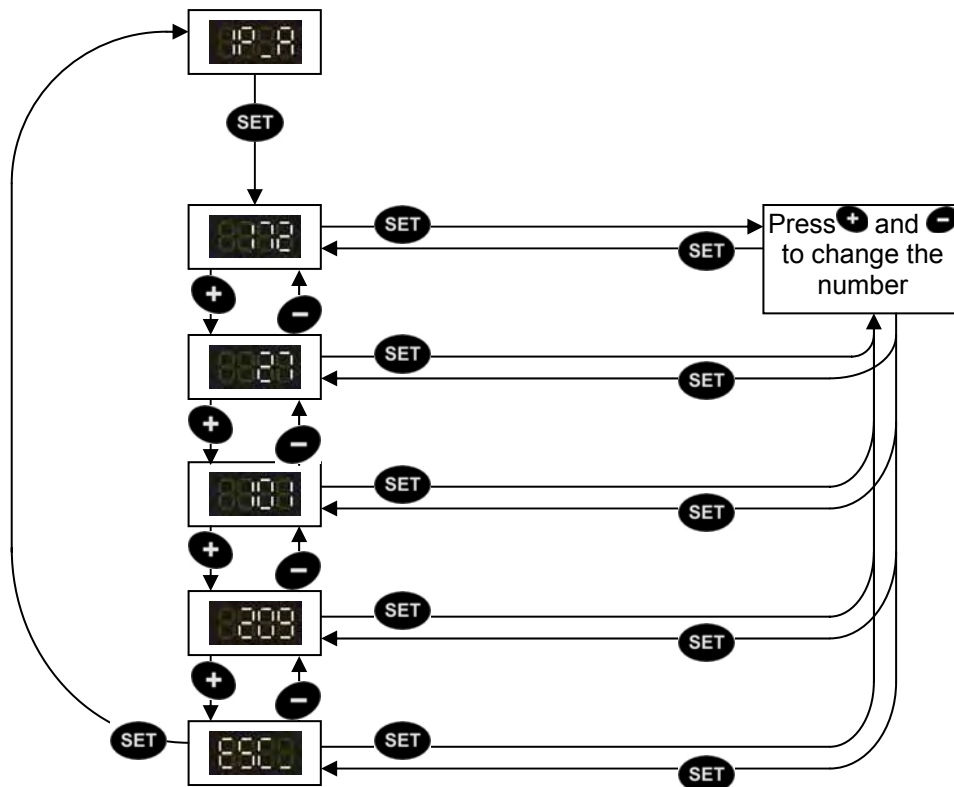


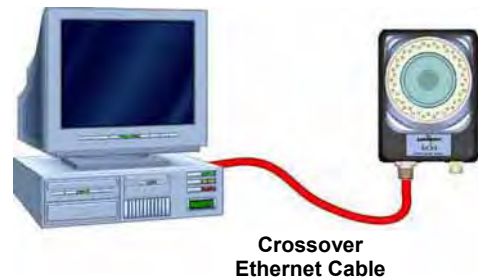
Figure 6



The PC and sensor connection can be established **directly** or using a **LAN network**.

### 3.3.3. Direct connection

An Ethernet crossover cable (with TX and RX crossed wires) has to be used in presence of a direct PC and sensor connection and the sensor and PC IP addresses must be configured

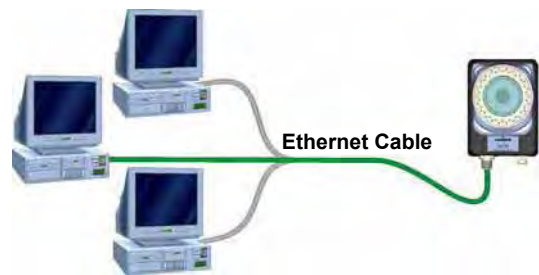


### 3.3.4. LAN network connection

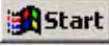
A traditional Ethernet cable has to be used in presence of a PC and sensor connection using a LAN network. Assure that the sensor IP address is compatible with the existing network.

If PC and sensor are connected directly, the user has more freedom in configuring the IP address. On the contrary, if the PC is connected to the sensor using a local LAN network, intrinsic constraints regarding the existing network configuration have to be respected.

For this reason the sensor's IP address can be changed using push-buttons and thus guaranteeing improved system flexibility.



If the PC's IP address is unknown, the user can find it as follows:

1. open a DOS window (from the  menu and select "Run", type "cmd" and click on OK);
2. type "ipconfig";
3. press "Enter".

The necessary information is then shown:

```
C:\>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale (LAN):

    Suffisso DNS specifico per connessione: datasensor.corp
    Indirizzo IP. . . . . : 172.27.30.153
    Subnet mask . . . . . : 255.255.0.0
    Gateway predefinito . . . . . : 172.27.2.254
```

Figure 7



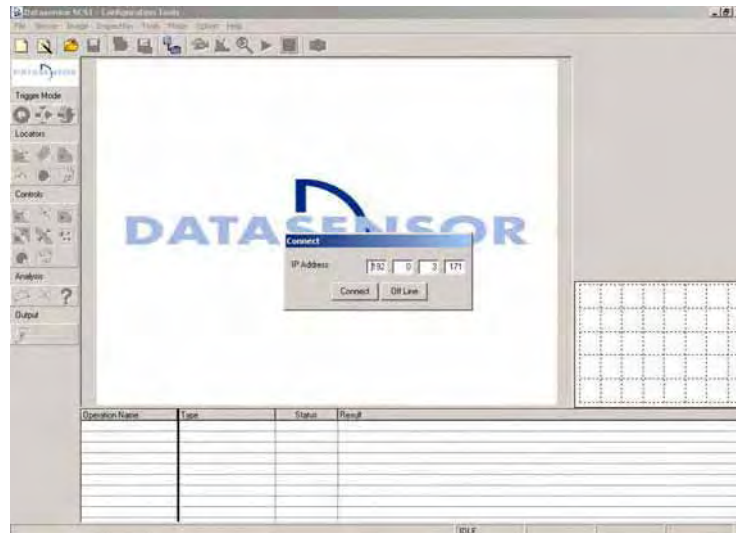
### 3.4. Software activation and sensor connection

Start the SCS1 graphic interface using a double click on the desktop icon.

Otherwise select SCS1 from the programme menu following the route:

Start > Programmes > Datasensor > SCS1.

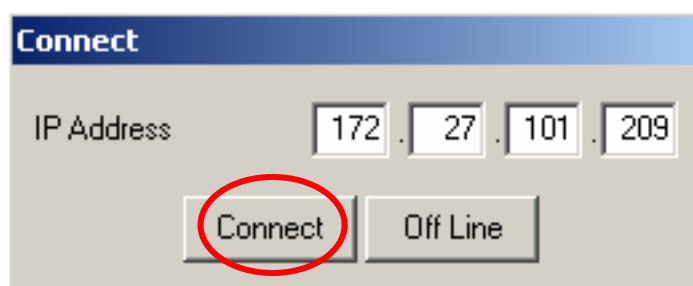
The following window appears:



**Important:** the SCS1 graphic interface can function either connected to the sensor in *on-line mode*, or not connected in *off-line mode*.

#### 3.4.1. On-line mode

To connect the sensor, set the communication parameters and then click "Connect". The default parameters are:



The "IP address" refers to the IP address associated to the sensor.

The programme will try to connect itself to the sensor. Two situations can occur:

- *correctly connected sensor*: the main programme window is visualised and the operating functions of the *on-line mode* are activated;
- *connected sensor and wrong communication parameters*: the sensor does not respond.



### 3.4.2. *Solution of connection problems*

If the connection between the sensor and PC is not established correctly, an error message is visualised on the screen. Click "OK" to continue. Before changing the communication parameters, please assure that the sensor is powered and the physical connection between the sensor and PC is correct. If properly installed, the sensor's IP address compatibility with the PC's IP address has to be then checked (refer to paragraph 3.2.2).

### 3.4.3. *Off-line mode*

The program is set to function also in the *off-line mode*, operating on images previously saved by the sensor and saved on the PC or on any other compatible image (bitmap format with 256 grey levels, 640x480 or 320x240 resolution).

The *off-line mode* can be accessed by clicking on "Off-line" instead of "Connect" after starting the programme.

The *off-line mode* offers all the options necessary for an inspection configuration (naturally with the exception of acquiring new images and checking the correct functioning, that are available only in the *on-line mode*).



## 4. SENSOR CONFIGURATION

The term “inspection” refers to all the operations the sensor carries-out to control object characteristics and control the respect of specific conditions.

The user has two possibilities to set correctly the sensor for inspection configuration:

- sensor integrated interface;
- SCS1 graphic interface.

### 4.1. General considerations

Some general conditions have to be respected when configuring an inspection:

- decide exactly what to detect;
- assure correct and constant lighting, minimising ambient light conditioning on working area;
- assure a good contrast between target-object and background;
- acquire a good reference image.

The reference image is used as a template to determine the specifications to be respected by the images acquired during the inspection.

It is thus very important to acquire a high-quality image, where the target-object presents well-defined edges and evident main characteristics:

- choose carefully the tools to use for the elaboration to guarantee the best result possible in the easiest manner (particularly in the software inspection where a wider range of tools is available);
- regulate precisely the parameters of the tools selected;
- test the inspection choices made on different variations of the same object, trying to carefully distinguish the constraints within which inspection results can be considered correct and which tolerance can be accepted without passing the constraints;
- compare results to discover any incongruence;
- re-configure tools to reduce incongruence;
- run inspection.



## 4.2. Inspection configuration using the sensor integrated interface

The sensor integrated interface consists in 3 push-buttons (SET, +, -) and a 4-digit display presenting four LEDs.



The important steps to configure an inspection using the display are three:

- 1 – inspection parameter regulation;
- 2 – background and reference image acquisition (Teach-in);
- 3 – inspection saving and starting.

### 4.2.1. Sensor configuration menu

The menu options can be visualised using the + and – push-buttons.

Press SET to select an option and the relative submenu is opened.

At powering, the device initialisation process lasts approximately 25 sec during which the “SCS1” text is visualised.

At the end, the display visualises the system status.

The main menu is accessed by pressing SET (Figure 8), offering the following options:

SETP

*Setup:* allows to access the parameters that control sensor functioning and inspection process.

REG

*Registers:* visualises and modifies the numeric values of the tool parameters set using the PC interface in the 16 sensor registers.

TEACH

*Teach-in:* self-acquisition process necessary to acquire the template used for comparison during inspection.

SAVE

*Save:* allows to save inspection and automatically enter in the *Run mode*;

NETS

*Network:* allows to access communication parameters.

DISP

*Display:* allows to change text orientation on the display.

Start

*Start inspection:* allows to return to the *Run mode*, carrying-out the previously active inspection, without memorising new settings.



## 4.2.2. Sensor configuration diagram

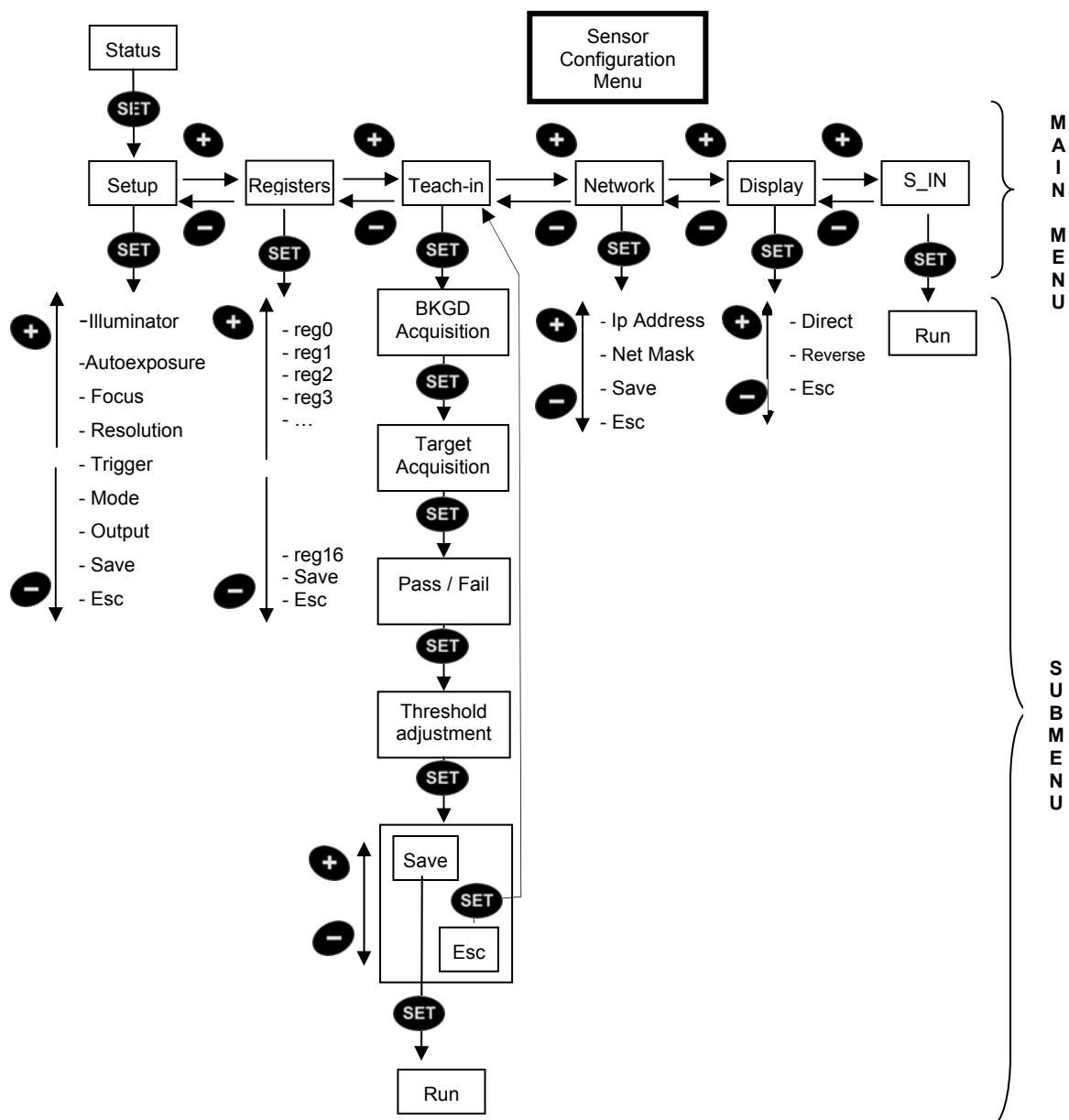










Figure 8




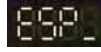
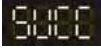








#### 4.2.3. Inspection parameter regulation





The options that control the sensor and inspection mode parameters are present in the Setup menu:

-  *Light*
-  *Autoexposure*
-  *Focus*
-  *Resolution*
-  *Trigger*
-  *Mode*
-  *Save*
-  *Exit*

The following are only the basic instructions necessary for inspection configuration. For a more detailed explanation of each option refer to paragraph 5.4.






1. Select  to activate () or deactivate () the integrated illuminator.
2. Select  to begin sensor autoexposure. Assure that, during the autoexposure process, the target-object and background to inspect are in front of the sensor. At the end of the process, the display will visualise , in case of a successful result or , in case of failure. In both cases press SET to return to previous menu.
  - Select  to focus the image. The user has to act on the sensor optics to increase the number that appears on the display. Press SET to return to previous menu.
  - Select  to choose the necessary resolution. The possible resolutions are given below:
    -  ¼ VGA (320x240 pixels)
    -  full VGA (640x480 pixels)
    -  exit from resolution setting submenu

Scroll down the options using + and –. Press SET to select the option.

- Select  to choose the *Trigger mode*. The possible modes are given below:
  -  *continuous*
  -  *external*
  -  exit from trigger mode selection submenu





Scroll down the options using + and -. Press SET to select option.

- Select  to choose inspection method. The possibilities are given below:
  -  Blob count
  -  Pattern count
  -  Contour match
  -  Exit

Scroll down the options using + and -. Press SET to select option.

Selecting Blob count and Pattern count, a number appears on the display that can be changed using + and -. The number corresponds to the number of objects expected to be found during inspection. Press SET to return to previous menu.

- Select  to save present configuration or  to exit without saving.






#### 4.2.4. Registers

The Register menu consents the configuration, through the PC interface, of the value associated to the 24 registers representing the parameter tool values set for the current inspection. These values can be modified in the Register menu (refer to specific section at page 54).



The numeric value can be modified choosing the specific register:

- 
- 
- ...
- 

The <register>/<parameter type > association (e.g *reg1/blob count threshold*) is not expressed in the register menu, but has to be remembered by the user that has set the current inspection.

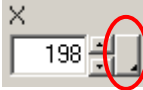
To change the value of a register press . The current register value is visualised. To change the value press  or . To save press  or to leave unchanged the value press  once and re-press it without modifying anything.

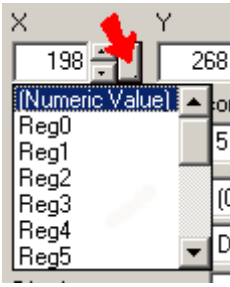
The sequence has to be repeated to modify each single value/register.

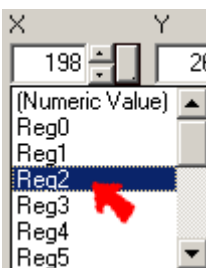
To save the new configuration press  or press  to cancel modifications made.

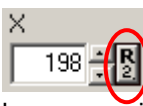


To associate a parameter to a register, the following 4 simple steps have to be followed using the graphic interface provided:

1.  Click on the grey box on the left of each parameter.

2.  A window appears.

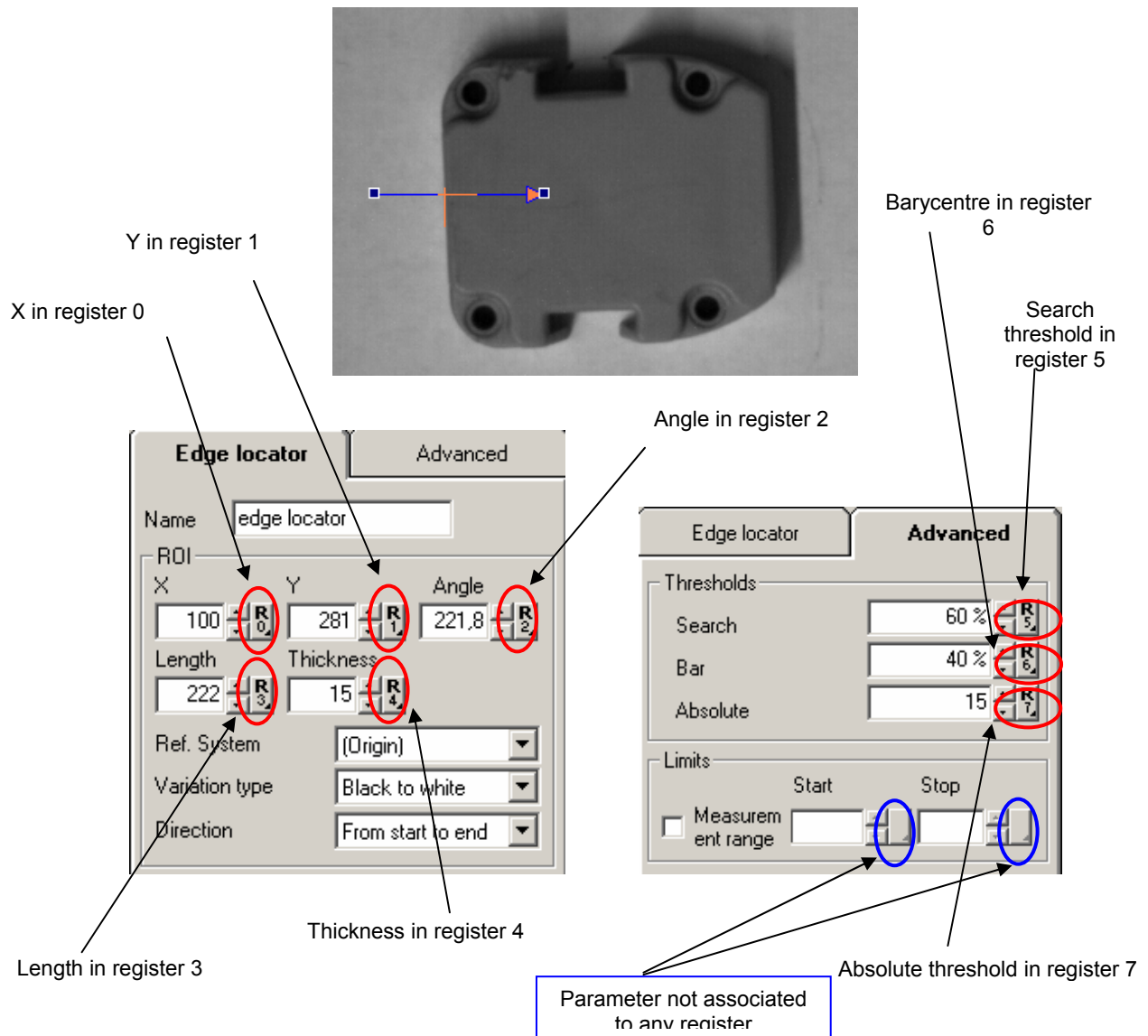
3.  Select the register to be associated to the parameter (e.g. register 2).

4.  The **R2** text will appear inside the grey box signalling that the parameter has been associated to *register 2*.



**Example:**

An example is provided of how to use parameters in order to better understand their functioning. An inspection will be created using the *Edge locator* tool and some parameters will be associated to different register tools. Then, the control and change of the register values will be shown using the SCS1 integrated interface (numeric button).



The above images show the configuration of an inspection containing an *Edge locator* and how the parameters are associated to the different SCS registers.

In particular, the following associations have been made:

- The *ordinate* value of the intersection point of the position vector with the edge to *register 0*.
- The value of the intersection point of the position vector with the edge to *register 1*.
- The *angle* value formed by the direction vector of the shape tool with the horizontal axis to *register 2*
- The *length* value of the tool vector to *register 3*
- The *thickness* value to *register 4*
- The *search threshold* value to *register 5*
- The *barycentre* value to *register 6*
- The *absolute threshold* value to *register 7*



*Note:* the user configuring the inspection has to remember the association between the particular register and the relative parameter as this information is not given in the display of the integrated interface.

Once save the inspection on the device, the register values can be then controlled without using the graphic interface setting the sensor in the *Run mode*. The values can be modified using the integrated device interface in the following manner:

1. In *Run mode* press **SET**.
2. Using **+** and **-** choose the **rEG** menu.
3. Select the menu using **SET**.
4. The name of the first **r000** register will appear.
5. Using **+** and **-** choose the register amongst the 16 available **r000** ÷ **r115**.
6. Once chosen the register (e.g. **r1000**) press **SET** to visualise value (in this case **281** - the Y parameter value of the *Edge locator* tool).
7. To modify the value press **SET**. Using **+** and **-** set the new value. Re-press **SET** to confirm (to leave the value unchanged press **SET** once and re-press it without modifying anything).
8. Repeat 5, 6 and 7 steps when changing the registers.
9. To save each change choose **SAVE** or to cancel changes choose **ESC** and so return to main menu.

#### 4.2.5. Teach-in setting

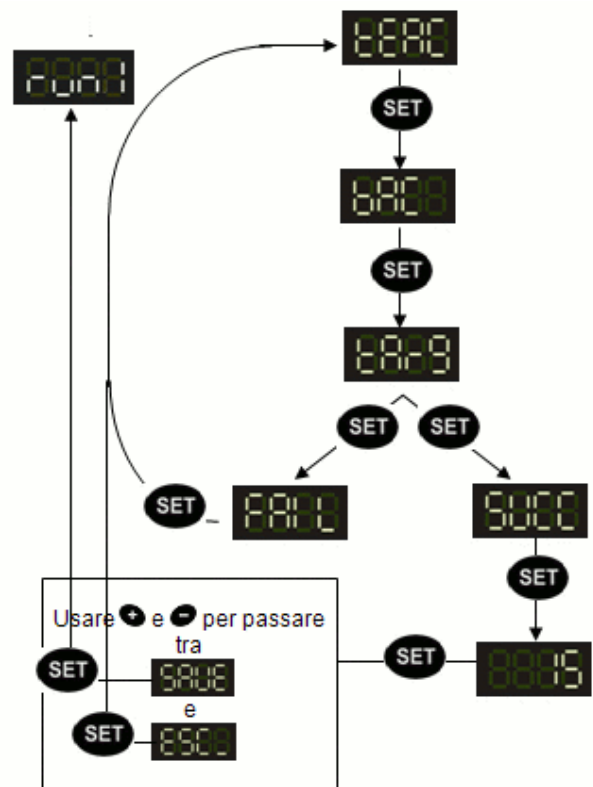
The device self-acquisition or Teach-in is activated by selecting "TEAC". The following operations are sequential. The user proceeds step by step pressing SET.

**Step 1 "BAC".** Background acquisition. Press SET to acquire background and pass to next step.

**Step 2 "TARG".** Reference image acquisition. Position target-object in front of the sensor and press SET to acquire and then pass to next step.

**Step 3 "SUCC"/"FAIL".** Depending on the success or failure of the acquisition the "SUCC" or "FAIL" text is visualised. In case of failure press SET and return to main menu. In case of a positive result press SET to pass to next step.

**Step 4 "Threshold** Regulation of the acceptance threshold. The value visualised represents the acceptance threshold that will be visualised during inspection. The lower the threshold, the more template and target-object have to correspond to reach a successful inspection. Use + or - to





change the threshold. Press SET to reconfirm the Teach-in sequence. The lower the threshold, the more template and target-object have to correspond to reach a successful inspection. User + or – to change the threshold.

Press SET to reconfirm the Teach-in sequence.

**Step 5 “Confirm”.** Press SET to confirm Teach-in sequence. The inspection can now be saved or exit from the menu pressing respectively SAVE or ESC (using + or -).

The sensor is now configured and inspection can begin. If it's necessary, press S\_In to make the inspection run.



### 4.3. Inspection configuration using the SCS1 graphic interface

The SCS1 graphic interface can be used to configure an inspection. Then the user has to choose "Connect" in the window to enter in the system's *on-line mode* (Paragraph 3.4).

The main window of the SCS1 graphic interface is visualised (Figure 9) and the system enters in an "IDLE" status awaiting instructions.



Figure 9

The inspection configuration can now begin in one of the following ways:

1. *Free new inspection creation* (for advanced users): all the available options are active and the user has complete freedom in developing inspections.
  - *Wizard-guided new inspection creation* (recommended to new users): the SCS1 graphic interface suggests some steps to follow and guides the user in the creation of a new inspection.



#### 4.3.1. Free new inspection creation

The SCS1 graphic interface functioning is characterized by different operating steps (*Live mode*, *Setup mode*, *Test mode*, *Run mode*, *Monitor mode*). Each operation carries-out specific functions. Please refer to chapter 6 for complete details of each function. The creation of a new inspection follows a logic order that can be summarised as follows:







- Open a new inspection
  - Target positioning
  - Resolution selection
  - Lighting check
  - Exposure time setting
  - Image focussing
  - Discarded image selection
  - Output configuration
  - Saving of reference images
- }  *Live mode*
- 10. Trigger selection
  - 11. Locator insertion
  - 12. Control insertion
  - 13. Analysis tool insertion
  - 14. Test tool insertion
  - 15. Inspection saving
- }  *Setup mode*
- 16. Functioning check of tools used
-  *Run mode*
- 17. Inspection time check
-  *Monitor mode*
- 18. Visualises the last 10 images elaborated by the the current scrip.
-  **Test Mode**
- 19. Visualises the last image saved showing the tool status
-  *Test Mode da telecamera*

Table 1 summarises the actions to carry-out during each operating status:

Live mode	Setup mode	Run mode
1) Target positioning	1) Trigger selection	Tool check
2) Resolution selection	2) Locator insertion	<b>Monitor mode</b>
3) Lighting check	3) Control insertion	Inspection time check
4) Exposure time	4) Analysis tool insertion	<b>Test Mode</b>
5) Image focussing	5) Test insertion	Visualises the last image saved showing the tool status
6) Test image saving	6) Serial Output Insertion	<b>Test mode from telecamera</b>
7) Output configuration	7) Inspection saving	Visualises the last image saved showing the tool status
8) Reference image saving		

Table 1

Note: the SCS1 graphic interface is only a means that enables the user to fully exploit all the more advanced sensor functions. Once configured and saved the inspection on the sensor, the



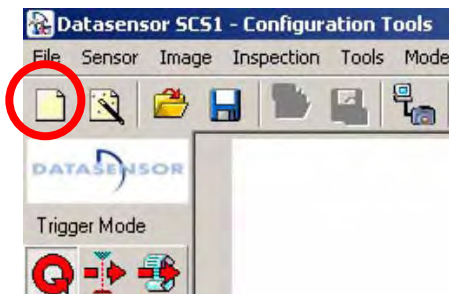
device can function in the stand-alone mode and can be disconnected from the PC. Moreover, the digital outputs can be activated only if the sensor is disconnected from the PC or runs in *Monitor mode*.



## 1. Opening a new inspection

The creation of a new inspection can be selected either from the menu or from the toolbar.

**Toolbar**



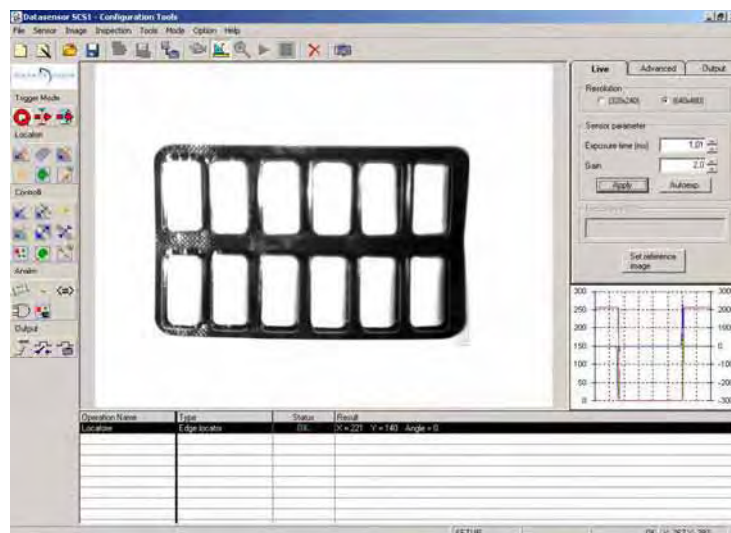
**Menu**



Once the “New Inspection” is selected, the system enters automatically in the *Live mode*. The images acquired by the sensor are visualised in this mode.

An application example is herein provided.

The application consists in the counting of tablets in a blister and integrity control. Figure 10 shows the image visualised during the *Live mode* phase.



**Figure 10**

The user can control what the sensor acquires during *Live mode* and so all aspects concerning the quality of the acquired image have to be optimised.

This is a key operation to guarantee the success of the application.

## 2. Positioning

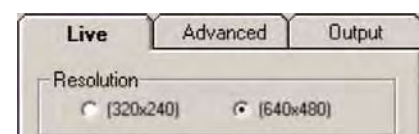
Position the target-object inside the field of view.

## 3. Resolution

Select the resolution of the image to visualise.

Different resolutions are available: VGA (640x480 pixel) and ¼ VGA (320x240 pixel).

A larger working field is obtained by selecting VGA while ¼ VGA favours inspection speed, reducing the image acquisition time.





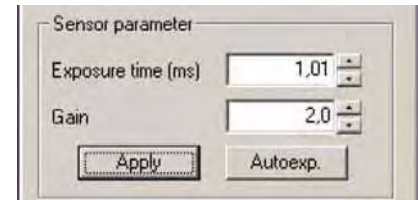
#### 4. Lighting

Ensure that lighting is uniformly distributed on the entire field of view (FOV), paying attention to reflective surfaces. The integrated illuminator can be activated or deactivated, selecting or not the option. External illuminators are available as accessories.

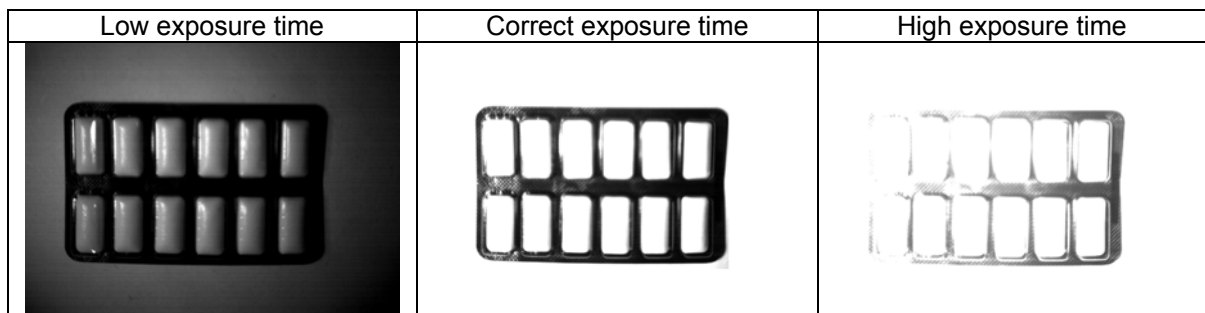


#### 5. Exposure time

Start the sensor autoexposure process clicking on the "Autoexpose" push-button. The user can control the period in which the image-acquiring device (CMOS) is exposed to light. The longer the exposure times, the more light enters into the device and consequently the image is brighter. If the result obtained is not satisfying, the regulation can be made manually, changing the "Exposure" parameter.



The results obtained by increasing the exposure time are obtained also by increasing the sensor's *gain* value. The user must be careful as gain increase generates electric noise increase in the amplifying circuit and the image quality is reduced.



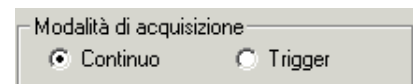
#### 6. Image focusing

Correct focusing is indicated by:

- numeric value

To maximise focus, turn the sensor lens' ring trying to increase the focus value (or level indicator).

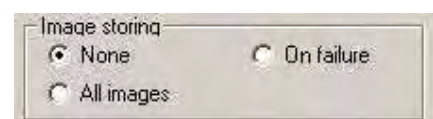
#### 7. Acquisition mode



Configures the reference image using the trigger hardware, if set in the *trigger mode*. For example, if a dynamic scene is controlled, a particular phase is difficult to obtain by pressing the "saving the reference image" icon. The acquisition of the reference image in the *trigger mode* and, using a trigger hardware, has to be set and then save the singular phases of the inspected scene until reaching the necessary configuration. In this case, pressing "saving the reference image" and providing another trigger pulse, a particular image can be saved as our template.

#### 8. Image saving

The elaborated images can be stored in the sensor memory during inspection and visualised successively in *Test mode* to check tool elaboration functioning. Different saving modes can be selected: saving all elaborated images, only those which have generated an inspection failure or none. Up to 10 images can be saved. Surpassed this limit, the new images are saved over the existing ones, following a FIFO logic: the oldest image is replaced by the latest image.



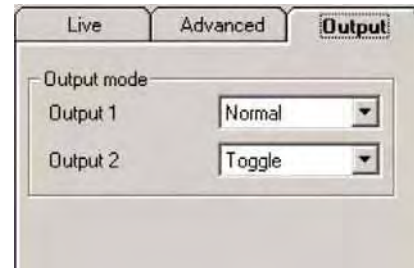


## 9. Output configuration

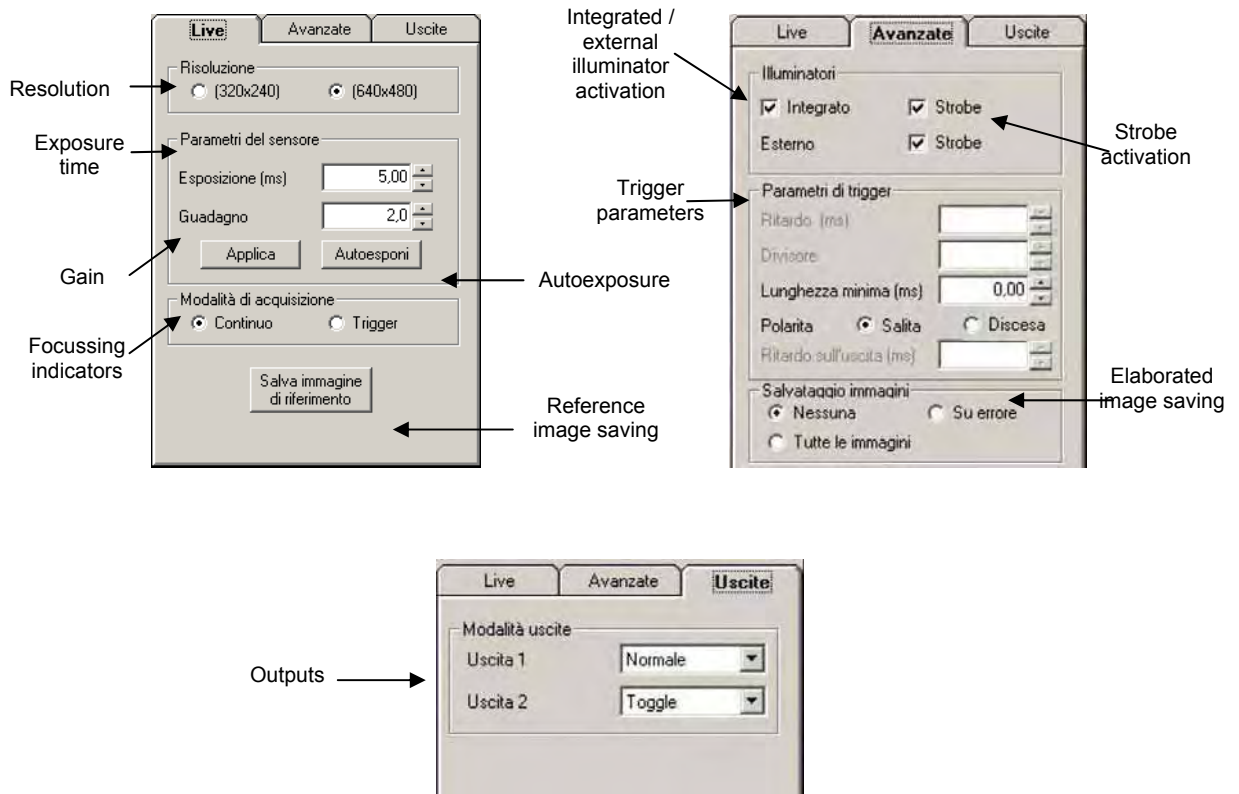
Two digital outputs are available to gather inspection information.

The outputs can be configured as follows:

- *Normal functioning*: the output switches according to the inspection results (**OK**/**Failure**), if the tools have been configured correctly.
- *Toggle mode*: the output switches each time the inspection cycle is completed.



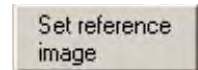
### Live mode panel parameters





## 10. Reference image saving and Setup Mode

When the image quality is acceptable, click on “Set reference image” and the image visualised is saved. It is then available as a reference image in the inspection Setup phase. The programme enters automatically in the *Setup mode*.



The real inspection is developed in the *Setup mode*. The SCS1 graphic interface offers a series of tools necessary to recognise the characteristics that the object has to respect to be considered valid. Each tool carries-out an inspection operating step and the entry order establishes the execution order during the elaboration. The tools are divided in three groups: *Locators*, *Control* and *Analysis*. Theoretically this should also represent the insertion order, even if it is only a logical sequence followed in most cases.

The only really fundamental tool is the *Test tool*. The Test result in fact guides the device outputs.



## 11.Trigger selection

Trigger available:    *Continuous trigger*  
                               *Hardware trigger*  
                               *Software trigger*

The following example uses the *Software trigger* to start image elaboration.

The *Software trigger* is represented by a dashed line and the insertion is made graphically on the reference image. The functioning can be compared to a photocell functioning. The elaboration runs when a light variation appears along the line due to the passage from white zones to black zones and vice versa. Typically, this condition occurs when the target-object passes through (object must have a sufficient contrast with the background).

- Select the tool clicking on the toolbox icon (Figure 11) or from the menu (Operators → Snap mode → Software).
- Position the trigger graphically on the image by moving the cursor with the mouse. Set the ends of the dashed line by pressing (first end) and then releasing the left button of the mouse (second end).
- Select the variation type that has to be tracked (from white to black in the example provided).

To begin inspection, the *Software trigger* has to find an edge on the dashed line.

**What is an Edge:** a variation of the brightness between adjacent pixels, that presents higher values respect to a given threshold. In the following example, the variation occurs when the blister (black object on white background) intercepts the *Software trigger*'s dashed line.

**What is a Pixel:** the smallest image part (picture's element).

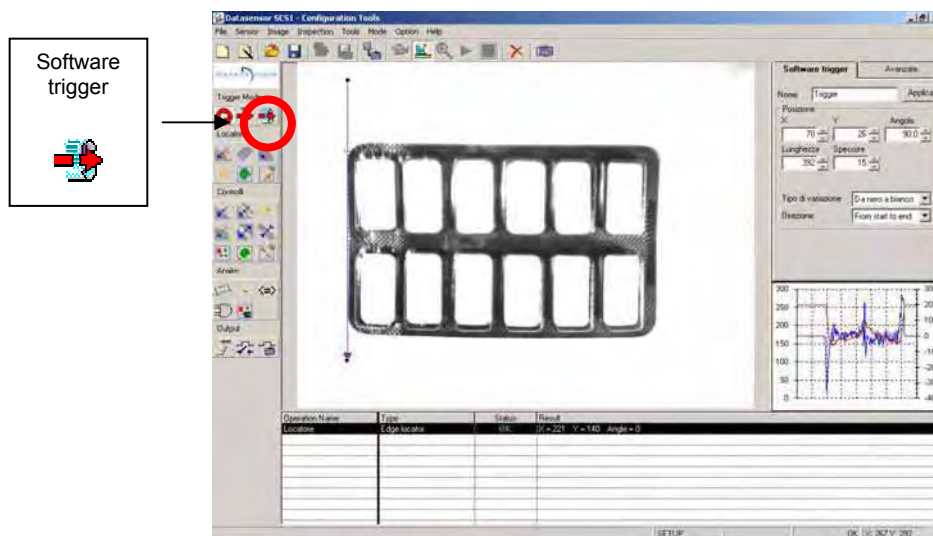


Figure 11



## 12.Locator insertion

The position of the target-object has to be properly identified in order to carry-out an inspection, considering also that, with respect to the template, the object can move inside the field of view (FOV) during device functioning.

Different locators are available, in the following example an *Edge locator* is used.

The check of the blister's horizontal position is ensured by the *Software trigger* (that switches only when the object reaches the desired position). In this manner, only the variations in the vertical position have to be avoided using an *Edge locator* (we will simply consider that the object will not rotate).

- Select the locator clicking on the toolbox icon (Figure 12) or from the menu (Operators → Locators → *Edge Locator*).
- Position the locator graphically on the image, by moving the cursor with the mouse. Set the ends of the continuous line representing the locator by pressing (first end) and then releasing the left button of the mouse (second end).
- Select the variation type that has to be tracked (from white to black in the example provided).

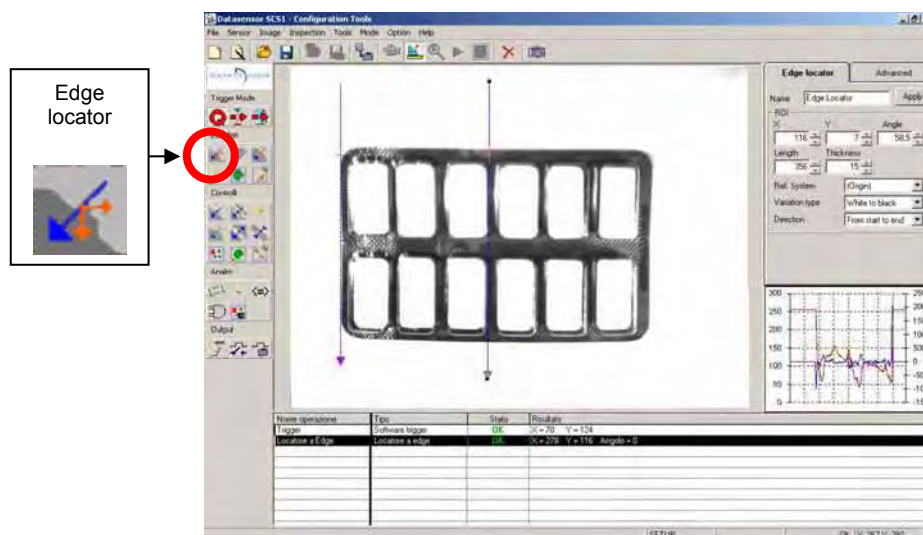


Figure 12

The *Edge locator* (similarly to the *Software trigger*) seeks an edge along the line, distinguishing the exact vertical position of the top side of the blister.

## 13.Control insertion

In the following example the *Blob count tool* is used for tablet counting. After selecting the tool, the user has to draw a rectangle around the blister.

**What is a Blob?** The image has 256 possible grey shades (0=black, 255=white). The *Blob count* sets automatically a threshold (denominated "Binarization") and distinguishes white pixels (brightness higher than the threshold) from black pixels (brightness lower than the threshold). The Blob is a group of adjacent pixels of the same colour, white or black.

- Select the tool clicking on the toolbox icon (Figure 13) or from the menu (Operators → Controls → Blob Count).
- Position the tool graphically on the image, by moving the cursor with the mouse. Set the corners of the rectangle representing the *Blob count* by pressing and then releasing the left button of the mouse.
- Select the blob type that has to be tracked (white in the example provided).
- Select the *Edge locator* used previously as the reference system (Figure 14). In this manner the *Blob count* position is relative to the edge position identified by the *Edge locator*.



moves up or down, the edge position changes and the *Blob count* rectangle is consequently modified.

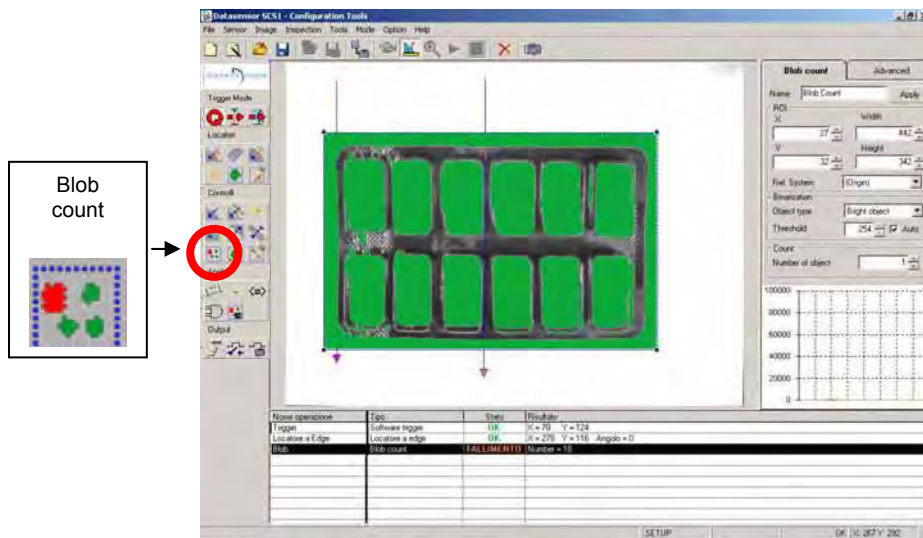


Figure 13

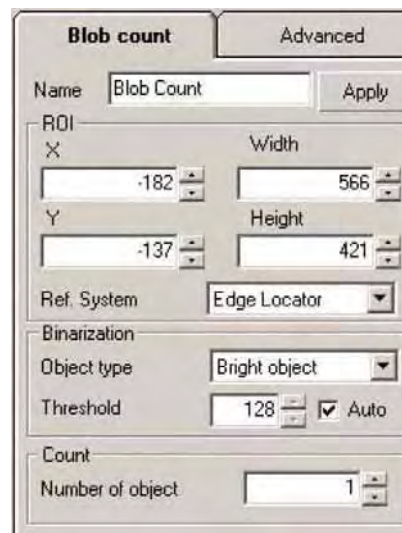


Figure 14

The *Blob count* tracks white blobs and in our example finds 18: 12 are tablets, the remaining are blobs generated by background (also white-coloured) and noise. The blobs found are coloured green.

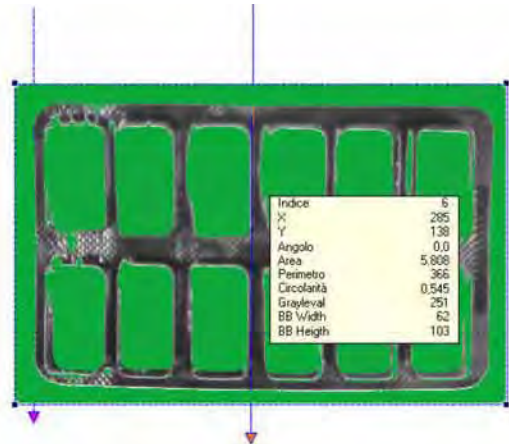
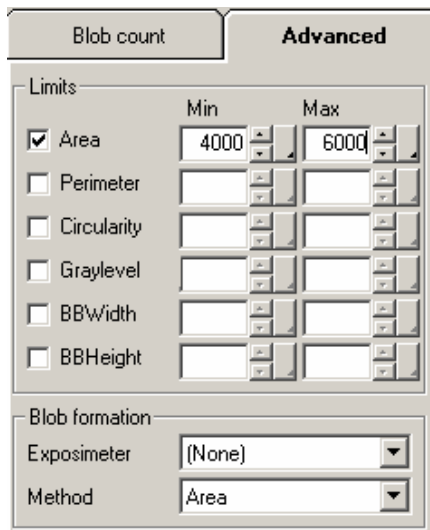
To overcome this situation, the user can limit the blob size. By doing this, unwanted blobs are discarded and the tablets, usually of the same size, are counted.



Limit setting can be made in two different modes:

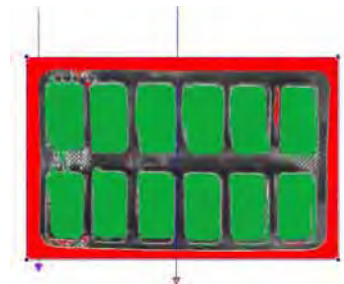
### Manual configuration

- Right-click on one of the blobs to visualise the relative technical data, including the area. In the example provided, the selected blob refers to a tablet with a 5808 pixel area.



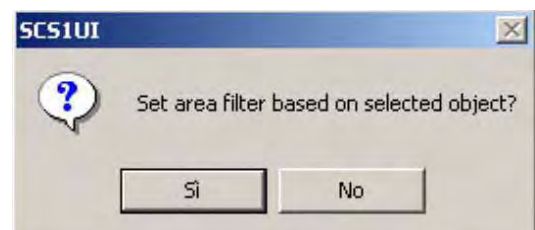
-Select the advanced options and click on “Area”, then set the minimum and maximum constraints of the blob area. In the example 4000 and 6000 pixels are set.

- The constraints are activated by clicking on “Apply”. The tablet blobs remain green and the background and noise blobs are marked in red (they are still recognised as blobs but they are discarded as they do not respect specifications). The counting result is thus 12.



### Automatic configuration

- A double click on a blob area allows the user to set automatically the constraints according to the selected blob area.
- The constraints are activated by clicking on “Yes” ( $\pm 10\%$  of selected area). The blobs generated by the background and noise are marked in red (they are still recognised as blobs but they are discarded as they do not respect specifications). The counting result is thus 12.



During this process, the operating status is a **Failure**, as the operating status (**OK** / **failure**; omitting **Error** for the moment) is determined comparing the counting result (12) and the “Count” parameter value (1 by default). “Count” is the parameter that indicates how many blobs of a specific type are expected to be found. The user just has to set “Count” = 12 and click on “Apply” and the operating status becomes **OK**.

### 14. Analysis tool insertion

*Analysis tools* are able to elaborate the results of all the other tools used in the previous operating phases.

In our example no *Analysis tools* are necessary.



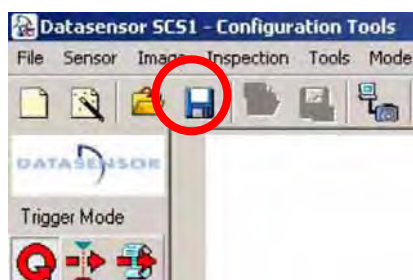
## 15. Inspection saving

Once inserted all the necessary tools, the inspection can be saved either on the PC or on the sensor.

### PC saving

To save the inspection on PC, the user has to simply click on the toolbar icon or select "Save" from the File menu and assign a name when requested.

Toolbar



Menu

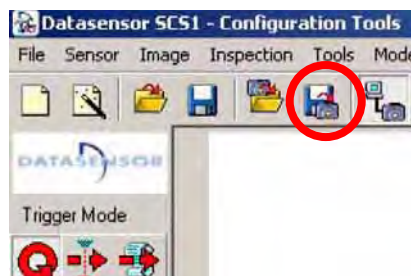




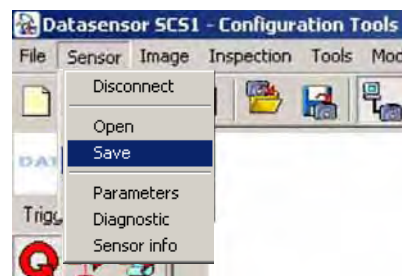
## Sensor saving

The sensor has four memory slots that can store four different inspections. To save the inspection on the sensor, the user has to simply click on the toolbar icon or select "Save" from the Sensor menu. The program will request PC saving, if not previously done. This procedure is recommended but not compulsory.

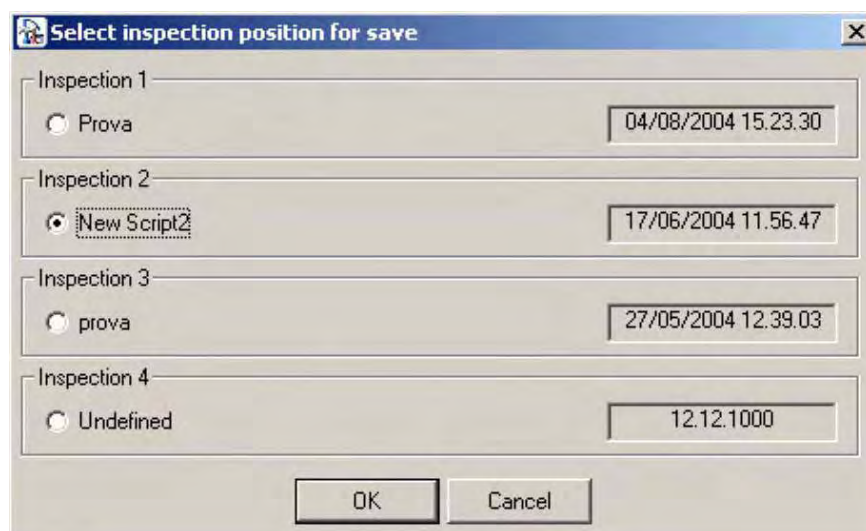
**Toolbar**



**Menu**



The following window is visualised:



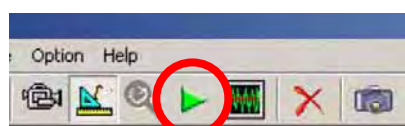
Select the memory slot where the inspection saving is desired and click on "OK".

## 16.Functioning check of tools used

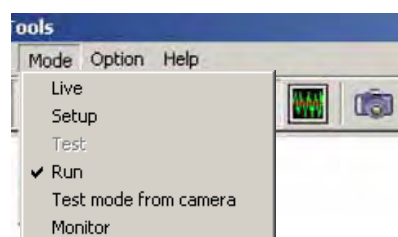
Terminated the Setup phase, the user can pass in *Run mode* and control if the inspection is carried-out correctly.

To access the *Run mode*, the user has to select the toolbar icon or select "Run" from the Mode menu:

**Toolbar**



**Menu**





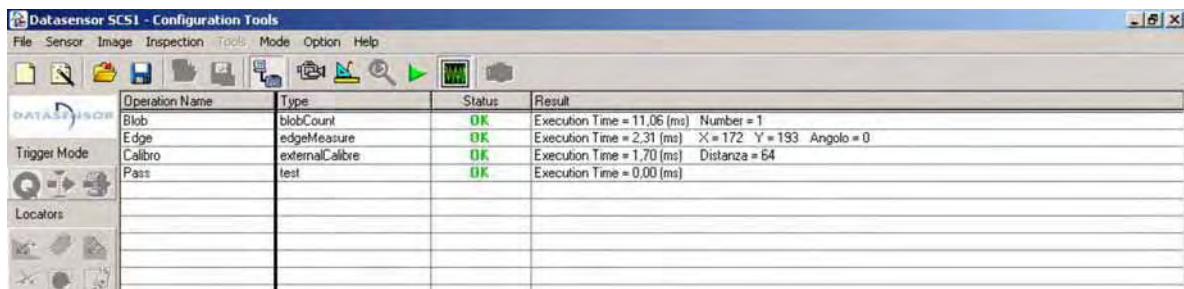
During the *Run mode*, the sensor acquires images that are elaborated by the PC. The tools status changes according to the image acquired. In this manner the correct functioning of each tool can be verified.

If inspection functioning is not acceptable, the user can return to the *Setup mode* (clicking on toolbar icon) and correct the parameters of the operating phases, insert new tools and remove others.

When results are acceptable, the inspection can be saved again either on PC or on the sensor.

## 17. Inspection time check

The *Monitor mode* allows to check the inspection time. The sensor elaborates the acquired images and transmits the results obtained to the PC. The images acquired are not visualised on the screen but the results are given in a text format. During sensor functioning, tool information about status, results and execution time is visualised.



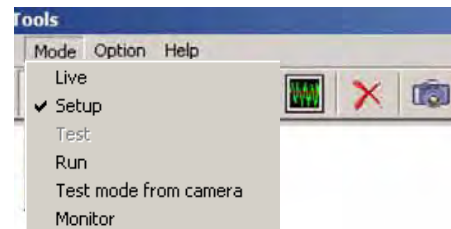
Operation Name	Type	Status	Result
Blob	blobCount	OK	Execution Time = 11,06 (ms) Number = 1
Edge	edgeMeasure	OK	Execution Time = 2,31 (ms) X = 172 Y = 193 Angolo = 0
Calibro	externalCalibre	OK	Execution Time = 1,70 (ms) Distanza = 64
Pass	test	OK	Execution Time = 0,00 (ms)

The user has to save an inspection on the sensor before passing to the *Monitor mode* and so activating the elaboration. The *Monitor mode* can be selected from the toolbar or from the menu:

**Toolbar**



**Menu**



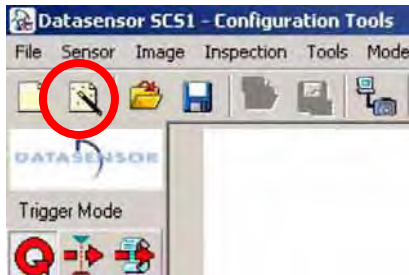


#### 4.3.2. Guided new inspection creation: Wizard

The creation of an inspection guided by the Wizard allows even a new user to configure an inspection. The Wizard supplies a series of guided steps. Each step requests the user to select a specific inspection feature. Each choice conditions the successive choices.

The Wizard guided inspection creation can be selected from the toolbar or from the menu:

**Toolbar**



**Menu**



Seven steps have to be carried-out:

- Step 1      *Live: regulation of parameters relative to image acquisition and saving of reference image*
- Step 2      *Inspection type selection*
- Step 3      *Trigger selection*
- Step 4      *Locator selection*
- Step 5      *Locator positioning*
- Step 6      *Control positioning*
- Step 7      *Output selection*



## ▪ Step 1 – Live

The images acquired by the sensor can be visualised on the image panel.

Some sensor parameters have to be set before creating the inspection in order to guarantee the best reference image possible. The parameters are located on the panel on the right-side of the window. The user can obtain a description of the operations to carry-out, by positioning the mouse cursor on the options of the left-side of the Wizard panel.

- select the resolution;
  - run the autoexposure process;
  - focus the image;
  - if the image obtained is not acceptable, refine the exposure time manually and repeat focusing;
  - when a acceptable image is obtained, click on “Set reference image” to continue.
- **Resolution:** this parameter identifies the resolution used in the inspection elaboration phase. Two different inspection resolutions can be selected: 640x480 or 320x240, that respectively favour the dimensions of the area acquired by the sensor and the elaboration speed.
  - **Autoexposure:** click on “Autoexpose” to run the automatic regulation process of the sensor exposure time.
  - **Focussing:** to maximise focus turn the sensor lens' ring. The focus level is visualised on the right side of the image in a numeric value or bar.
  - **Exposure time regulation:** change manually the exposure time to improve image quality. The longer the exposure time, the brighter the image.
- Note:* the ambient light must be the most constant possible.

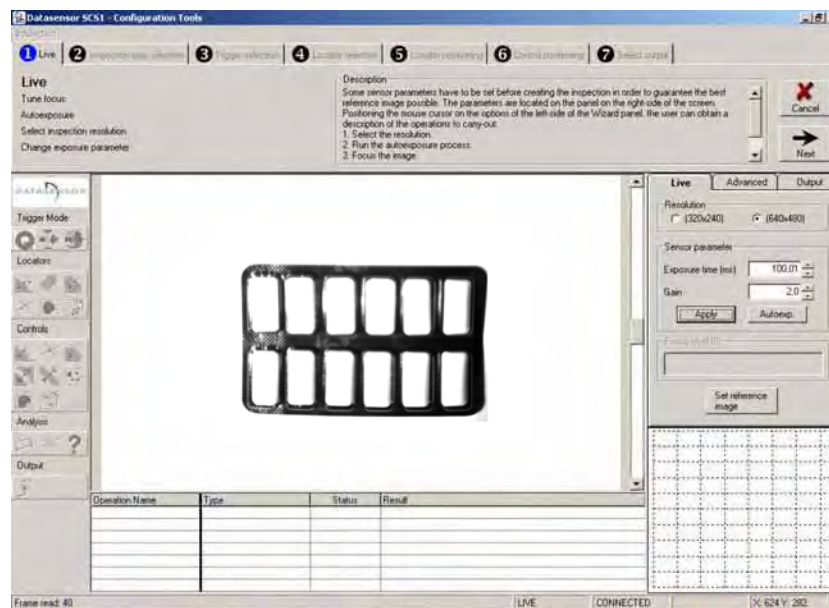


Figure 16



## Step 2 – Inspection type selection

The user must select one of the predefined Wizard inspections. A brief description of each description facilitates the user in selecting the inspection that best suits the application. To visualise the description the user has to position the cursor over the desired inspection.

To select an inspection the user has to simply click on the icon of the specific inspection, next to the inspection title.

No other operations are required. Just click on “Next” to continue.

The selection made affects the successive steps. Not all inspections require all steps and so the Wizard might skip automatically some steps.

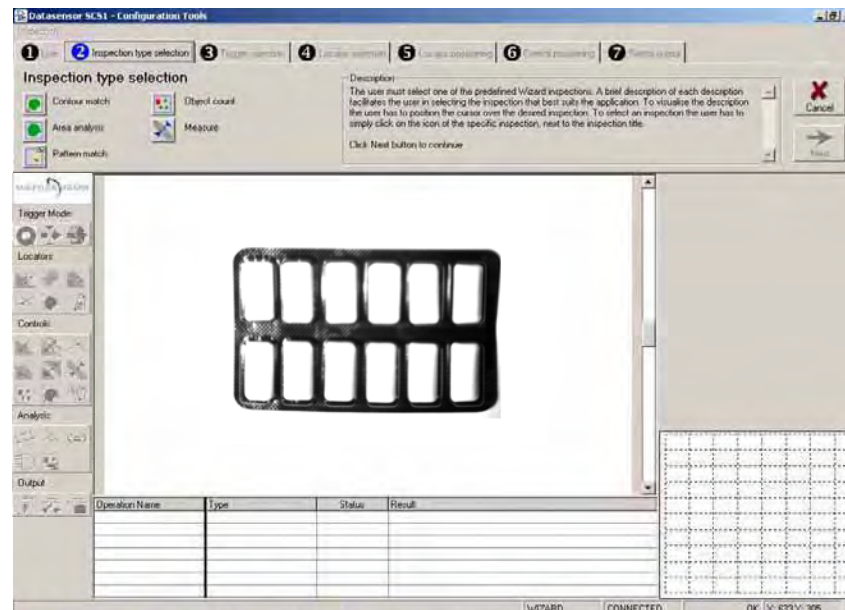


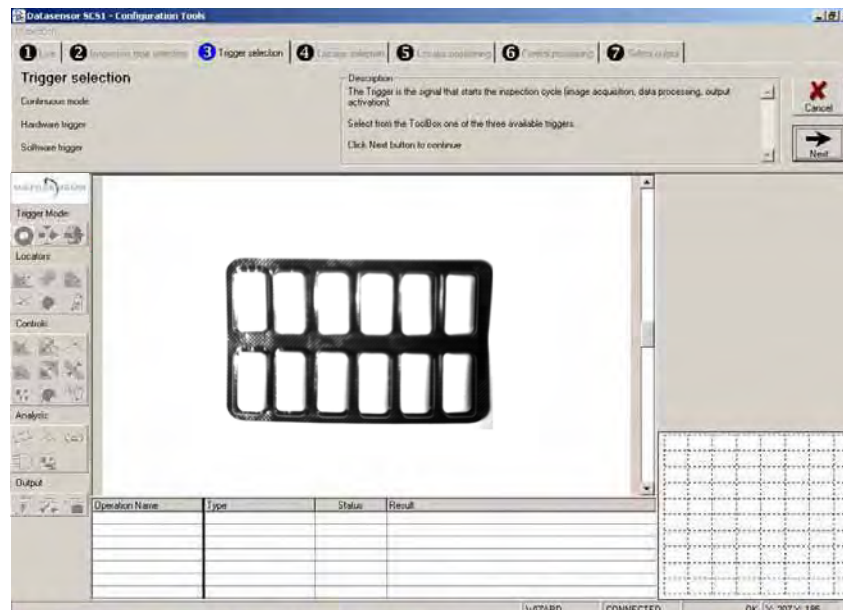
Figure 17

- **Contour match**  
Recognises an object analysing the contour. Saved the target-object contour, an object with a similar contour is tracked inside the working area.
- **Area analysis**  
Recognises an object checking area, perimeter and other geometrical characteristics.
- **Pattern match**  
Recognises an object comparing it with the target-object. The comparison involves each single pixel, evaluating brightness differences between the template and the target-object.
- **Object count**  
Counts objects inside a specific area. The object type (white or black) and dimensions (in pixel) have to be specified.
- **Measurements**  
Carries-out length measurements.



### Step 3 – Trigger selection

The trigger starts the inspection cycle (image acquisition, elaboration and result).  
The trigger is selected by clicking on the toolbox icon.



**Figure 18**

The user has three choices:

- *Continuous trigger*: the images are acquired and elaborated in the fastest speed possible. Elaborated the first image, a second image is immediately acquired and elaborated. And so on.
- *Hardware trigger*: the trigger signal is sent by an external device (e.g. a photocell).
- *Software trigger*: similar to the *Continuous mode*, but with one difference. This tool verifies a condition on the acquired image before carrying-out the complete elaboration. The *Software trigger* is represented graphically by a dashed line, where an arrow indicates the elaboration direction. This line identifies also the part of image elaborated. The full inspection runs only when a light variation appears along the line due to the passage from white zones to black zones and vice versa. Typically, this condition occurs when the target-object passes through (object must have a sufficient contrast with the background).

If the *Software trigger* is selected, the tool must be positioned graphically on the window.

Click on “Next” to continue.



## Step 4 – Locator selection

Locators track objects with respect to the template position and compensates for translation and rotation.

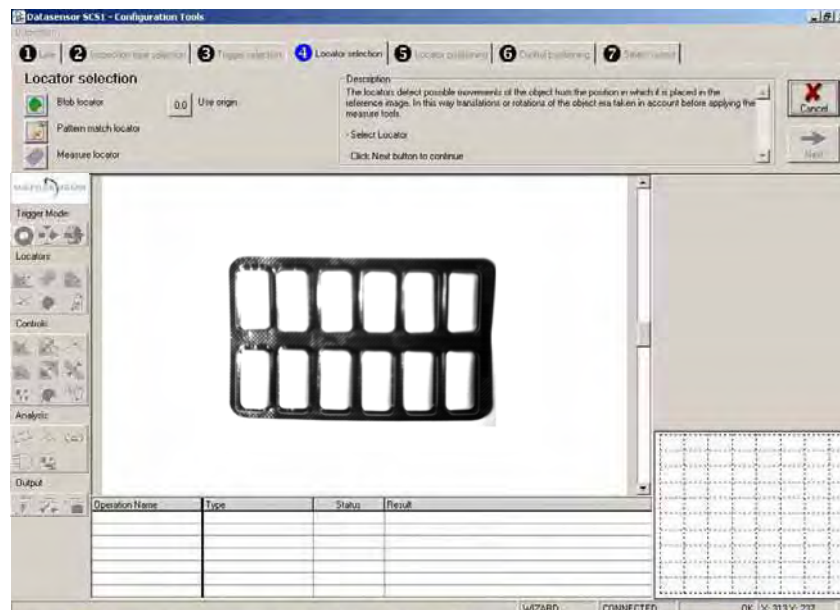


Figure 19

This step requires only the selection of the locator. The graph positioning is carried-out in a successive phase.

The selection is made clicking on the icon of the Wizard panel, using the description supplied. Click on “Next” to continue.

### Blob locator

**What is a Blob?** A group of adjacent pixels of the same colour, white or black.

**Description:** The *Blob locator* tracks inside a specific area the largest blob. *Blob locators* basically carry-out two functions: an initial binarization and a successive blob identification. Binarization allows the user to pass from a grey-tone image to a black and white image. Same-coloured pixel groups are gathered together and form blobs. The locator is represented graphically by a rectangle, that forms the ROI (*Region of Interest*) where the blobs are tracked. After the selection of the blob type to isolate, the locator tracks inside the ROI the largest blob, locating its barycentre coordinates, to be used in the successive steps as relative reference system's origin.

### Pattern match locator

Pattern match consists in checking the similarity between a template and a target-object. The *Pattern match locator* tracks a target-object inside a specific area and sets the position. The locator is represented graphically by two rectangles, one inside the other that respectively form the tracking area (external one marked in blue) and the target-object (internal one marked in orange). During inspection, the locator tracks inside the blue rectangle an image similar to the target-object in the orange rectangle.

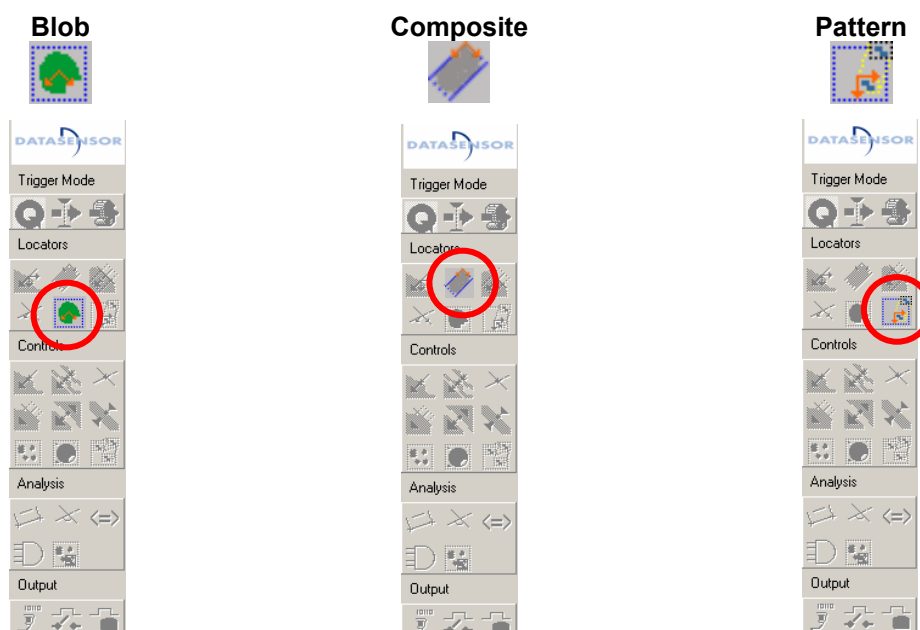
### Measurement location

Location is carried-out by a *Composite locator*. The *Composite locator* isolates the position of rectangular-shaped objects that have a specific contrast with the background. The tool has also a rectangular shape and has to be positioned enclosing the object to be tracked. For correct functioning, the locator must be slightly bigger than the target-object.



## Step 5 – Locator positioning

To graphically position the locator on the template, the user has to first select it clicking on the toolbox icon.



The Wizard provides instructions for locator positioning. At the end of the sequence, the user can modify the locator parameters present on the panel on the right side of the window.

Click on “Next” to continue.

The image shows a configuration window titled 'Blob count' with an 'Advanced' tab. The window contains the following fields and controls:

- Name:** A text box containing 'Blob Count' and an 'Apply' button.
- ROI:** A section with 'X' and 'Y' coordinates and 'Width' and 'Height' dimensions. The 'X' value is -182, 'Y' is -137, 'Width' is 566, and 'Height' is 421.
- Ref. System:** A dropdown menu set to 'Edge Locator'.
- Binarization:** A section with 'Object type' set to 'Bright object' and 'Threshold' set to 128. There is a checked 'Auto' checkbox next to the threshold.
- Count:** A section with 'Number of object' set to 1.



## Step 6 – Control positioning

Each inspection is associated to a control which has to be carried-out on the target-object. This step requires the insertion of the specific control.

To position graphically the control on the template, the user has to first select it clicking on the toolbox icon.

The Wizard provides instructions for control positioning. At the end of the sequence, the user can modify the control parameters present on the panel on the right side of the window.

Click on “Next” to continue.

### Contour match



### Area analysis



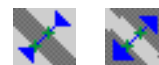
### Pattern match



### Object count



### Measurement

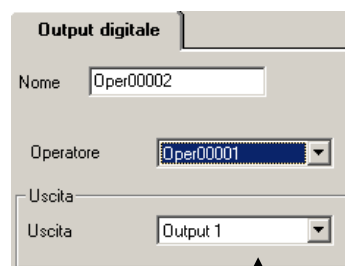




### Step 7 – Output selection

This last operation consists in associating the inspection result to one of the available outputs.

Select the Digital output tool from the Toolbox

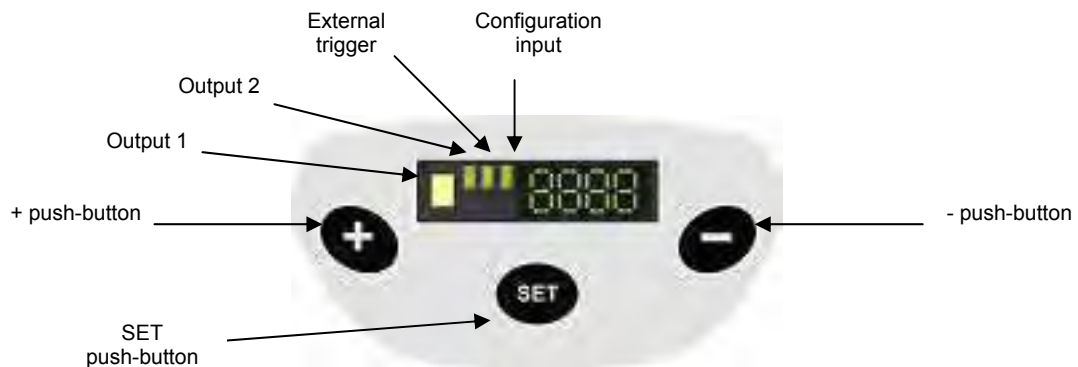


The guided creation is ending by clicking on “Next” and the user can save the inspection obtained.



## 5. SENSOR INTEGRATED USER INTERFACE

The sensor integrated user interface presents three push-buttons (SET, +, -), 4-digit display and 4 signalling LEDs.



- the SET, + and – push-buttons are used for sensor configuration
- the yellow LED indicates the sensor Output 1 status
- the three green LEDs are connected respectively to (from left to right):
  - Output 2
  - External trigger
  - Configurable input

### 5.1. Powering

At powering, the sensor enters in an initialisation phase that lasts for 25 seconds and then automatically carries-out an inspection.

During initialisation, the display visualises the following texts:

SCS1:



INIT (boot):



### 5.2. Factory configuration reloading








During initialisation there is the chance to reload the factory configuration.

Press + and – push-buttons at the same time when the display visualises **SCS1** text. Keep them pressed until **17 12** text is visualised and then for 3 seconds at least to start the factory configuration reloading process. When the system reset is over, the display will visualise **6222** text.



### 5.3. Sensor status

During the functioning the sensor can be in one of the following status:

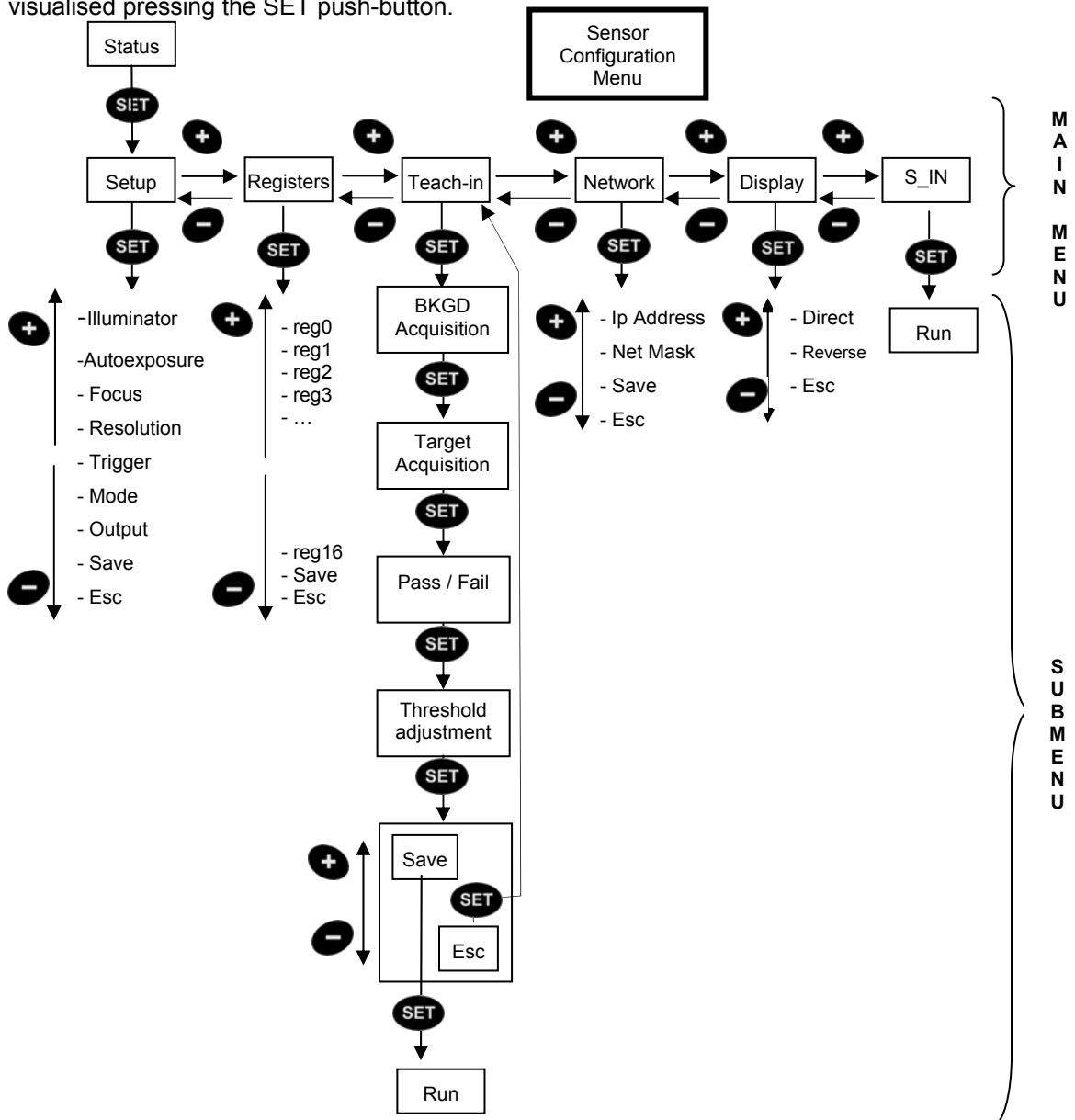
-  e2pr: sensor not configured;
- 2.  run1,  run2,  run3,  run4: an inspection is being carried-out. The number indicated the memory slot occupied by the inspection. The sensor has four memory slots that can contain four different inspections;
- 3.  eth: sensor connected to PC. A new inspection can be in the Setup phase or the sensor is acquiring images for the inspection elaborated on the PC;
-  err: error presence.

*Note:* during functioning, the sensor automatically carries-out a selected inspection. As default this inspection is the one in the first memory slot. All the inspections stored in the sensor can be selected, but only using the SCS1 Graphic User Interface via PC.



## 5.4. Sensor configuration menu

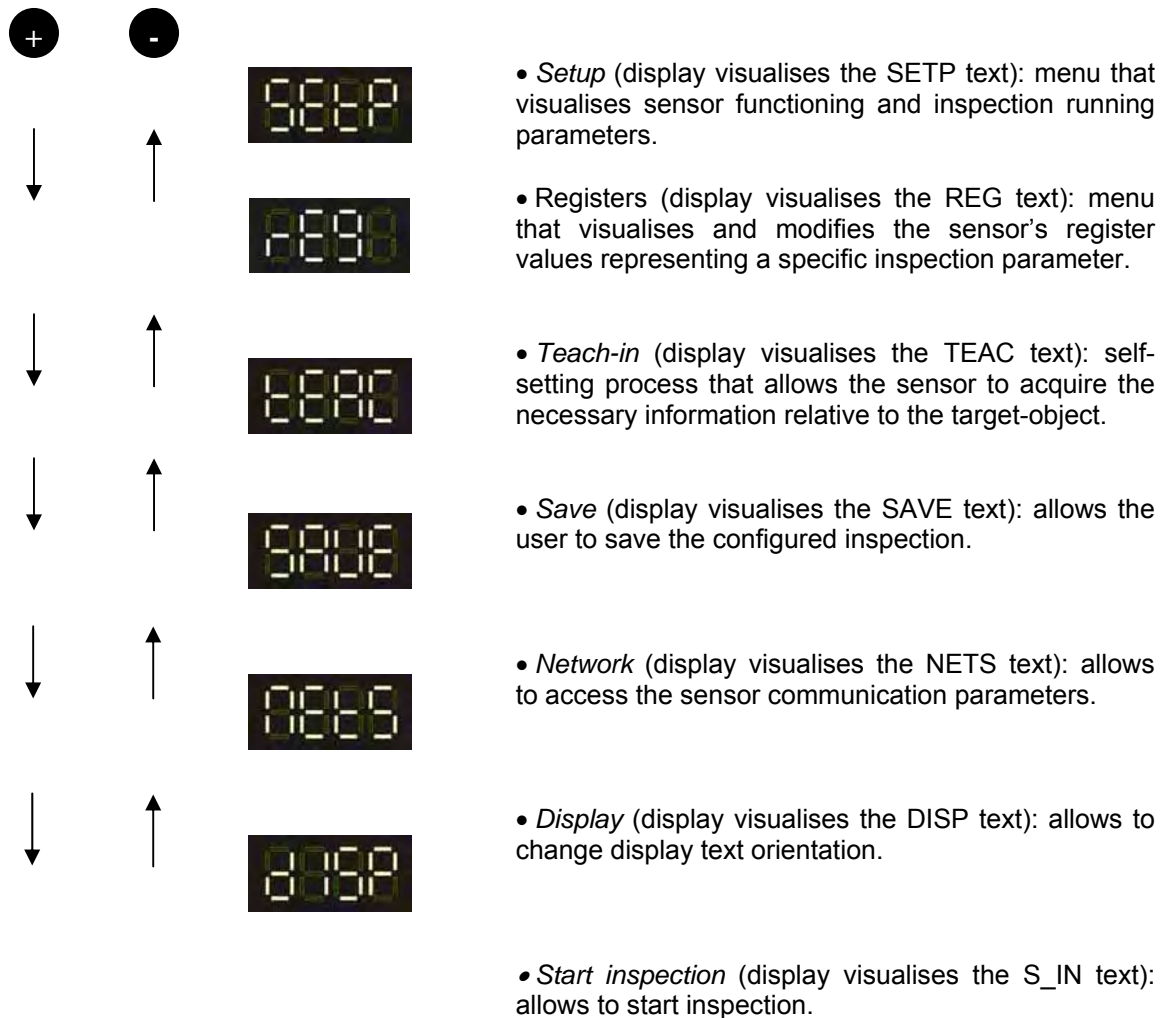
Pressing the SET push-button for 1 second the user enters the sensor configuration menu. The menu is divided in two levels, each presenting different options that configure the system parameters. The user can scroll down the other options using the + and – push-buttons. The submenus are visualised pressing the SET push-button.





### 5.5. Main menu

The first menu level represents the main menu, which is divided in seven categories, each accessing specific sensor parameters.

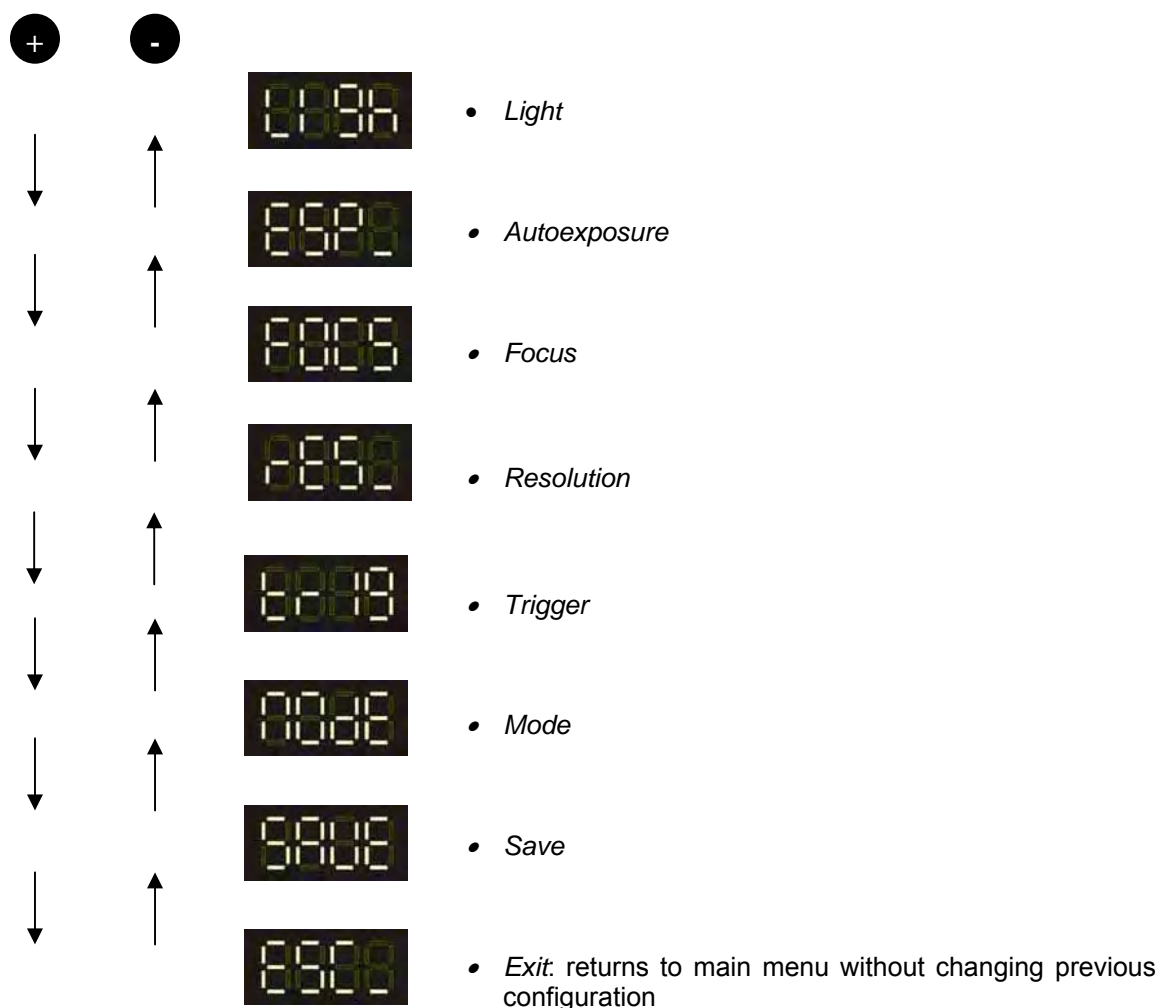


The user can scroll down the other options using the + and – push-buttons. The submenus are visualised pressing the SET push-button.



## 5.6. Setup menu

The Setup menu options are visualised by pressing the + and – push-buttons:



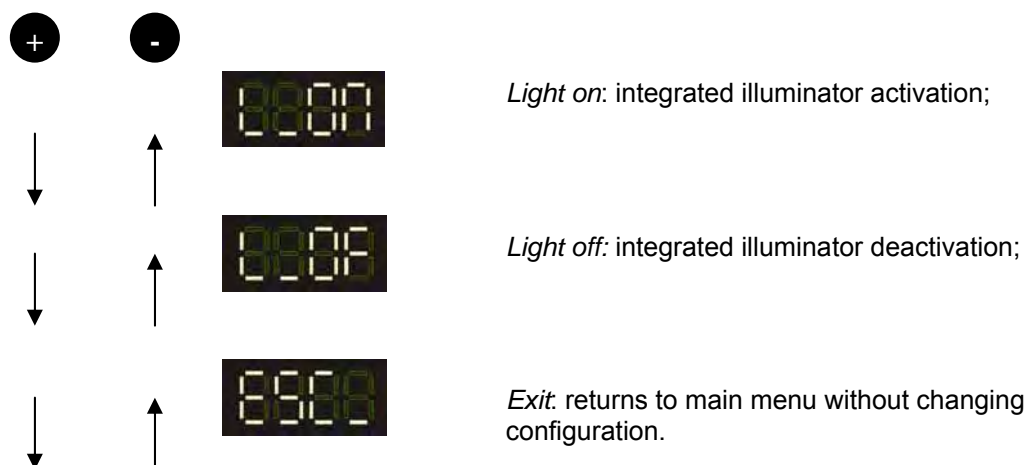
The option visualised on the display is selected by pressing the SET push-button. Some options are connected to a submenu, while others carry-out functions. The user can exit from the submenus in two manners:

- pressing SET while display visualises ESC (Exit). The configuration is left unchanged;
- pressing SET on any other option the selection is saved.



### 5.6.1. Light

Activates and deactivates the sensor integrated illuminator.



### 5.6.2. Autoexposure

Regulates automatically the sensor exposure time, adapting it to ambient light conditions.

- press SET when display visualises **ESP** to run autoexposure process;
- during process the display text blinks;
- terminated the autoexposure phase, the display visualises if the operation has been a success (SUCC) or a failure (FAIL);



- press SET to confirm and return to Setup menu, or to try again if the operation failed.

*Note:* to carry-out correctly the autoexposure process the target-object must be placed in the inspection area, simulating the operating conditions.



### 5.6.3. Focus

A numeric value is visualised on the display representing the focus quality. The focus can be regulated turning the shutter, positioned on the sensor lens. The value on the display is updated automatically. Higher the value, the better the focus.

Press SET again to return to the Setup menu.

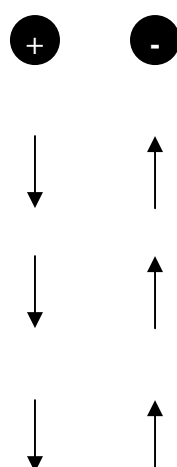
*Note:* the lens presents two rings: one for shutter regulation and the other for focus. Between the rings there is a white notch that acts as reference point. The focus ring is characterised by “near” text and “∞” symbol. Turning the ring making closer the reference notch and *near*, the sensor will progressively focus a closer plane. Turning the ring making closer the notch and ∞, the sensor will progressively focus a further plane.



### 5.6.4. Resolution

Two resolutions are available:

- Full VGA (640x480 pixels)
- ¼ VGA (320x240 pixels).



- ¼ VGA



- Full VGA



- *Exit:* returns to previous menu without changing configuration






### 5.6.5. Trigger

The *trigger mode* establishes the image acquisition instant. Two different triggers are available:

+  
↓  
↓

-  
↑  
↑





- *Continuous*: the sensor functions at maximum speed, starting a new elaboration as soon as the previous one is completed.
- *External*: trigger is supplied by an external device.
- *Exit*: returns to previous menu without changing configuration.

### External trigger

The *External trigger* is characterised by the following configurable parameters:

+  
↓  
↓  
↓

-  
↑  
↑  
↑

- *Filter*: minimum duration requested for the trigger pulse.
- *Polarity*: low-high level transition or high-low level transition.
- *Save*: saves the present configuration and exits from the trigger menu activating the *External trigger*.
- *Exit*: returns to previous menu without changing configuration.

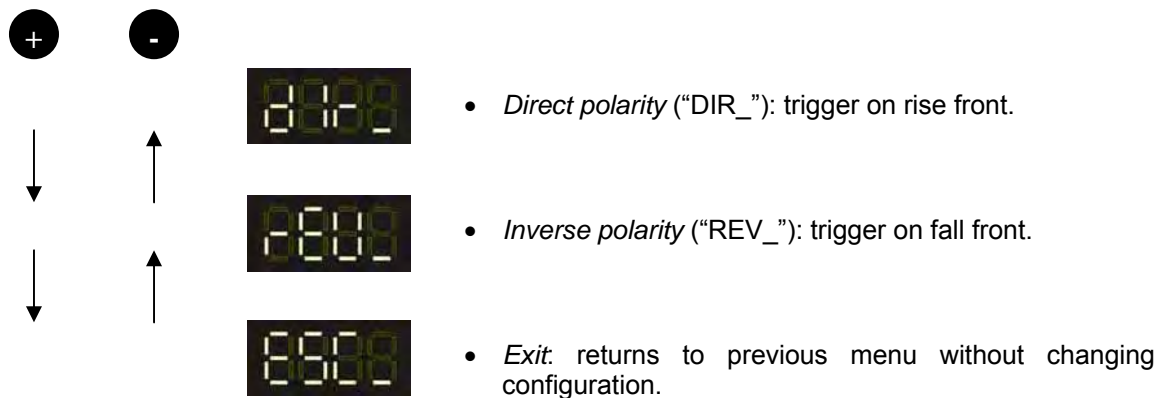
• *Filter*: short pulses due to disturbances can appear and can be confused with trigger signals. To avoid unwanted acquisitions, the user can set a minimum trigger pulse duration and so shorter pulses will not be considered.

A number representing the minimum trigger pulse duration will appear by selecting "FILT" (in ms). The minimum trigger duration can be varied using the + and – push-buttons. The user returns to the previous menu by pressing SET.

- *Polarity*: the trigger signal consists in an input voltage variation. This variation can vary from a high to a low level (fall) or from a low to a high level (rise).

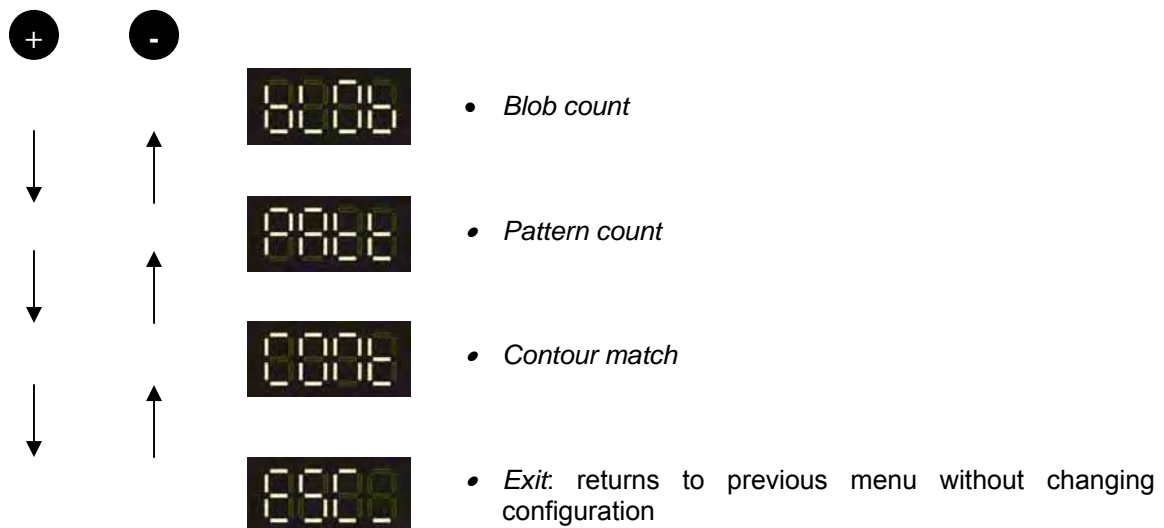


Another submenu is visualised by selecting "POL\_":



#### 5.6.6. Mode

The algorithm to be used during inspection is selected as follows:



#### Blob count

Algorithm that counts objects according to the *Blob analysis*, able to distinguish black objects on white backgrounds or vice versa.

- Algorithm is selected by pressing SET when display visualises
- A number automatically appears on the display representing the number of objects to find during inspection. During Teach-in, when target-object acquisition is requested (), the user must position in the sensor inspection area a number of objects equal to the saved number.



*Example:* 3 is set as the number of objects to count. The inspection parameters are configured and user enters in Teach-in. The background is acquired and then the target-object. Supposing that the object to count is a tap and 3 are the taps that I expect to find, then 3 have to be the taps to position in the inspection area during the target-object acquisition phase.

- The number can be changed by pressing on the + and – push-buttons.
- Press SET to confirm.



### Pattern count

Algorithm that counts parts according to *Pattern matching*, comparing the similarity between the template and target-object.


- Algorithm is selected by pressing SET when display visualises .
- A number automatically appears on the display representing the number of objects to find during inspection. Differently from *Blob count*, during Teach-in only one object has to be placed in the sensor inspection area when target-object is acquired () independently from the number of objects to be counted. Reached the acceptance threshold regulation phase, the user must position in the sensor inspection area a number of objects equal to the saved number.

*Example:* 3 is set as the number of objects to count. The inspection parameters are configured and user enters in Teach-in. The background is acquired and then the target-object. Supposing that the object to count is a postage stamp and 3 are the postage stamps that I expect to find, just 1 is the stamp to position in the inspection area during the target-object acquisition phase. During the acceptance threshold regulation phase, the output will be deactivated as the sensor expects to find 3 stamps. If 3 stamps are instead positioned in the inspection area (only during the threshold regulation, not during the acquisition phase), the output is activated and can thus switch.

- The number can be changed by pressing on the + and – push-buttons.
- Press SET to confirm.

### Contour match

Compares the template with the target-object according to the contour characteristics.

- Algorithm is selected by pressing SET when display visualises .

## 5.7. Registers

In the *Register* menu the value associated to the 16 registers, representing the tool parameter current inspection values, can be configured using the PC interface and that can be modified successively using the *Register* menu.

The numeric value can be modified in the *Register* menu, choosing the appropriate register:

- 
- 
- ...
- 

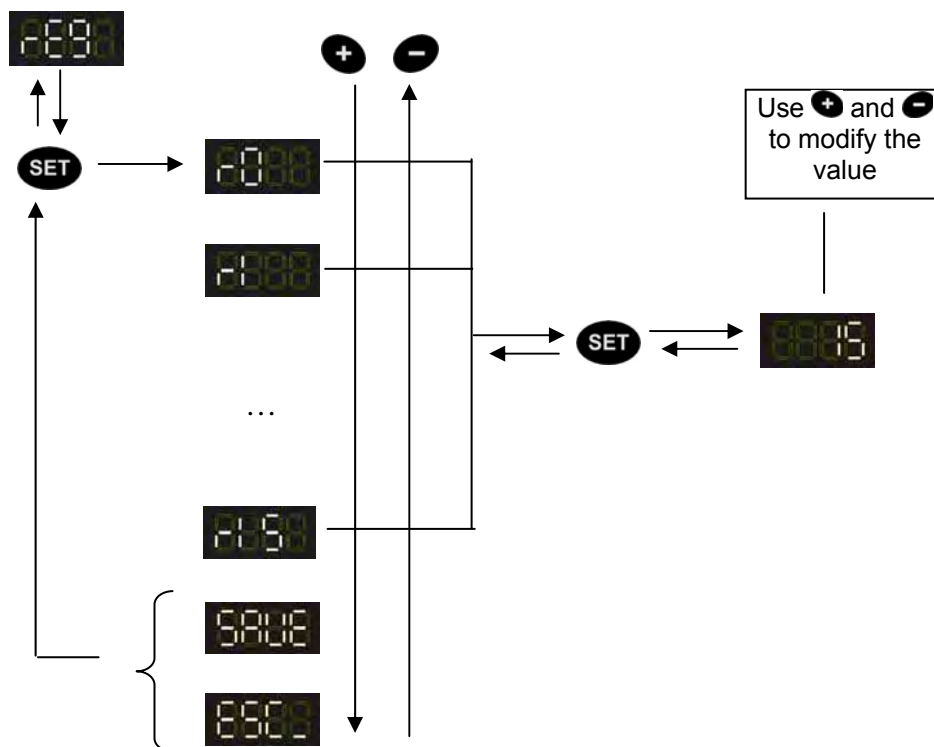
The <register>/<parameter type> association (e.g *reg1/blob count threshold*) is not indicated in the *Register* menu, but has to be remembered by the operator during inspection.



To modify the value of a register press **SET**. In this manner the current register value is visualised. To change it press **+** or **-**. To save press **SET** again (or to leave the value unchanged repress **SET** without making changes).

To save each modification in the register value **SAVE**, otherwise to discard changes press **ESC**.

The sequence is repeated for each value/register to be modified.

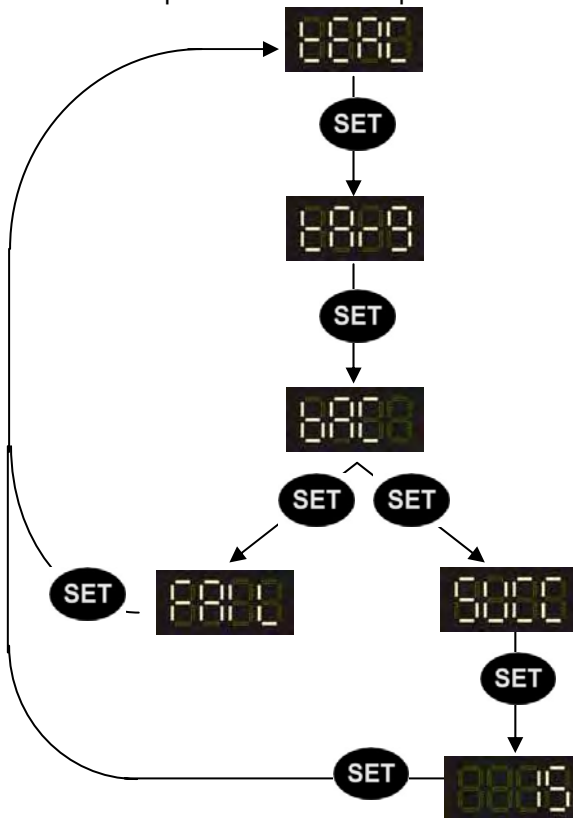




## 5.8. Teach-in

Self-setting process that allows the sensor to acquire the information relative to the target-object and its background.

The Teach-in process follows the phases listed below:



### Step 1 - "BAC"

Background acquisition. Press SET to acquire background and to pass to next phase.

### Step 2 - "TARG"

Acquisition of the object to detect. Position the sample in front of the sensor and press SET to acquire it and pass to next phase.

### Step 3 - "SUCC"/"FAIL"

According to the acquisition result, the SUCC or FAIL text is visualised. In case of failure press SET and return to main menu. In case of success press SET to pass to next phase.

### Step 4 - Threshold

Acceptance threshold regulation. The value visualised represents the acceptance threshold used during inspection. The lower the threshold, the more the sample and image have to correspond in order to obtain a successful result. Use the + or – push-buttons to change the threshold.

During threshold regulation, the sensor outputs are activated. Practically, the same conditions of the inspection running are present (*Run mode*), with however the possibility of changing the sensor sensitivity to distinguish or not an object.

Once acquired the information relative to the target-object, the sensor recognises the object and consequently the yellow Output 1 LED turns on, indicating inspection success.

The threshold can now be lowered until the LED turns off. The threshold has become so narrow that even the target-object is no longer recognised. A safety margin can be thus established increasing the threshold with respect to the constraint found.

The use of an integral target-object and a damaged one can allow the user to configure the threshold in order to recognise the integral target-object from the damaged one.

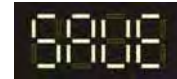
### Step 5 – Confirmation

Press SET to confirm the Teach-in sequence. The user returns automatically to the main menu. The inspection can be saved by scrolling down the options until reaching **SAVE** and selecting it. The sensor enters automatically in *Run mode*. Otherwise select **Quit** to run the inspection.



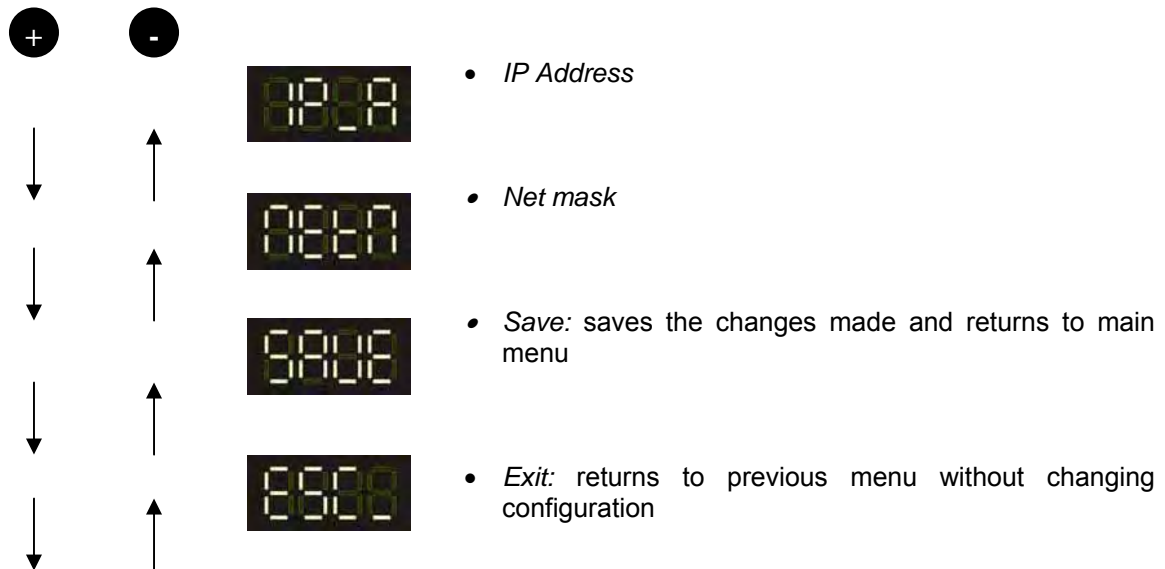
## 5.9. Save

Press SET to save the present configuration and pass automatically in *Run mode*. The inspection will be automatically saved in the first sensor's memory slot.



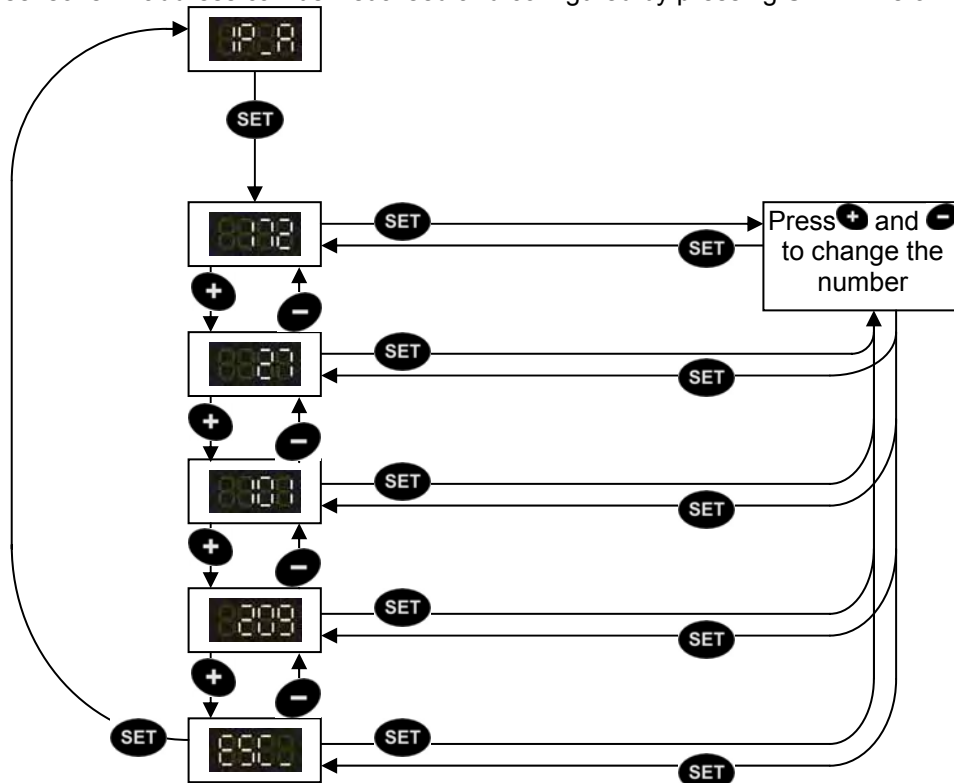
## 5.10. Network

Configures the sensor's *IP address* and *Subnet mask*.



### 5.10.1. IP address

The sensor's IP address can be visualised and configured by pressing SET while on "IP\_A":



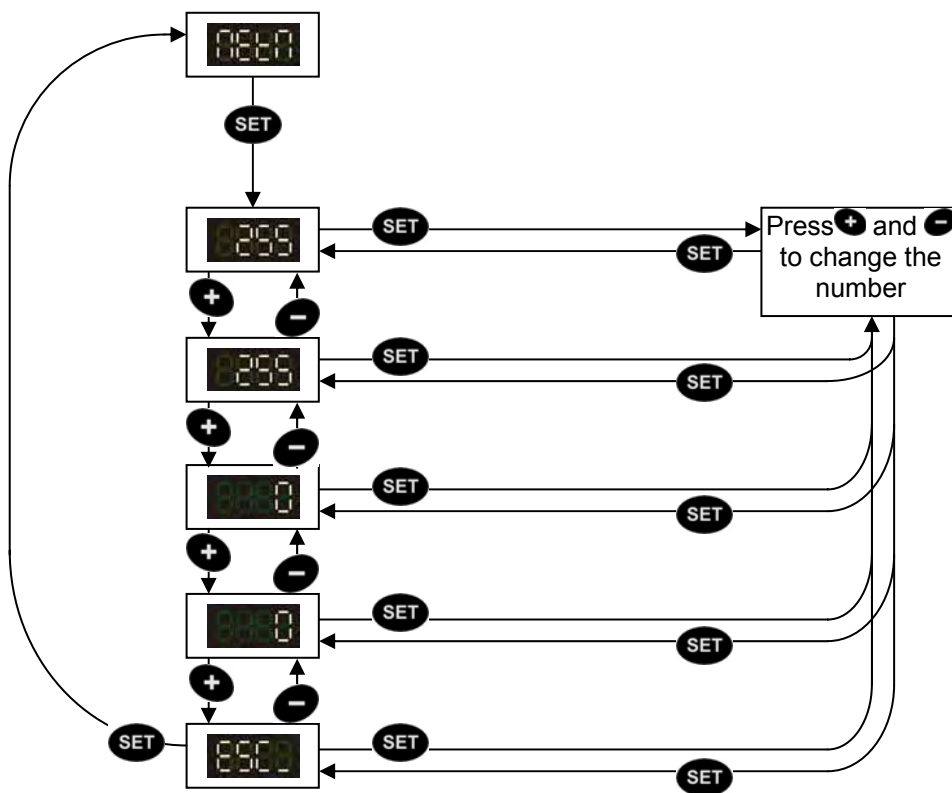


- The display visualises a 3-digit number which represents the first field of the sensor IP address.
- The 4 fields of the IP address can be visualised using + and -. The field currently visualised can be selected by pressing SET.
- Pressing again + and -, the numeric value of the selected field can be changed. A continuous pressure on the push-button allows to proceed more rapidly. Press SET to confirm changes.
- Press SET when **ESCU** text is visualised to return to the Network menu and decide whether save the new configuration (**SAVE**) or not (**ESCU**).

**Note: the changes made are effective only after sensor re-booting.**

#### 5.10.2. Net mask

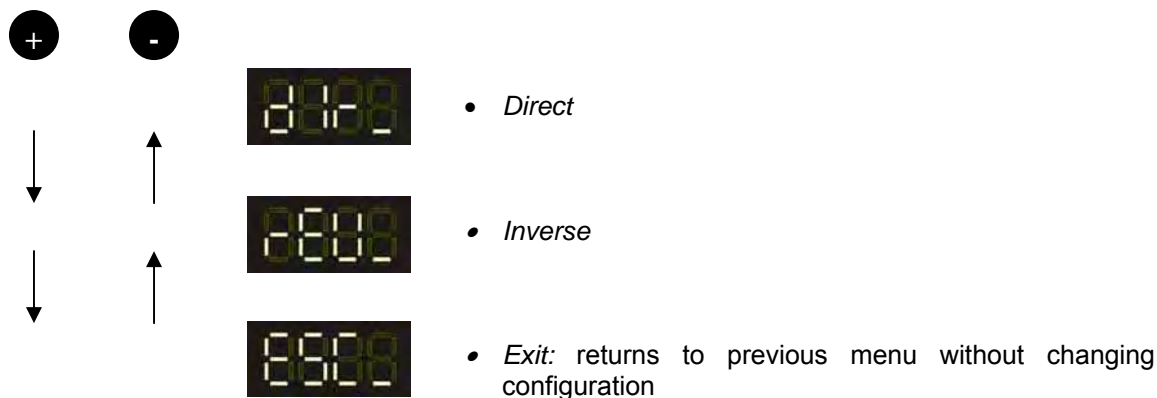
The *Net mask* can be configured by selecting "**NETM**". The process is similar to the IP address configuration.





### 5.10.3. Display

Changes the orientation of the text visualised on the display.



Example:



### 5.10.4. Start inspection

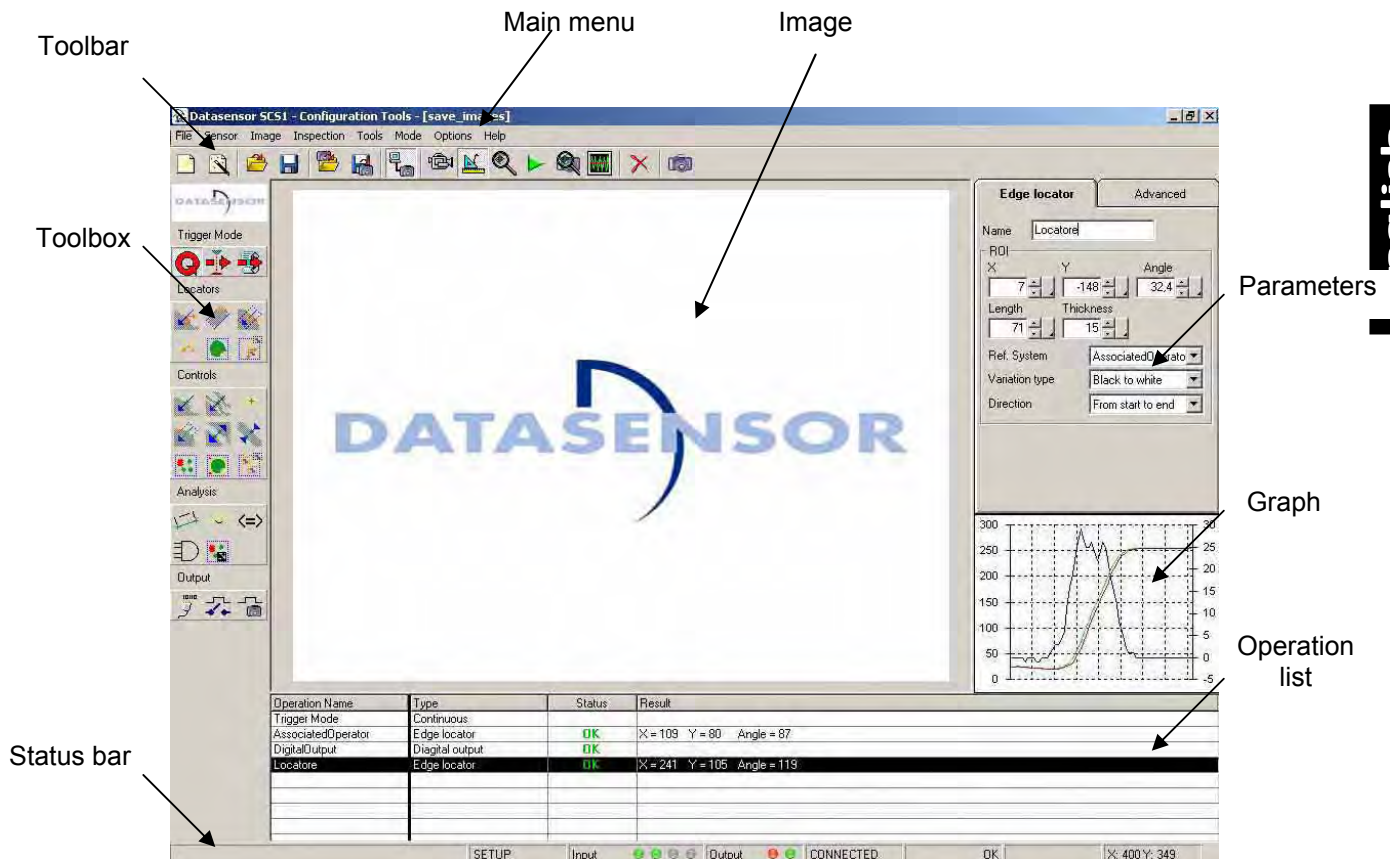
Press SET to enter in *Run mode* without saving the parameter changes.





## 6. SCS1 GRAPHIC INTERFACE

The SCS1 graphic interface allows sensor configuration through PC. The following figure shows the main window.



The main window contains the following panels:

**Main menu:** includes all the programme management options divided in categories.

**Toolbar:** contains all the most-frequently used functions present in the menu. The toolbar is pre-defined and can not be programmed by the user.

**Toolbox:** part of the window containing the square icons of the image elaboration tools divided in categories (*Triggers, Locators, Controls, Analysis, Output*).

**Image:** panel that visualises the images acquired by the sensor. Thanks to the *Off-line mode* (disconnected sensor and PC), the images visualised can be loaded on PC.

**Status bar:** panel containing some general information such as the system operating status, selected tool status, cursor coordinates.

**Operating steps:** this table indicates all the operating steps belonging to an inspection. The tool name, tool type, operating status (OK/Failure/Error) and the numeric result are highlighted for each operating step.

**Parameters:** panel containing parameters relative to the selected tool (during Setup) or relative to the template focus and acquisition (in *Live mode*).

**Graph:** represents the brightness distribution and its derivative.



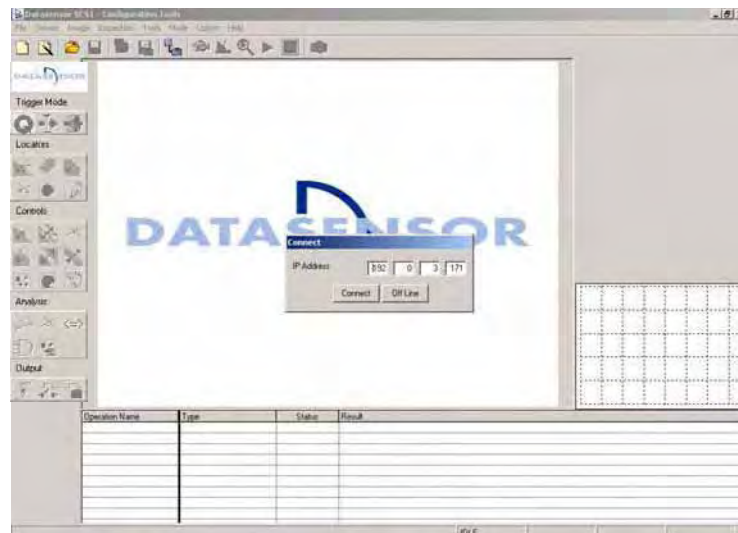
## 6.1. Functioning mode

The SCS1 graphic interface functions elaborating acquired images as well as operating on images saved on PC. Consequently *On-line* and *Off-line* modes are available.

*On-line*: Sensor and PC are connected. Images acquired by the sensor can be visualised, elaborated and saved on disk.

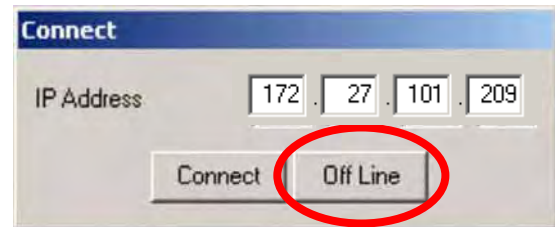
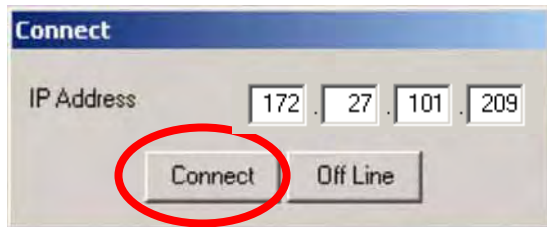
*Off-line*: Sensor and PC are disconnected. Previously saved images can be loaded and elaborated.

The functioning mode is selected at programming running, through the following window:



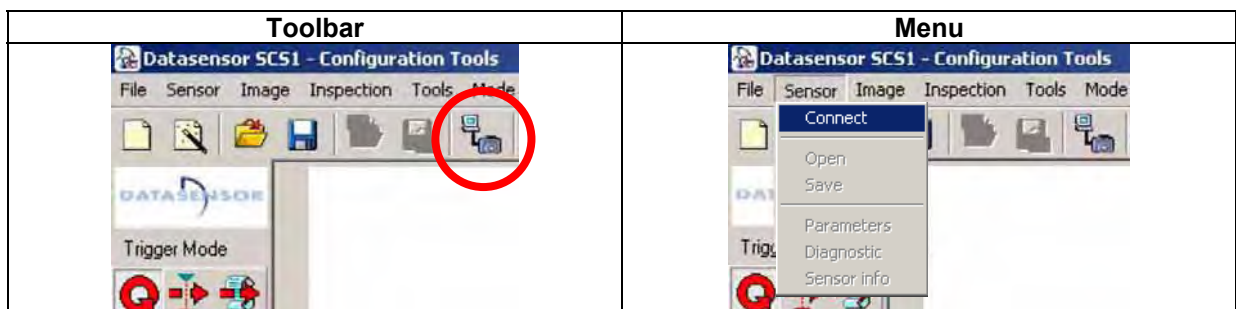
The On-line mode is selected by clicking on "Connect"

The Off-line mode is selected by clicking on "Off-line"



Please refer to paragraphs 3.3 / 3.4 for further details concerning the IP Address and connection problems.

The user can change mode at any time using the toolbar or menu.





## 6.2. System operating status

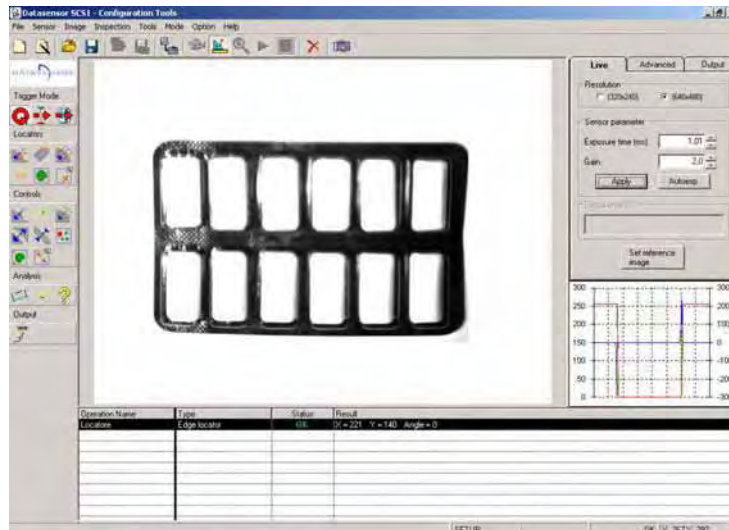
The system offers five different operating status, each carrying-out specific operations aimed at inspection configuration and optimisation. Each operating status is associated to an icon on the toolbar.

System operating status	
1) <i>Live mode</i>	
2) <i>Setup mode</i>	
3) <i>Test mode</i>	
4) <i>Run mode</i>	
5) <i>Test mode from telecamera</i>	
6) <i>Monitor mode</i>	

### 6.2.1. *Live mode*



During *Live mode*, the images acquired by the sensor are visualised on the image panel in real time. In this operating mode the user can modify all the image parameters in order to optimise quality.

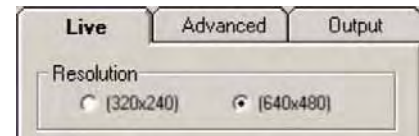


Live mode options		
Parameters	Advanced	Outputs
Resolution	Illuminators	Output1 mode
Exposure time	Trigger	Output2 mode
Gain	Image saving	
Focus level		
Reference image		

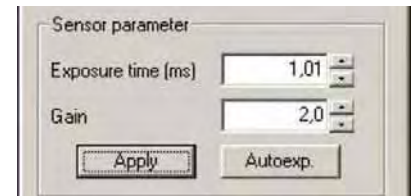


### Live mode parameters

**Resolution:** VGA (640x480 pixels) and ¼ VGA (320x240 pixels) resolutions are available.



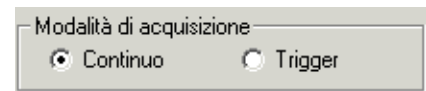
**Exposure time:** the duration of light exposure during the image acquisition phase. The exposure time can be set both manually (modifying the number corresponding to the duration in ms and clicking on "Apply") or automatically (clicking on "Autoexpose"). The higher the exposure time, the brighter the image.



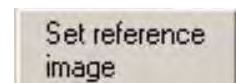
**Gain:** electronic enhance of the image signal. Gain increases image brightness without increasing exposure time. A larger gain corresponds to major noise and consequently the image quality decreases.

**Note:** the lens presents two rings: one for shutter regulation and the other for focus. Between the rings there is a white notch that acts as reference point. The focus ring is characterised by "near" text and "∞" symbol. Turning the ring making closer the reference notch and *near*, the sensor will progressively focus a closer plane. Turning the ring making closer the notch and ∞, the sensor will progressively focus a further plane.

**Acquisition mode:** configures the reference image using the trigger hardware, if set in the *trigger* mode. For example, if a dynamic scene is controlled, a particular phase is difficult to obtain by pressing the "saving the reference image" icon. The acquisition of the reference image in the *trigger* mode and, using a trigger hardware, has to be set and then save the singular phases of the inspected scene until reaching the necessary configuration. In this case, pressing "saving the reference image" and providing another trigger pulse, a particular image can be saved as our template.



**Reference image:** the image used to configure inspection. The image visualised on the window is saved by pressing "Set reference image". The image will be available during the Setup phase.



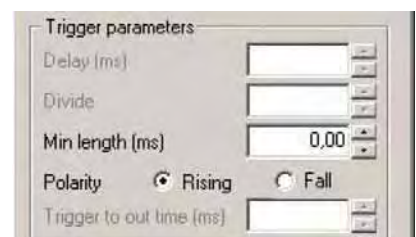
### Advanced parameters

**Illuminators:** an external illuminator can be activated or deactivated. The strobe function can also be activated or deactivated.



**Trigger parameters:** parameters that refer to the external trigger and that do not affect the *Continuous mode* and *Software trigger*.

- Minimum trigger pulse duration: short pulses due to disturbances can appear and can be confused with trigger signals. To avoid unwanted acquisitions, the user can set a minimum trigger pulse duration and so shorter pulses will not be considered.





- Polarity: the trigger signal consists in an input voltage variation. This variation can vary from a high to a low level (fall) or from a low to a high level (rise).

Selecting “fall” or “rise” the variation type to consider as trigger is chosen.

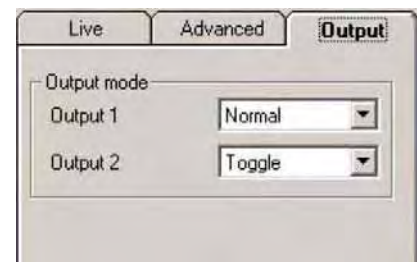
*Image saving:* the elaborated images can be stored in the sensor memory during inspection and visualised successively in *Test mode* to check tool elaboration functioning. Different saving modes can be selected: saving all elaborated images, only those which have generated an inspection failure or none. Up to 10 images can be saved. Surpassed this limit, the new images are saved over the existing ones, following a FIFO logic: the oldest image is replaced by the latest image.



## Outputs

Two digital outputs offer inspection information. These outputs can be configured as follows:

- *Normal functioning:* output switches according to inspection results (OK/Failure), if tools have been correctly configured.
- *Toggle mode:* output switches at each inspection cycle completion.



## Image saving

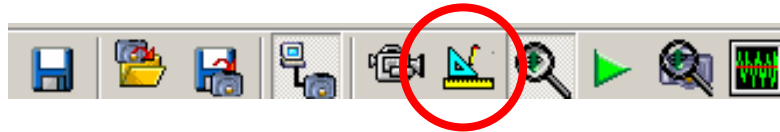
The currently visualised image can be saved in the *Live mode* in a Bitmap format, by simply clicking on the toolbar icon:



or selecting “Save image” in the Image menu.



### 6.2.2. Setup mode



The image elaboration tools can be included in this mode. Each tool represents operating phases that form the inspection. Please refer to paragraphs 6.4 and 6.5 for more complete tool details.

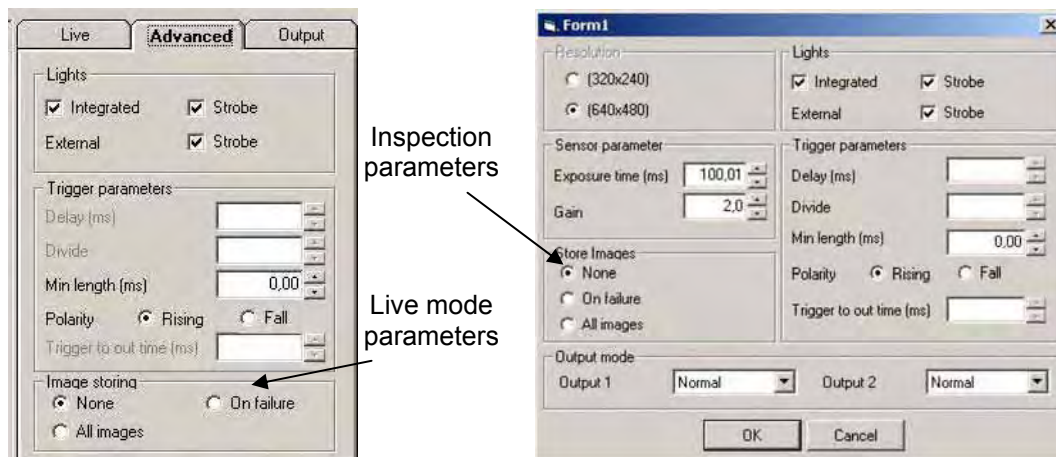
### 6.2.3. Test mode



This mode analyses the images acquired and elaborated during inspection. The user can save the images elaborated during inspection using the advanced parameters of the *Live mode* or selecting "Inspection parameters" from the "Inspection" menu. The user can save all elaborated images, only images that have generated an inspection failure or none at all. The images are stored in the sensor memory and are available for the *Test mode*.

The following steps indicate how to access the images once saved:

- Ensure that during the Setup phase the image saving option has been selected:



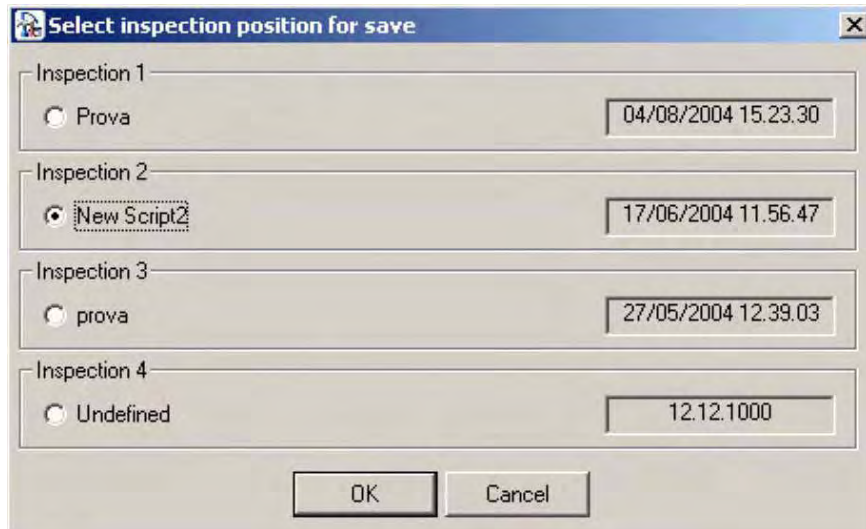
- Run inspection on the sensor (disconnected from PC) to elaborate images.
- Re-connect sensor and upload the inspection from the sensor, clicking on the toolbar icon:



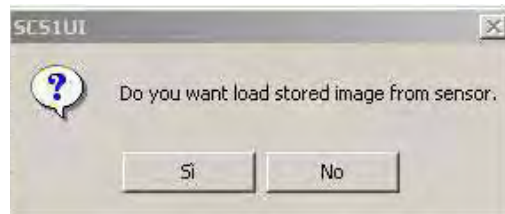
or select "Open" from the sensor menu.

Select the desired inspection from the new window and click on OK.

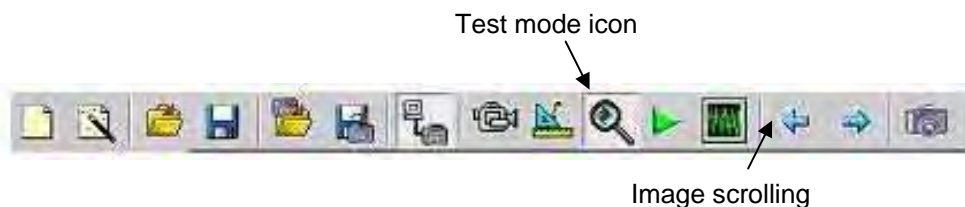




- A new window appears requesting if the images saved with the inspection have to be opened. Click on "Yes" to confirm.



- At this point the images available for the test are available. The user can pass to the *Test mode* using the toolbar icon. Once in *Test mode*, two light-blue arrows on the toolbar allow the user to scroll through the images. Maximum 10 images can be saved in a FIFO logic: the oldest image is replaced by the latest image.



*Note:* once the SCS1 graphic interface is closed the images are automatically cancelled. To access them again for future needs, the current inspection must be saved on file after loading the images from the sensor. In this manner, the user can access the test images each time the inspection is re-opened.

#### 6.2.4. **Run mode**



The PC processes in real time the images acquired by the sensor. In this operating mode, the sensor only acquires images. The operating steps can not be selected or modified.

**NOTE:** if you set up the hardware trigger, you must give trigger impulses in order to process images.



### 6.2.5. Test Mode from telecamera



Mode that analyses the last image acquired and elaborated during inspection.

This tool can be used in two different modes:

#### 1. First mode:

The same as per *Test Mode* and it is thus necessary to set the inspection image saving as described in paragraphs 1 and 2 of section 6.2.3 describing the *Test Mode*.

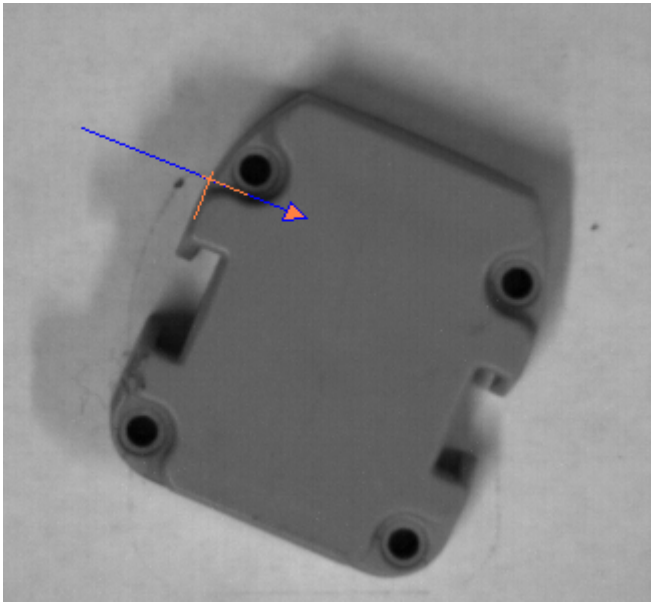
#### 2. Second mode:

This mode uses the *Image Conditioned Memorisation* tool (refer to section 6.10). This tool allows to save during inspection an image currently acquired and elaborated, according to the tool status. For example, if the *Image Conditioned Memorisation* tool is configured, in order to save the image when a tool fails. When the device is reconnected, the user can control the image of the last inspection which has caused the tool error by clicking the *Test Mode from Telecamera* icon.

The following example should better clarify this mode:

1. An inspection is set (even without saving the images in the *Advanced* menu of the *Live* mode using a Locator or Control tool). If the images has to be saved only when the tool chosen fails (as in the example), a *logic tool* has to be used to carry-out the *not* status of the operator tool

Logic tool for the *not* status of Edge locator




Tool for image saving

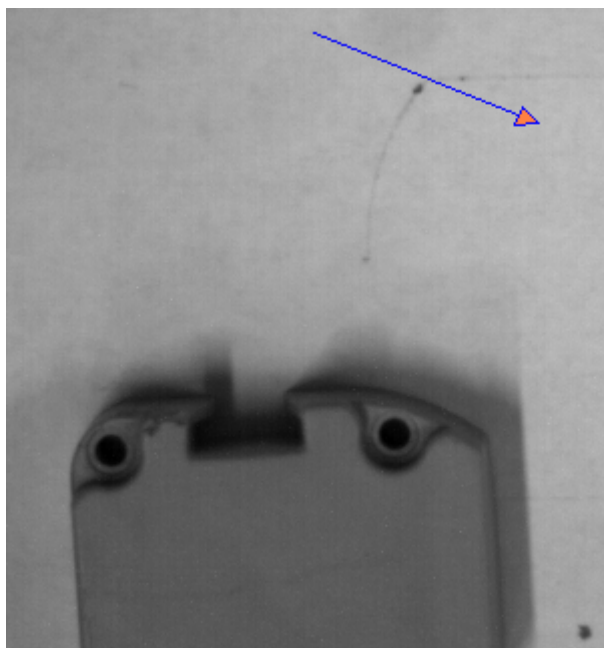
Tool that manages image saving

Nome operazione	Tipo	Stato	Risultato
Trigger Mode	Continuo		
Edge Locator	Locatore a edge	OK	X = 275 Y = 196 Angolo = 22
Not Edge Locator	Operatore Logico	FALLIMENTO	
Save IF(Not Edge Locator)	Salva immagine		


2. The Image saving tool can be set associating it the tool used or to the *not* status logic tool, depending if the image has to be saved in the OK or FAIL status.
3. The inspection is sent to the device according to the procedure described in page **Errore. Il segnalibro non è definito..**



4. When the image saving condition is satisfied, the image can be visualised reconnecting the device and clicking on *Test Mode*  from *telecamera icon*. The last image saved by the SCS is thus visualised on the main screen of the graphic interface.



Nome operazione	Tipo	Stato	Risultato
Trigger Mode	Continuo		
Edge Locator	Locatore a edge	<b>ERRORE</b>	
Not Edge Locator	Operatore Logico	<b>OK</b>	
Save IF(Not Edge Locator)	Salva immagine		

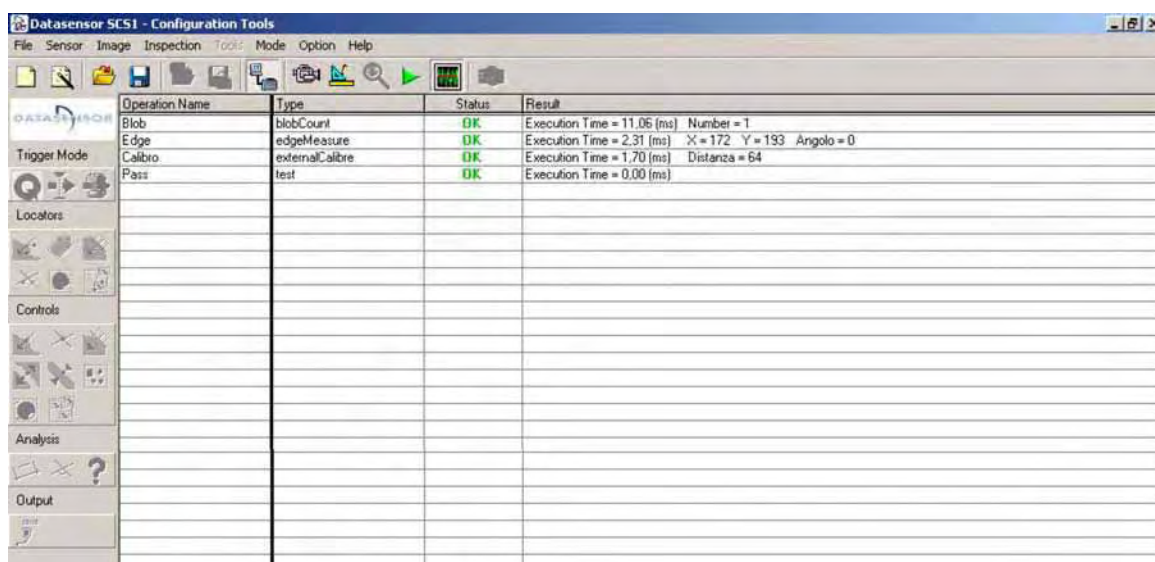
The images show what the device visualises clicking on the *Test Mode from telecamera icon*. In this case, the inspected object has been moved and thus the *Edge locator* fails and the *logic operator* enters in the OK status. The image currently elaborated and visualised is saved clicking on the  icon.



### 6.2.6. Monitor mode



The *Monitor mode* is used to evaluate the current inspection performance. The sensor elaborates the acquired images and transmits the results obtained to the PC. The images acquired are not visualised on the window but the results are given in a text format. During sensor functioning, tool information such as status, results and execution time is visualised.



The user has to save an inspection on the sensor before passing to the *Monitor mode* and so activate the elaboration.

NOTE: if you set up the hardware trigger, you must give trigger impulses in order to process images.



## 6.3. Menu and toolbar

### 6.3.1. Main menu

The main menu is divided in eight categories, each accessing specific sensor parameters.



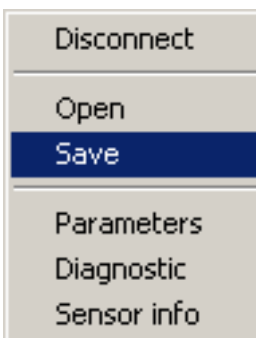
The menu options are listed below. “\*” indicates the toolbar options while “\*\*\*” indicates the toolbox options.

#### File



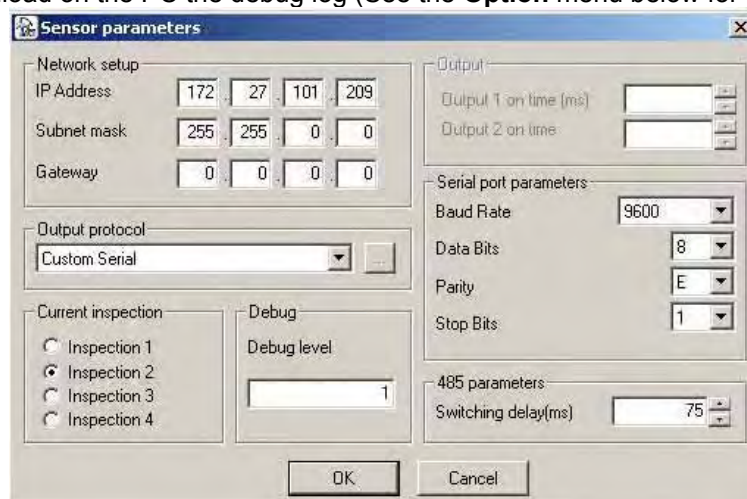
- *New inspection (\*)*: creates a new inspection.
- *Guided inspection creation (\*)*: creates a new Wizard-guided inspection.
- *Open (\*)*: opens an existing inspection.
- *Save(\*)*: saves a inspection on file memorising images.
- *Save as*: saves a inspection assigning a different name (respect to the one specified).
- *Exit*: closes the SCS1 graphic interface.

#### Sensor



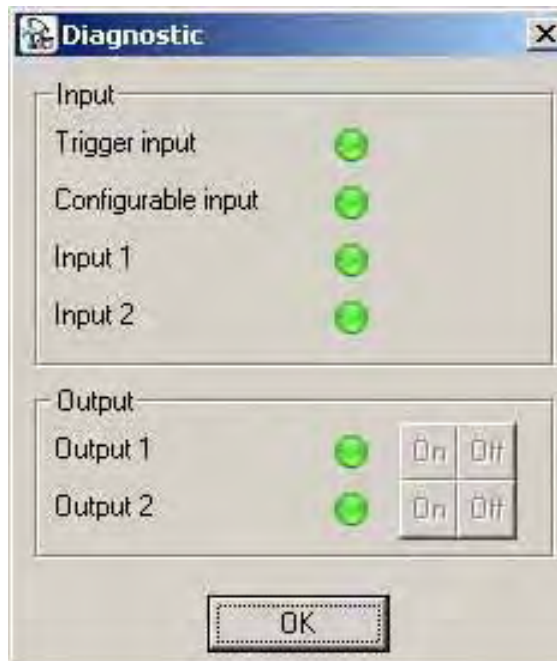
- *Connect/Disconnect*: connects/disconnects sensor from PC.
- *Open (\*)*: opens an existing inspection.
- *Save (\*)*: saves a inspection on sensor memory.
- *Parameters*: provides access to sensor parameters.
- *Diagnostics*: provides access to the sensor diagnostics.
- *Sensor info*: provides access to default sensor information.

The current inspection (inspection that also runs automatically at powering) and serial port can be configured by selecting “Parameters”. Here you can also configuring the Debug level, required if you want to download on the PC the debug log (See the **Option** menu below for more details).

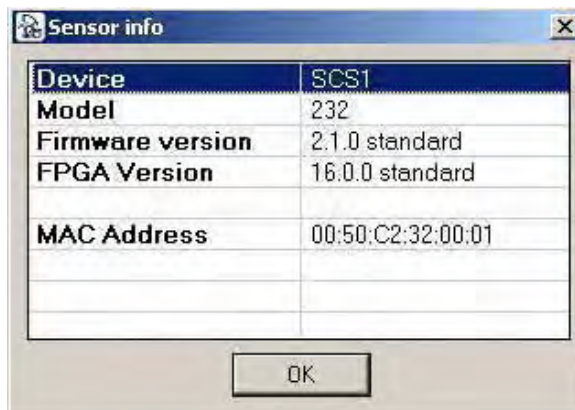




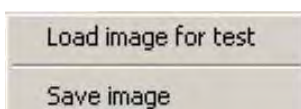
Selecting "Diagnostics" the following window appears enabling the control of the digital inputs status (trigger, configurable input, input 1 and input 2) as well as the digital outputs (output 1 and output 2). The control of the correct activation/deactivation of the digital outputs using the specific ON/OFF icons.



The sensor parameters, such as *device name*, *model type* (digital input, RS485, RS232), *firmware version* and the *MAC Address* of the device network interface, can be controlled by selecting "Information".

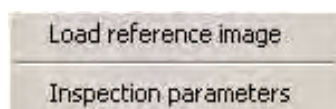


## Image



- *Load image for test*: load the image inside the group of images available for test. The image can be an acquired image or a Bitmap file (with 648x480 or 320x240 resolution, grey-scale image).
- *Save image*: saves current image on file in a Bitmap format.

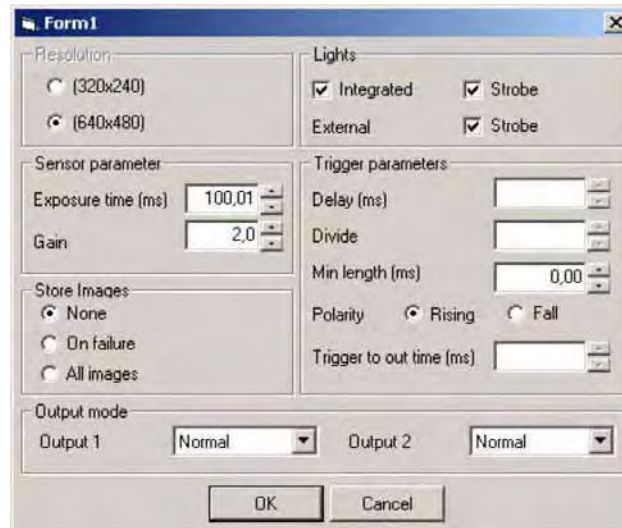
## Inspection



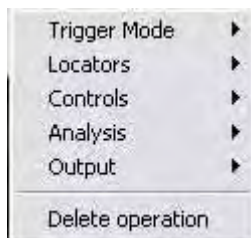
- *Load reference image*: a previously acquired and saved image can be used as a reference image.
- *Inspection parameters*: provide access to inspection parameters.

The parameters available also in *Live mode* can be accessed (and in some cases modified) by selecting "Inspection parameters" (refer to paragraph 6.2.1):



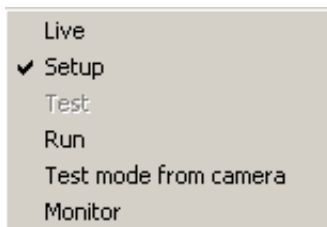


## Tools



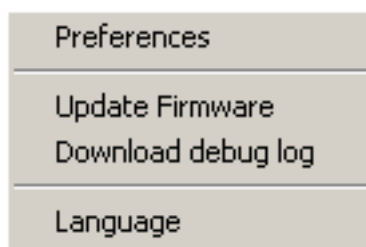
- *Trigger mode* \*: list of *Trigger modes* available.
- *Locators* \*\*: list of locators.
- *Measurements* \*\*: list of measurements.
- *Analysis* \*\*: list of analyses.
- *Outputs* \*\*: list of outputs.
- *Cancel operation* \*: cancels option selected.

## Mode



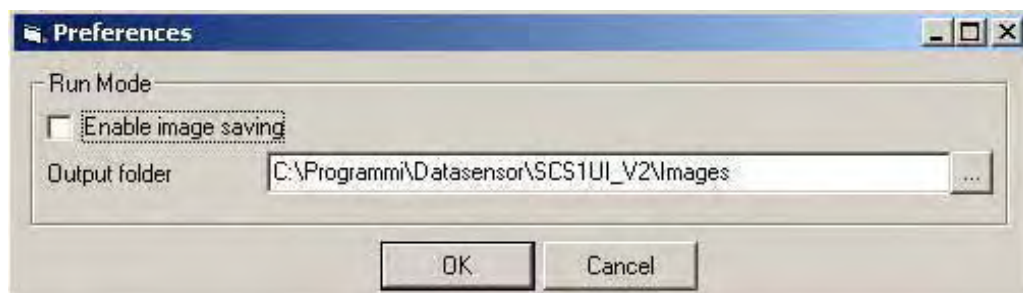
- *Live* \*: programme enters in the *Live mode*.
- *Setup* \*: programme enters in the *Setup mode*.
- *Test* \*: programme enters in the *Test mode*.
- *Run* \*: programme enters in the *Run mode*.
- *Test mode from telecamera*: programme enters in the *Test mode from telecamera*.
- *Monitor* \*: programme enters in the *Monitor mode*.

## Options



- *Preferences*: provides access to preference menu.
- *Update Firmware*: runs firmware update.
- *Debug file download*: saves the debug file on the PC according to the debug set.
- *Language*: selects graphic interface language.



**Preferences:**

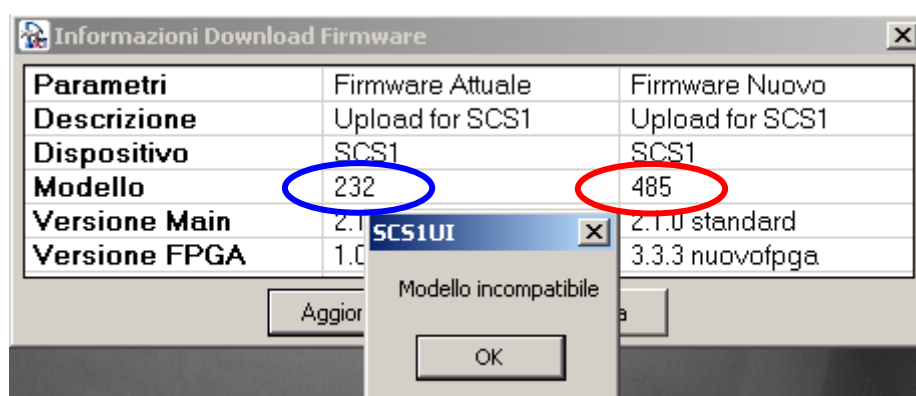
*Activate image saving:* clicking on this option, the user can save the images acquired by the device on your PC in the *Folder* specified in the text box. Setting the *Run Mode*, using the graphic interface, the device can store up to 10.000 images. The successive images saved by e device will overwrite the images previously saved. This supplies another analysis tool if anomalies occur during inspection.

**Firmware update:**

The following information is visualised clicking on *Firmware Update*:

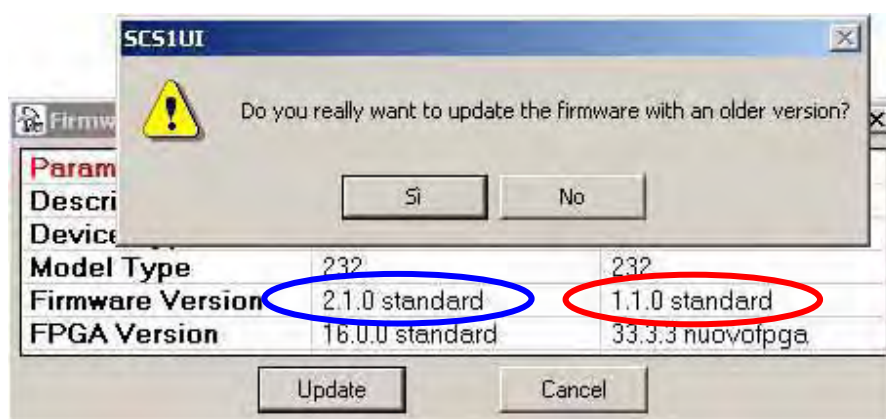


The specifications of the installed *firmware* can be controlled and compared with the new version to be saved. The new SCS1 version offers new controls to ease the *firmware* update procedure. The new controls concern the impossibility to update a particular SCS1 model (e.g. RS232) with a different *firmware* model(e.g. RS485) as shown in the following image:

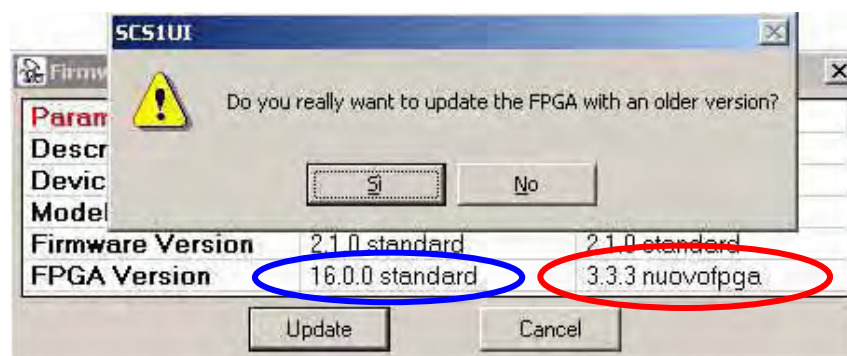


A control on the same *firmware* version has been added, including a warning during the update of a SCS with a *firmware* version **previous** to the default version or, previous to the *firmware* currently installed:

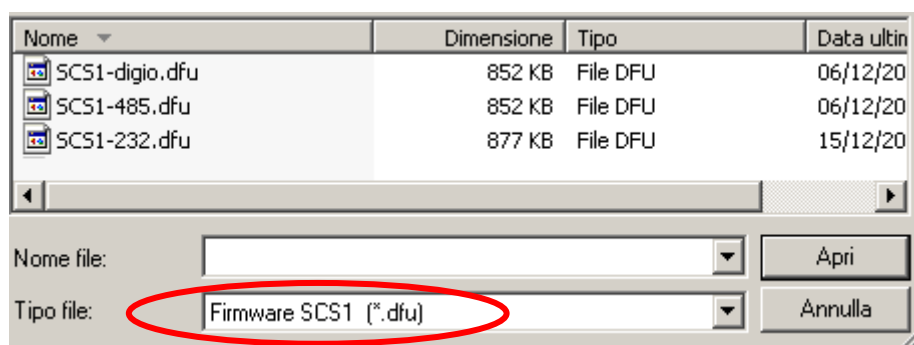




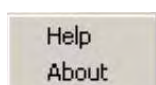
Lastly, a control on the *bitstream* version of the *FPGA* has been included. If the user tries to install an “older” *bitstream*, the updating procedure will be warned with a notice similar to the one previously described:



*Note:* Beginning from this new Graphic Interface version, the package extension of the *firmware* updated will no longer have the “.fw” ending but the “.dfu” extension:



## Help

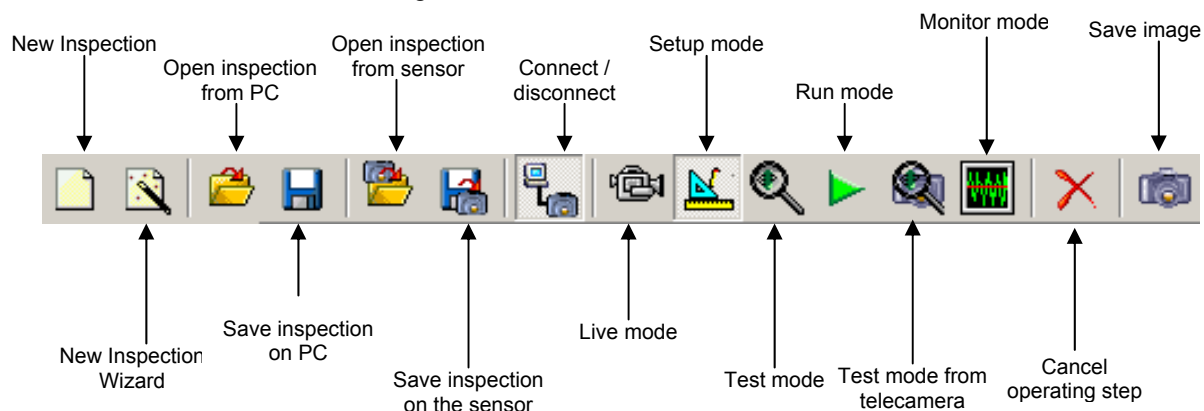


- *Guide*: opens programme’s help on-line.
- *Information* : contextual information.

## 6.3.2. Toolbar



The toolbar contains icons for quick access to all the most-frequently used functions. The toolbar must not be confused with the toolbox. The former provides access to the system's general functions, while the latter is used to select the image elaboration tools.





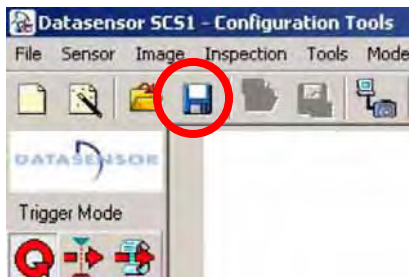
## 6.4. Inspection saving and configuration

The inspections configured through the SCS1 graphic interface can be saved. The sensor can store up to four inspections that can be recalled easily and quickly.

### 6.4.1. PC saving

To save the inspection on PC, the user has to simply click on the toolbar icon or select "Save" from the File menu and assign a name when requested.

**Toolbar**



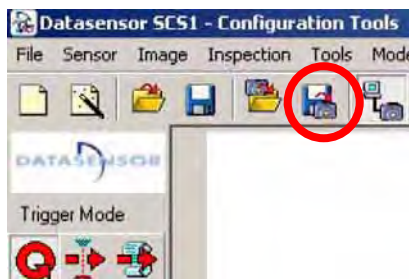
**Menu**



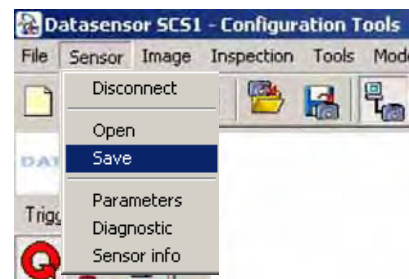
### 6.4.2. Inspection transfer and sensor saving

The sensor has four memory slots that can store four different inspections. To save the inspection on the sensor, the user has to simply click on the toolbar icon or select "Save" from the sensor menu. The program will request PC saving, if not previously done. This procedure is recommended but not compulsory.

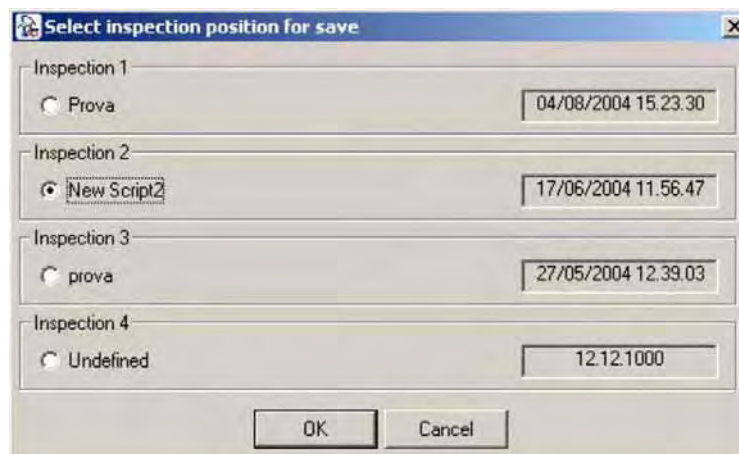
**Toolbar**



**Menu**



The following window is visualised:



Select the memory slot where the inspection saving is desired and click on "OK".



## 6.5. Trigger mode

In presence of moving objects inspections, the image acquisition instants have to be defined to guarantee the presence of objects to detect in the inspection area during acquisition. The system has three *Trigger modes* in order to best suit the different application needs.

### 6.5.1. Hardware trigger

An external device (e.g. photocell) sends the trigger that establishes the image acquisition instant and thus runs the inspection cycle.

### 6.5.2. Continuous mode

The images are acquired and elaborated at the highest rate possible. The sensor acquires a first image and elaborates it, then a second image is immediately acquired and elaborated. When ready, starts another acquisition, repeating the cycle continuously. The acquisition of the current image acquisition is made at the end of the previous inspection cycle.

The *Continuous mode*, thanks to the *Run mode*, is useful to test a recently-configured inspection guaranteeing constant updating of the results obtained independently from the information contents of the acquired images. Hence, the number of acquired images per second is limited to the current inspection duration and the images are not connected to the object's position.

### 6.5.3. Software trigger

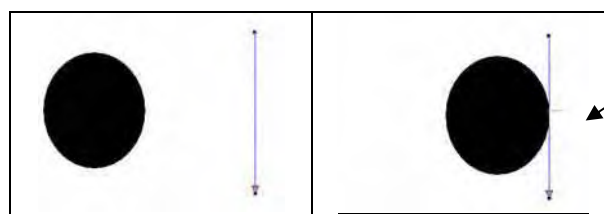
The *Software trigger* inserts graphically a software tool that conditions the elaboration of the acquired images.

The image acquisition and elaboration cycle is similar to the *Continuous mode*, but with one difference. Acquired the image, the sensor has to be satisfy the condition established by the *Software trigger* before completing the elaboration. The condition is usually connected to the position of the target-object. In this manner, inspection is more reliable and time is optimised as only meaningful inspections are carried-out.

**Description:** the *Software trigger* can be compared to a photocell sensitive to light variations of the acquired image.

The *Software trigger* is represented graphically by a dashed line, where an arrow indicates the elaboration direction. This line identifies also the part of image elaborated.

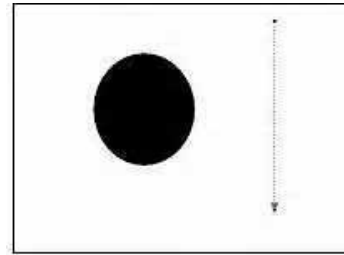
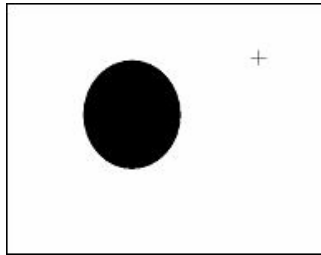
The elaboration runs when a light variation appears along the line due to the passage from white zones to black zones and vice versa. Typically, this condition occurs when the target-object passes through (object must have a sufficient contrast with the background).



The Software trigger detects brightness variation due to the black object passage on the white background.



**Positioning:** The *Software trigger* is represented graphically by selecting two points on the image: the first set pressing the left mouse button and the second releasing the button.



Once positioned, the trigger can be moved and rotated along one of its vertex. The movement is carried-out using the left mouse button in correspondence to the object's field of view and dragging it to the new position. The movement is represented by a dashed line that identifies the new object position. The definite drawing is obtained only when the button is released.

If the dragging begins from one of the arrow's vertex, only the position of one vertex is changed and the entire object remains in the same position. The movement is represented again by a dashed line that follows the mouse movements. During rotation the object length can change.

The user can use one *Software trigger* that automatically becomes the first inspection operating step.

*Example:*

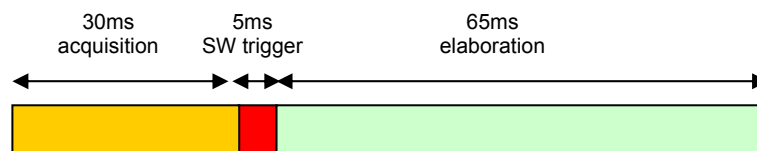
*Continuous mode – Software trigger comparison*

Supposing to configure an inspection that lasts 100 ms using the *Software trigger*.

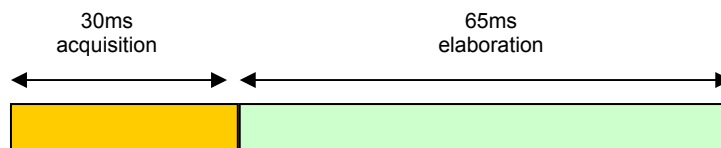
The time necessary to acquire an image is about 30 ms.

The *Software trigger* elaboration time is about 5 ms.

The remaining 65 ms are dedicated to image elaboration (e.g. area evaluation).



The same inspection carried-out in *Continuous mode* would last 95 ms:

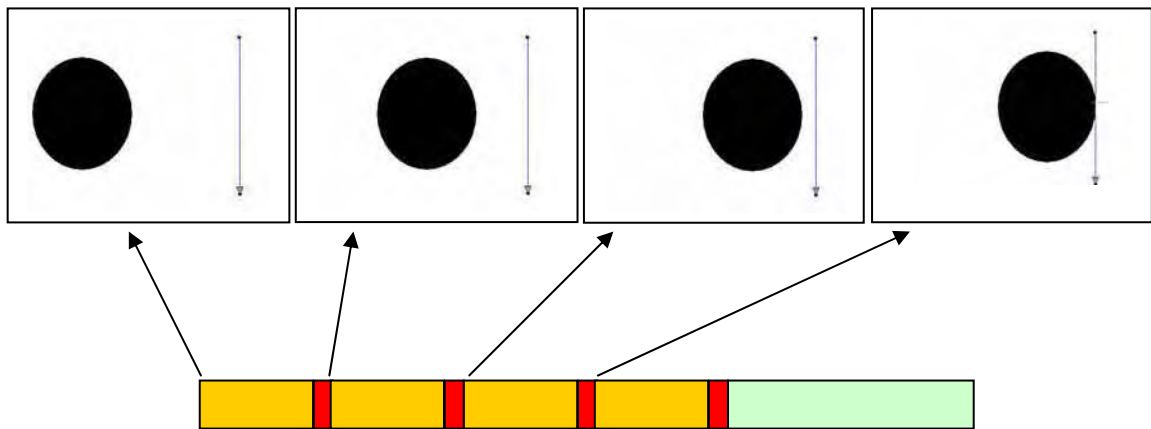


Supposing to detect a black object moving from left to right on a white background and to calculate its area. During acquisition, the object has to be included completely in the inspection area in order to guarantee a correct inspection.



- *Software trigger* mode

The *Software trigger* analyses the acquired image and tracks a light variation along the dashed line. This condition is necessary to complete elaboration.

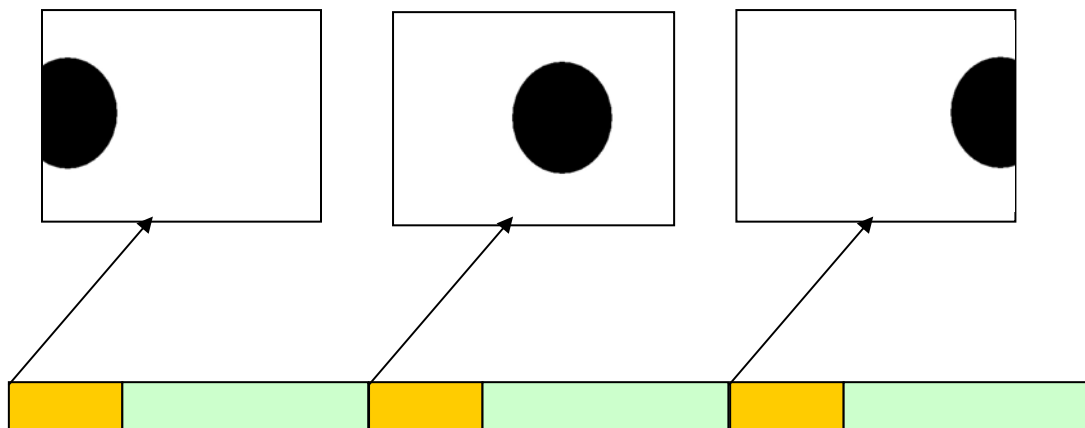


During the first three inspection cycles (referred to the first three photograms), the sensor acquires the image but the elaboration is immediately blocked as the *Software trigger* condition is not respected. At the fourth cycle, the object interrupts the line and consequently the elaboration is completed.

In this manner, the image analysed is the right image to inspect.

2) *Continuous mode*

No elaboration condition has to be respected and the images are all acquired and elaborated.



Obviously not all the images acquired are meaningful. For example, the first and the third would generate an inspection failure even in presence of a valid object.

However the *Continuous mode* is very useful during functioning verification of a recently-configured inspection.



## Parameters

**X:** coordinate that identifies the horizontal position of the first locator end.

**Y:** coordinate that identifies the vertical position of the first locator end.

**Length:** ROI length.

**Thickness:** elaboration area thickness.

**Angle:** expressed in degrees, calculated according to the tool rotation respect to the X axis.

### Variation type:

- "from white to black": only negative edges are considered;
- "from black to white": only positive edges are considered;
- "both": both edges are considered;
- "automatic": tool selects automatically which variation to distinguish.

**Direction:** determines the direction of the elaboration.

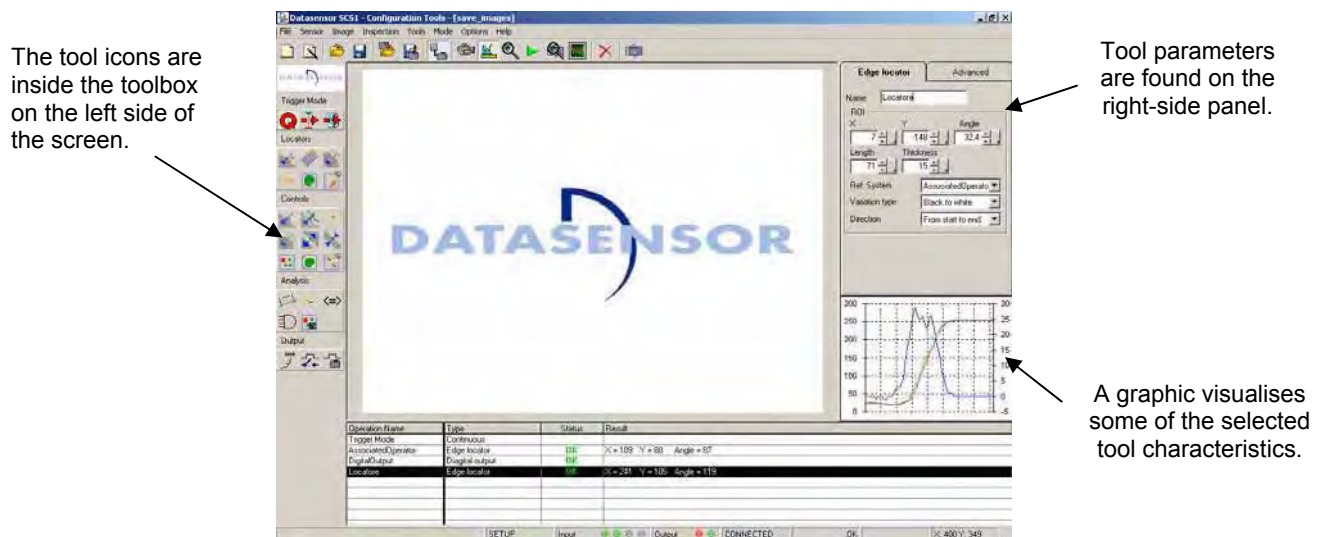
- "from start to end": elaboration starts from the first end and continues towards the arrow;
- "from end to start": elaboration starts from the arrow and continues towards the first end (opposite to the arrow's direction).

Please refer to "*Edge locators*" paragraph for further details.



## 6.6. Tools

**What is a tool?** A tool is a software means used to carry-out a specific operation on the acquired image. The insertion of each tool represents an inspection operating step.



**Positioning:** most tools are graphically positioned using the mouse. The position and dimensions can be changed either graphically or by modifying the parameters.

**Parameters:** each tool has different parameters that regulate position, operations carried-out, project specifications. The parameters have default values that can however be modified according to specific requirements.

More detailed parameter information can be found under the *Edge locator* and *Blob locator* paragraphs.

**Measurement unit:** coordinates and distances are expressed in pixels.

**Results:** all tools supply both a numeric result as well as a status result. The latter indicates the success (OK) or failure (Failure/Error) of the inspection.

Each tool carries-out specific operations and the user can configure the relative specifications (e.g. distance measurement, area surface). If the specifications are not respected the tool status generates **Failure**.

If the measurement is not carried-out, the tool status generates an **Error**. The **Error** status can have different causes. Position the cursor using the mouse on the **Error** text (Figure 20) to visualise the code factor that has generated the error.

<sup>1</sup> Important: in the new graphic interface version, the Apply icon is no longer available and the changes are made automatically when the parameters are modified or pressing Enter.



Operation Name	Type	Status	Result
Trigger	Software trigger	ERROR	
Blob	Blob count	ERROR	
		Trigger error	

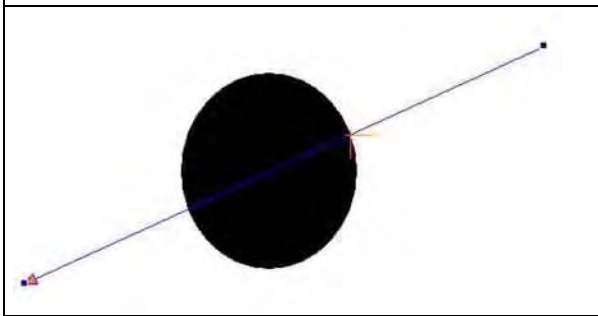
Figure 20

Some possible error causes are listed below:

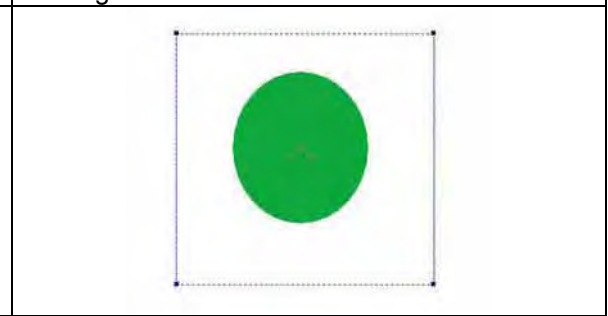
- *Software trigger error*: the *Software trigger* is not able to find the elaboration running condition and consequently all successive tools are blocked.
- *Reference error*: the tool reference system indicates another tool which has generated an error.
- *Operating error*: the tool is unable to carry-out the operation.
- *Operator-generated error*: some tools (e.g. *Test*) process the result of other tools used previously. If a previous tool generated an **Error**, the new tool will generate **Error**.

**Region of interest (ROI)**: indicates the image part that is analysed by a specific tool. The area outside the ROI is ignored during elaboration but remains visualised. The ROI can be linear (e.g. *Edge locator*) or area (e.g. *Blob locator*).

**Linear ROI**: in the *Edge locator*, indicated in the figure, the ROI consists in a blue line that represents the locator.



**Area ROI**: the *Blob locator* shown in the figure is represented by a blue rectangle. The ROI in this case is the entire surface included in the rectangle.

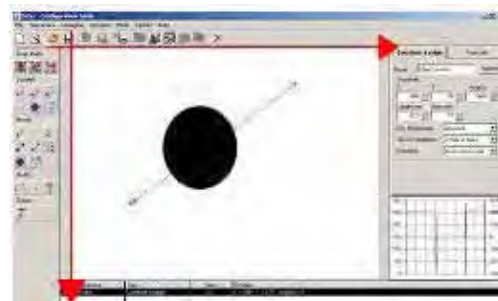
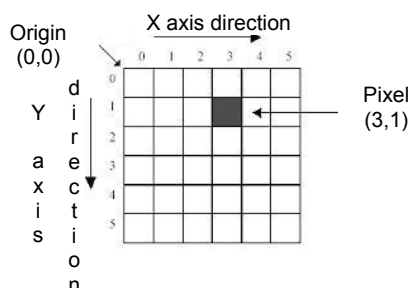


**Reference system**: all tools are referred either to an absolute reference system or to a relative reference system connected to the tool used previously.

*Absolute reference system*: its origin is positioned in the top-left corner of the image panel.

The X and Y coordinates of the *Edge locator* refer to this origin.

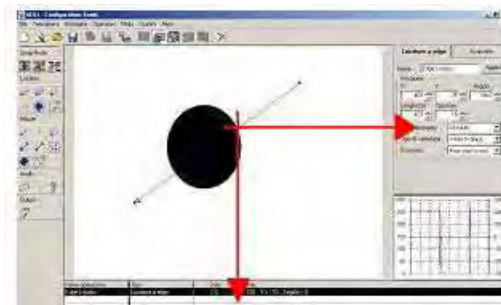
The axis orientation is shown in the figure.





*Relative reference system:* the system uses the result of tools used previously (e.g. position of an edge) to set its origin.

The coordinates are referred to the relative reference system's origin. The axes have the same orientation as the absolute system.



**Graph:** a graph visualising information relative to the image's brightness distribution is associated to some tools based on edge detection (*Software trigger*, *Edge locator*, *Edge*, *Internal caliper* and *External caliper*) and blob analysis (*Blob locator*, *Blob count* and *Contour match*). The graph can be interpreted differently according to the elaboration techniques (see below).

Position the mouse cursor on the graph to visualise a tooltip window containing the relative information.

#### Edge detection graph

Figure 21 and 22 represent respectively an Edge locator and the associated graph.

The graph is divided as follows:

**Red** profile: pixel's brightness distribution along the elaboration line. The tooltip parameter Absolute supplies the brightness numeric value.

**Green** profile: noise-filtered brightness distribution. The tooltip parameter Filtered supplies the filtered brightness numeric value.

**Blue** profile: brightness derivative. The tooltip parameter Derivative supplies the numeric value of the brightness derivative.

*Tooltip:*

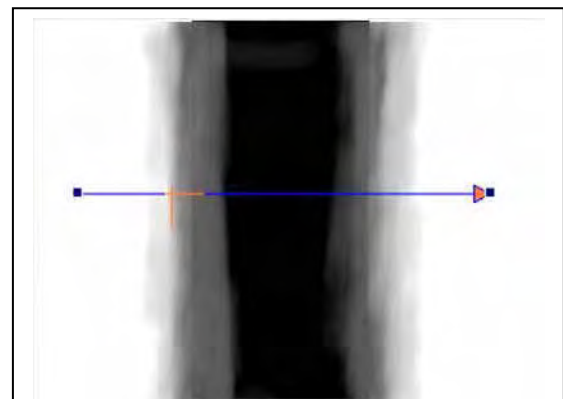
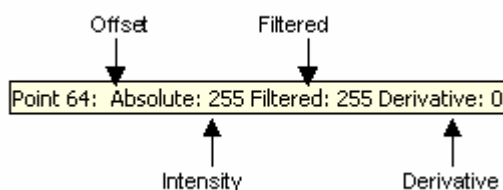


Figure 15: Edge locator

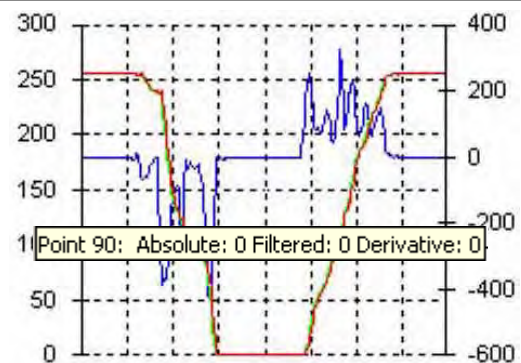


Figure 16: Edge locator graph



**Abscissa axis:** the offset values are indicated along the X axis with respect to the tool first end. The value of the "Point" parameter refers to this offset.

**Example:** in figure 17, the "Point" is equal to 69. This means that the cursor, referred to the Edge locator, is positioned 69 pixels away from the first end. All the other tooltip parameters refer to a value relative to this point.

**Ordinate axis:** has a double scale.

The values indicated on the left side refer to the intensity distribution and filtered intensity. Each value is associated to a grey tone. The value accepted range between 0 (black) and 255 (white). The values indicated on the right refer to the intensity derivative.

#### *Blob analysis graph*

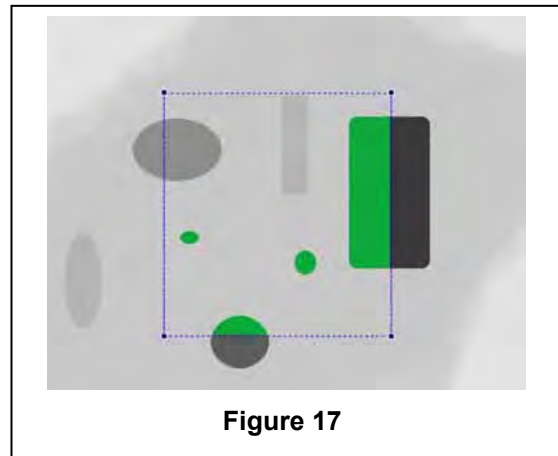
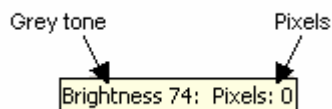
Figure 23 and 24 represent respectively a *Blob count* and the associated graph.

The graph is divided are follows:

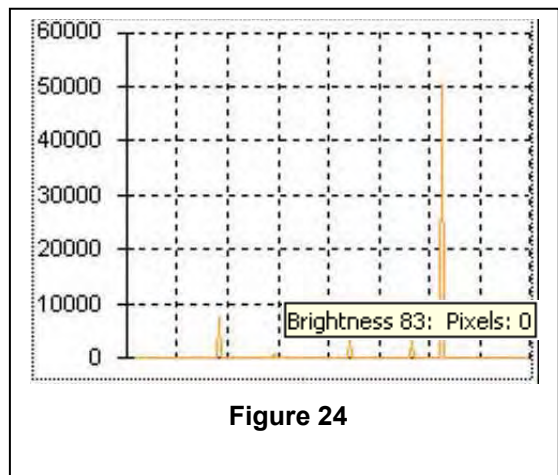
**Abscissa axis:** indicates the entire greyscale (0-255). The tooltip parameter Brightness refers to the grey tone of the mouse cursor's position.

**Ordinate axis:** indicates the number of pixels with a certain grey tone. This number is visualised in the tooltip pixel parameters.

**Tooltip:**



**Figure 17**



**Figure 24**



## 6.7. Locators

Locators determine the position of objects inside the sensor's field of view (FOV). Generally, locators distinguish one point and supply the coordinates to use in the successive phases as references. The locator's result is also highlighted graphically on the window with a cross presenting two long and two short sides, which represents a possible relative reference system for the successive operators. The locators supply two parameters of the relative reference system: the origin (point found by the tool) and the angle respect to the absolute reference system.

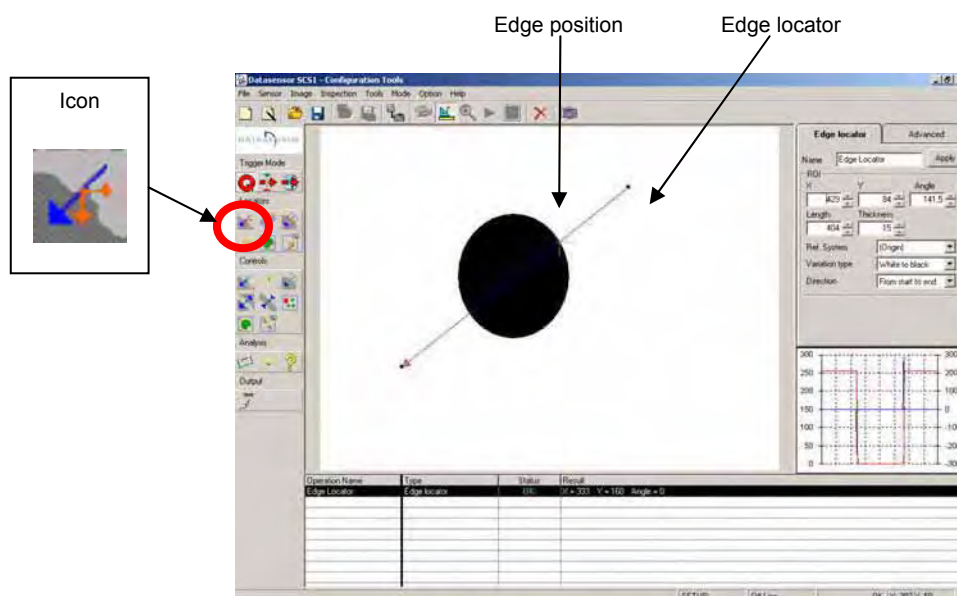
### 6.7.1. Edge locator

**What is an Edge?** A brightness variation between adjacent pixels with a value higher than a given threshold.

**Description:** the *Edge locator* distinguishes a transition point from bright areas to dark areas (negative edge) and from dark areas to bright areas (positive edge) of the image to inspect. The edge point is practically the boundary between two areas characterised by different brightness (e.g. white objects on black background). Once located the edge position, it can be used as a reference for other locators or measurement tools.

**Positioning:** refers to *Software trigger*. The *Edge locator*, differently from the *Software trigger*, is represented by a continuous line.

Once specified the edge type to be located (positive or negative), the tool elaborates the ROI individuated from the line and stops at the first edge that respects the assigned specifications.



#### Results:

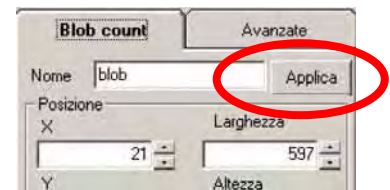
- located edge coordinates ("X", "Y");
- graphic representation of the reference system relative with origin in the edge point;
- relative reference system angle ("Angle", always 0° for the *Edge locator*);
- status.

Operation Name	Type	Status	Result
Edge Locator	Edge locator	OK	X = 333 Y = 160 Angle = 0
Blob	Blob count	OK	Number = 1



## Parameters

**Apply:** to confirm parameter changes the user has to simply click on "Apply".



**Reference system:** absolute or relative.

**X:** coordinate that identifies the horizontal position of the first locator end (opposite to the arrow point) with respect to the selected reference system. Corresponds to the pixel elaboration beginning.

*Accepted values: 0-640*

**Y:** coordinate that identifies the vertical position of the first locator end with respect to the selected reference system.

*Accepted values: 0-480*

**Length:** ROI length.

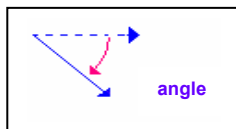
*Accepted values: 0-640*

**Thickness:** elaboration area thickness.

The edge is located along a line (linear ROI) but, to make the result more reliable, the brightness values of the surrounding pixels are taken in account by means of a projection operation.

Please refer to "Projection" (Appendix A, 12.7) for further details.

*Accepted values: 0-640*



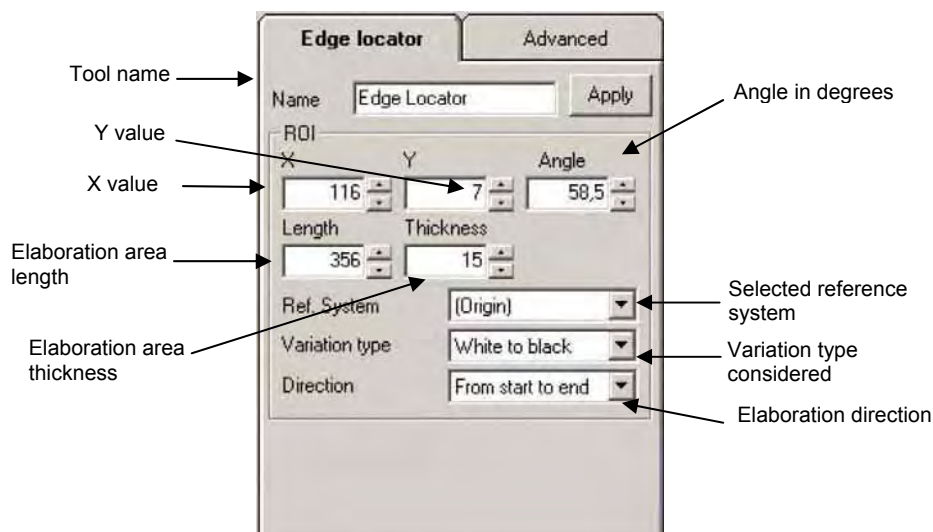
**Angle:** expressed in degrees, calculated according to the tool rotation respect to the X axis.

**Variation type:** two edge types can appear: one due to the transition from a dark to a bright area (positive edge) and one due to the transition from a bright to a dark area (negative edge).

- "from white to black": only negative edges are considered;
- "from black to white": only positive edges area considered;
- "both": both edges are considered;
- "automatic": tool selects automatically which variation to distinguish.

**Direction:** determines the direction of the elaboration:

- "from start to end": elaboration starts from the first end and continues towards the arrow;
- "from end to start": elaboration starts from the arrow and continues towards the first end (opposite to the arrow's direction).





### Advanced parameters

**Search threshold:** refers to the highest value, “maximum peak”; (in case of positive edge) or of the minimum value (negative edge) of the brightness derivative. The threshold that determines if a brightness variation can be considered as an edge or not is calculated in percentage of this value. Variations inferior to this threshold are not considered edges.

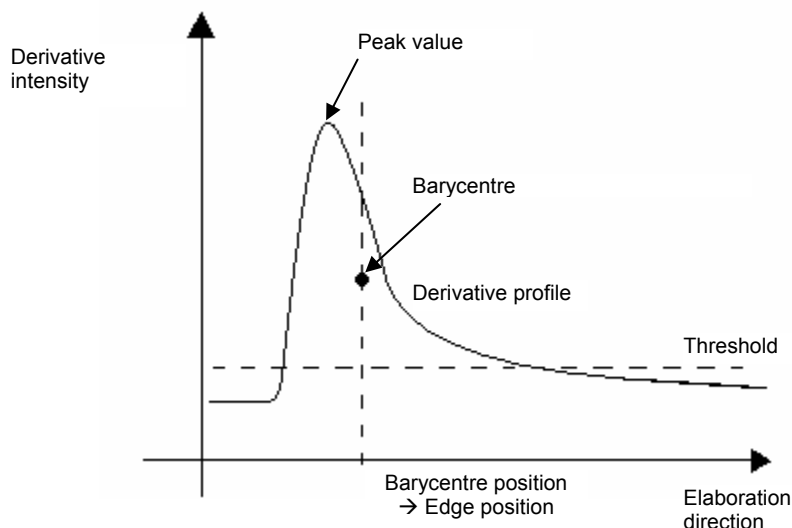
*Accepted values: 0-100. Default value: 60*

**Absolute threshold:** refers to the brightness derivative. The threshold sets a criteria to filter small noise variations during the edge's search. It is expressed in grey levels and establishes the minimum variation value.

*Accepted values: 0-255. Default value: 5*

**Barycentre threshold:** the reference barycentre is referred to the area under the curve individuated by the peak value of the brightness derivative profile. The barycentre position determines the edge position.

The threshold is expressed in percentage of the peak value and determines the area considered to calculate the barycentre. According to the threshold value, a line parallel to the abscissa axis is drawn. The area considered for the barycentre calculation is included between the line and the curve.



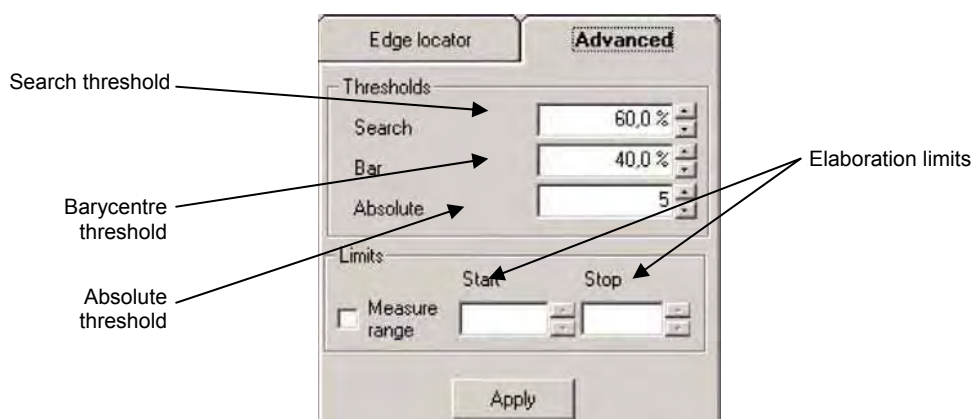
E.g. peak value =100, barycentre threshold=40%. A line parallel to the X axis at a 40 height is drawn. The area included between the line and the curve is used to calculate the barycentre.

**Constraints:** the *Edge locator* is characterised by a specific ROI.

The elaboration can be limited to a certain ROI part:

- Beginning: first elaboration starting point.
- End: first elaboration ending point.

*Note:* the threshold refers to the highest value that remains the entire *Edge locator* area.

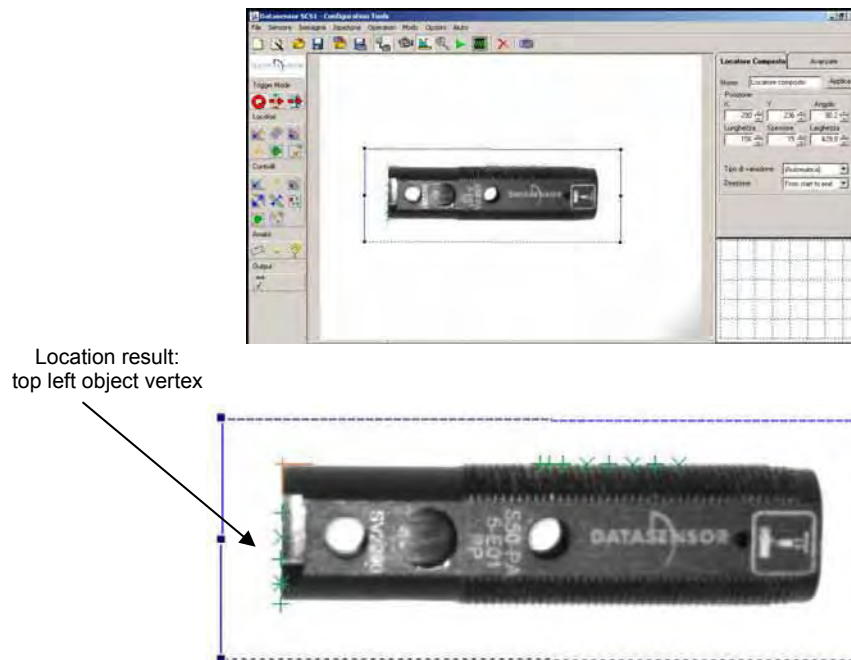




## 6.7.2.

**Composite locator**

The *Composite locator* identifies the position of objects having approximately a rectangular shape and a certain background contrast. The tool also has a rectangular shape and is positioned to include the target-object. For a correct functioning, the locator has to be larger than the target-object (refer to figure 25). The *Composite locator* is composed of different *Edge* and *Inclinometer* locators that determine the top left vertex position of the located object.

**Figure 25****Positioning:**

- click on left mouse button to set the rectangle vertex;
- drag the mouse (keeping button pressed) to set the first rectangle side;
- the rectangle near the first side is automatically drawn moving the mouse. Click again left mouse button to confirm position.

We recommend to position horizontally the object to locate in the reference image.

**Results:**

- top left vertex coordinates of the located object;
- status;
- relative reference system graphic representation with origin in the top left side of the object.

**Parameters**

**X, Y:** top left ROI coordinates.

**Height:** rectangle height delimiting the ROI.

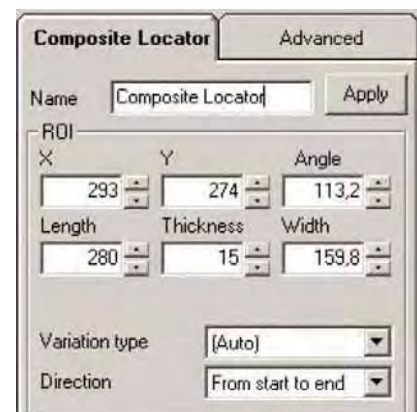
**Thickness:** thickness of each locator.

**Width:** rectangle width delimiting the ROI.

**Variation type:** edge type tracked by the locators that form the *Composite locator*.

- "from white to black": tracks a black object on a white background;
- "from black to white": tracks a black object on a white background;
- "both": considers both;
- "automatic": tool selects automatically which variation to distinguish.

**Direction:** determines the direction of the elaboration (beginning-end or end-beginning).

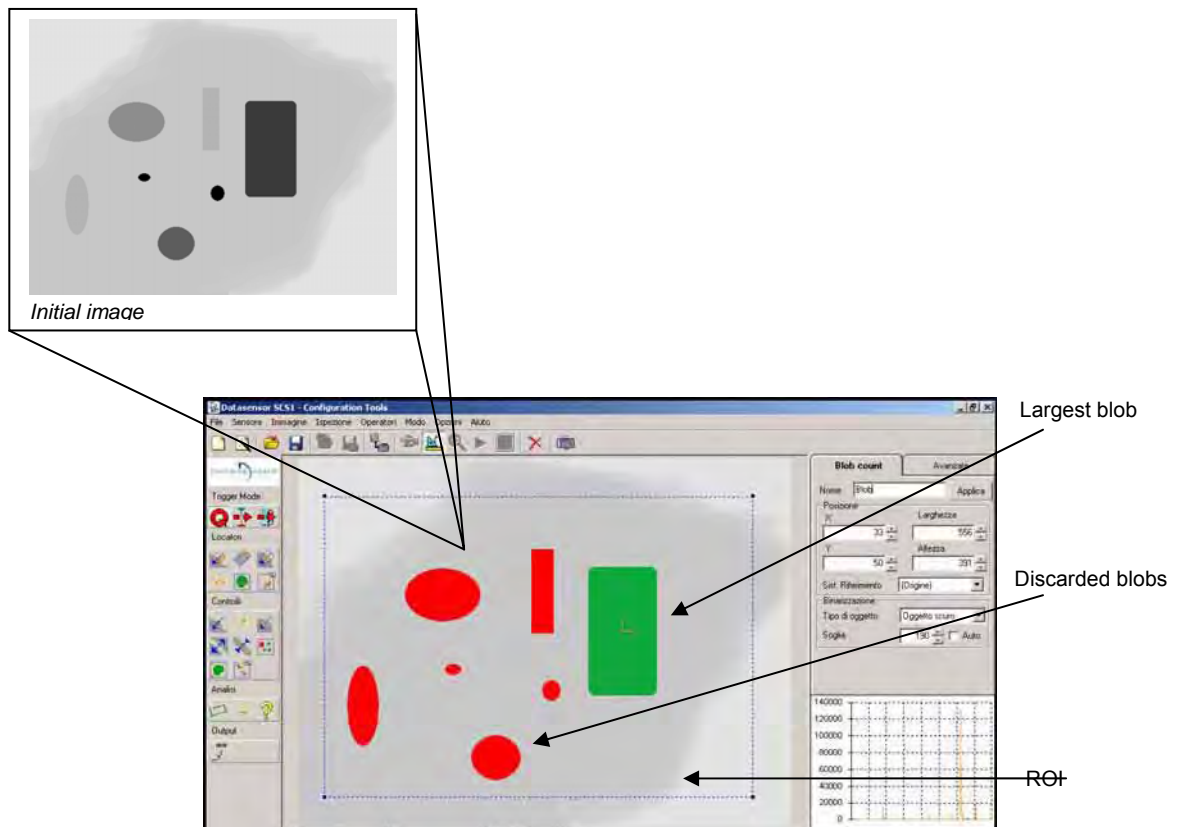




### 6.7.3. **Blob locator**

**What is a Blob?** A group of adjacent same-coloured pixels, white or black.

**Description:** the *Blob locator* tracks inside a specific area the largest blob. *Blob locators* basically carry-out two functions: an initial binarization and a successive blob identification. Binarization allows the user to pass from a grey-tone image to a black and white image. Same-coloured pixel groups are gathered together and form blobs. The locator is represented graphically by a rectangle, that forms the ROI (*Region of Interest*) where the blobs are tracked. Blobs consist in white or black pixels. After the selection of the blob type to isolate, the locator tracks inside the ROI the larger blobs.



The figure indicates how the locator, starting from the top image, locates blobs with dark pixels (selected previously). The largest blob is highlighted in green, while smaller blobs in red are discarded. Established the binarization threshold and pixel type to consider (white or black), the image part that does not respect the criteria remains visualised with no variations.





*Note:* the main characteristics of the selected blob are visualised by clicking on the right mouse button on the coloured area identifying a blob.



#### Positioning:

- click on left mouse button to set the rectangle vertex;
- drag the mouse (keeping button pressed) to set rectangle dimension;
- release button to confirm;
- the position and dimensions can be changed graphically.

#### Results:

- barycentre coordinates of the blob with larger dimensions ("X","Y");
- relative reference system graphic representation with origin in the barycentre of the blob with larger dimensions;
- relative reference system angle ("Angle");
- status.

#### Parameters:

**X, Y:** top left ROI coordinates.

**Height:** rectangle height delimiting the ROI.

**Width:** rectangle width delimiting the ROI.


**Object type:** specifies the blob type to locate:

- black object;
- white object.

**Binarization threshold:** value of the threshold used to distinguish light-coloured pixels from the dark ones during image passage from greyscale in black and white.

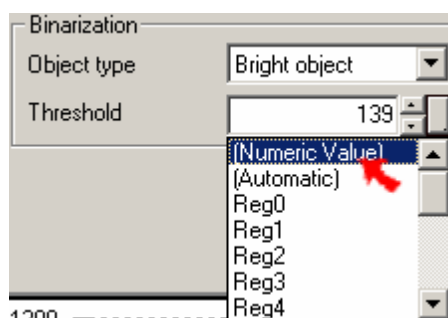
Pixels with a value superior to the threshold result to be white after binarization, those with an inferior value black.

The threshold is calculated automatically when the tools are inserted.

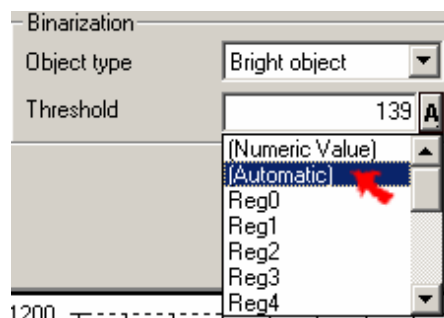
To fix it manually, the user has to click on the icon on the left of the numeric threshold value (  ), then click on "Numeric Value", and the "A" disappears. The value of the "Threshold" can be now changed.

Choosing again "Automatic" the device returns to the automatic mode (the letter "A" has to appear in the box).

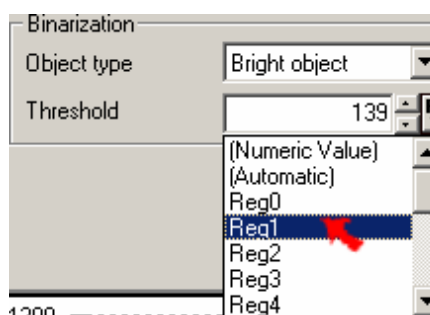




Selezione soglia manuale



Selezione soglia automatica



Associazione soglia/registro

Moreover, as already explained in the “Regist” section, the threshold is a parameter and so it can be associated to a sensor register. The threshold can be thus recalled and modified using the device integrated interface.

### Advanced parameters

**Minimum and maximum areas:** refers to the minimum/maximum number of pixels that can form a blob. Smaller and larger objects area discarded.

The minimum value is used, for example, to filter noise and other non-relevant object characteristics, while the maximum value can be used to filter the background.

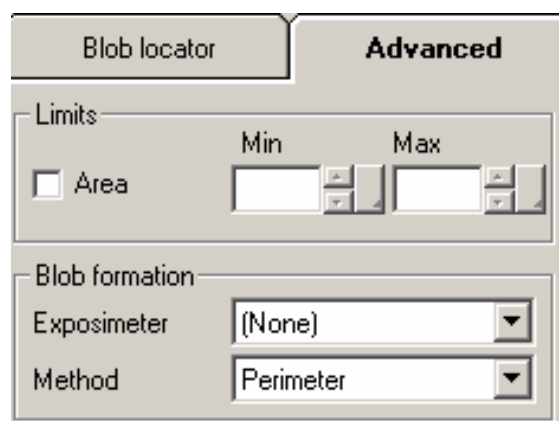
*Note:* a double click on the left mouse button on one of the coloured areas highlighting a blob sets automatically the minimum and maximum areas. The minimum area will be equal to the selected blob area minus 10%. The maximum area will be equal to the selected blob area plus 10%.

**Blob formation:** selects the method used to form blobs.

Two options are possible:

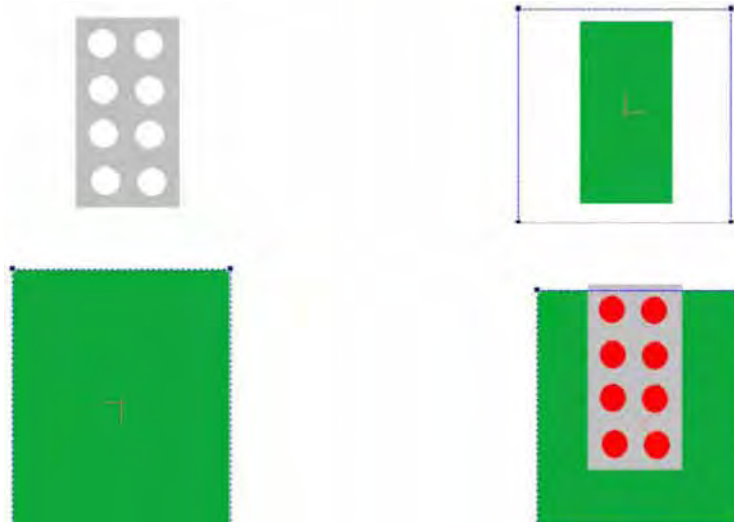
*Perimeter:* tool carries-out binarization inside the tracking area and follows the border of the selected objects (white or black). The borders characterised by closed lines are considered valid and the entire internal area forms the blob.

Internal perimeters are not considered. For example, if the black target-object is covered with holes or white spots, the located blob remains just one and includes the whole area inside the external perimeter. This methods reduces elaboration time, even if some image information is lost.





*Example: grey blister with white pills.*



The initial image is always the one on the top left. The result changes modifying tool position and parameters.

Top right image: in this case a black object has to be tracked. The object is correctly tracked but the pills forming internal perimeters are not considered.

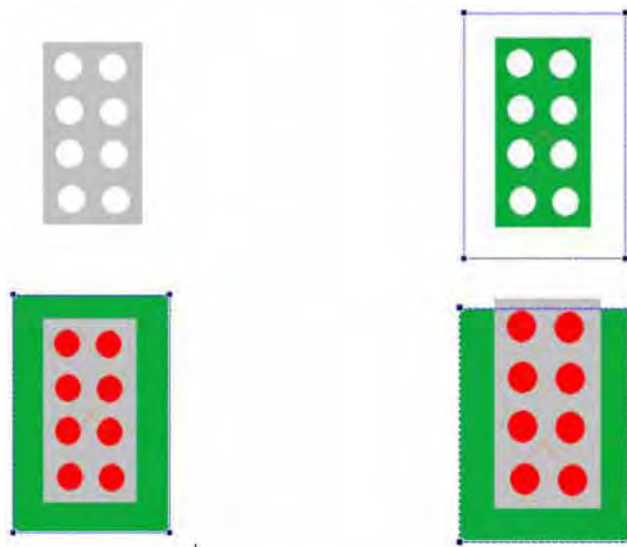
Bottom left image: in this case a white object has to be tracked.

The tool follows the white object's background (including completely the blister) and the blob tracked is formed by the area enclosed by the tool. Blister and pills form internal perimeters and so are not considered.

Bottom right image: a white object has to be tracked. The tool has been drawn to intercepts the top side of the blister. Thus the border that characterise the background does not enclose completely the blister and pills (also white) are not internal perimeters anymore. In this way pills can be tracked as blobs.

**Area:** all the characteristics of the same-coloured pixel groups have to be considered to form a blob. The internal parts are also tracked. This method provides major precision respect to the previous perimeter method, but requires longer elaboration times.

*Example:*





The images are referred to the same situation analysed using the perimeter method. The blobs are well-tracked in all four images.

The method has to be selected according to inspection timing as well as the results to reach.



#### 6.7.4. Pattern match locator

**Description:** *Pattern matching* compares the similarity between a template and a target-object to inspect. This locator tracks the target-object inside a given area and sets its position. The locator is graphically represented by two rectangles, one inside the other that respectively form the research area, ROI, (external one marked in blue) and the template (internal one marked in orange).

The target-object characteristics are saved during Setup with the following criteria: the brightness value (0-255) is stored for each pixel.

The locator tracks the target-object inside the ROI during inspection and sets its position. The similarity threshold and accuracy parameters can be modified according to the different inspection requirements.

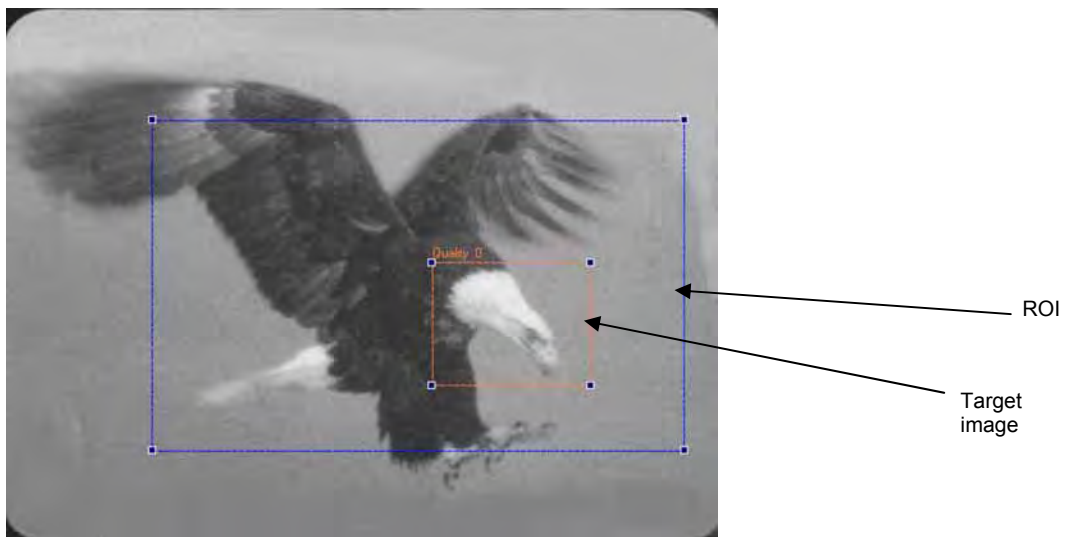
**Positioning:**

- draw a rectangle enclosing the target-object;
- after confirming the position by clicking on the left mouse button, an external rectangle will appear automatically representing the tracking area.

The position and dimension of the two rectangles can be changed, but not the rotation direction.

**Results:**

- rectangle's top left vertex coordinates enclosing the target-object ("X","Y");
- status.



**Parameters:**

The parameter area is divided in ROI and image.

The parameters are always the same and refer to the rectangle of the ROI or image.

**X, Y:** top left vertex coordinates.

**Height:** rectangle height.

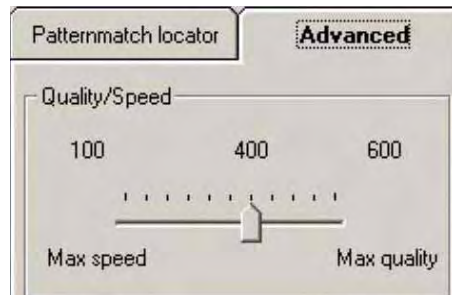
**Width:** rectangle width.

**Reference system:** selects the reference system (available only for the ROI).



### Advanced parameters

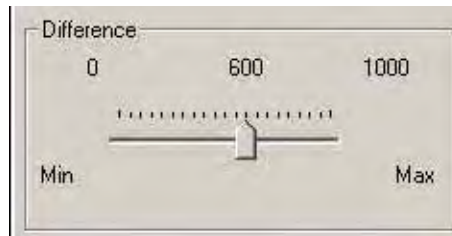
**Quality / Speed:** inspection precision and timing can be changed by using the mouse cursor. Higher the quality, the higher the inspection accuracy, but the longer the inspection time. Higher the speed and shorter the elaboration time, but the lower the inspection precision.



**Difference threshold:** parameter that takes in account the difference between the template and object to inspect. The lower this value, the more the two objects have to resemble each other to guarantee a positive result.

Difference = 0 → the two objects have to be identical;

Difference = 1000 → differences are tolerated.



#### 6.7.5. *Inclinometer locator*

**Description:** the *Inclinometer locator* determines the angle of an object or part and supplies a reference to the successive inspection phases.

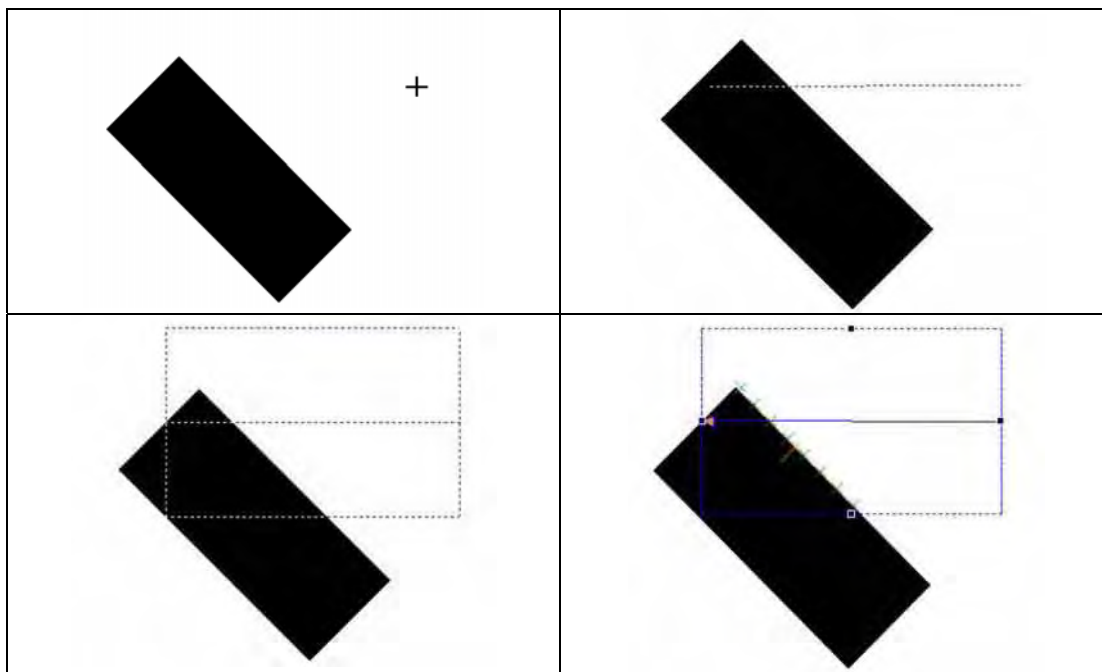
The locator is positioned on the window drawing a line (which represents a reference *Edge locator*), around which a rectangle is designed (dimensions can be changed using the mouse). The rectangle represents the elaborated area: around the reference *Edge Locator*, a certain number of parallel *Edge locators* is positioned automatically (but not graphically visualised). The number of *Edge locators* depends on the rectangle width and can be changed either graphically or manually modifying the "Points" parameters.

The points tracked by each *Edge locator* are interpolated to form a segment characterised by a certain angle, which is the result of *Inclinometer locator*.

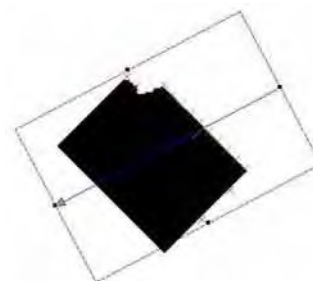


**Positioning:**

- click on left mouse button on the point that will become the first central *Edge Locator* point;
- drag the mouse (keeping the button pressed) to set the *Edge locator* length;
- release the button and use the mouse to change rectangle dimensions;
- click again on the left mouse button to confirm.



The edges tracked are graphically visualised by a green cross. Not all locators may track an edge and moreover some edges may not form a segment. In the latter, if an edge is outside the segment tracked by the majority, it will be visualised by a red cross.

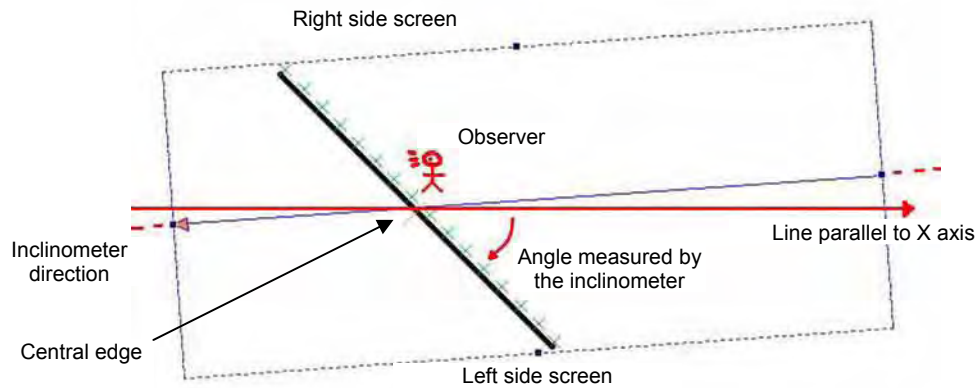
**Results:**

- angle between the line parallel to X axis of the absolute reference system passing through the edge tracked by the central *Edge locator* and the line identified by the segment interpolated by the tracked edges ("Angle"). Please refer to example given below;
- coordinates of the edge tracked by the central *Edge locator*;
- reference system graphic representation with origin in the central edge point and angle, respect to the absolute angle tracked by the inclinometer("Angle");
- status.



**Example:**

The angle is determined as follows. An hypothetical observer is positioned in correspondence to the edge tracked by the central *Edge locator* in the direction indicated by the inclinometer arrow. The line identified by the *Edge locator* divides the window in two parts. One on the left side of the observer and the other on the right side. The angle between the line parallel to the X axis passing through the observer's position (central edge) and the segment part obtained interpolating the edges on the left side has to be calculated in order to reach the inclinometer result.

**Figura 26**

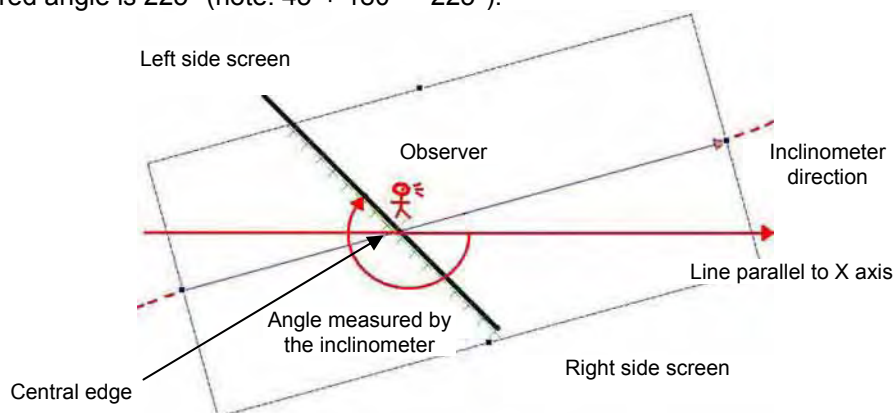
The figure 26 represents:

- *Inclinometer locator* (blue);
- object to track (black line);
- line parallel to the X axis passing through central edge (red);
- observer (red);
- angle tracked (red).

The red elements are useful to understand the inspection but are not visualised during the application.

In this example, the window part on the left side of the observer is the bottom one. The measured angle is 45°.

To complete the example, figure 27 represents the same object and the *Inclinometer* is rotated by 180°. The part of the window that was on the left side of the observer is now on his right side and vice versa. Starting from the line parallel to the X axis, the locator has to arrive to the segment part on the top part of the window before ending the angle measurement. The measured angle is 225° (note:  $45^\circ + 180^\circ = 225^\circ$ ).

**Figura 27**



**Parameters**

The **X, Y, angle and reference system** parameters refer to the central *Edge locator*.

**Length, thickness, variation type, direction, thresholds and constraints** are relative to all the locators that form the inclinometer.

**X:** coordinate that identifies the horizontal position of the first locator point.

**Y:** coordinate that identifies the vertical position of the first locator point.

**Length:** ROI length.

**Thickness:** elaboration area thickness.

**Angle:** expressed in degrees, calculated according to the tool rotation respect to the X axis.

**Variation type:**

- "from white to black": only negative edges are considered;
- "from black to white": only positive edges are considered;
- "both": both edges are considered;
- "automatic": tool selects automatically which variation to distinguish.

**Direction:** determines the direction of the elaboration:

- "from start to end": elaboration starts from the first end and continues towards the arrow;
- "from end to start": elaboration starts from the arrow and continues towards the first end (opposite to the arrow's direction).

**Points:** this parameter specifies the number of edge points the tool tracks and will be used to obtain the interpolated segment. Consequently the parameter establishes the number of *Edge locators* used. If the number is even, the reference *Edge locator* is not considered, while it is considered if the result is an odd number. Half of *Edge locators* are positioned on the left side of the reference *Edge locator* located and the other half on the right side. The distance between the locators is equal to half of the thickness of each.



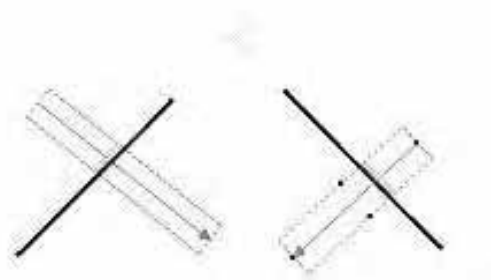
### 6.7.6. Intersection locator

**Description:** calculates the intersection point between two lines.

The locator has two operators that supply the data necessary to calculate the intersection. The operators can be selected amongst the tools used in the previous operating steps.

The *Intersection locator* calculates the angle between the two lines that pass through the points tracked by the operators, having an angle equal to the angle specified by the operators.

*Example:* intersection between the lines tracked by two *Inclinometer* locators.



Operation Name	Type	Status	Result
Inclinometer1	Inclinometer locator	OK	X = 132 Y = 370 Angle = -45
Inclinometer2	Inclinometer locator	OK	X = 420 Y = 312 Angle = 45
Intersection	Intersection locator	OK	X = 305 Y = 197 Angle = -45
Test	Test	OK	

#### Positioning:

- none, the user just has to assign the name of the two reference operators.

#### Results:

- coordinates of intersection point ("X", "Y");
- reference system graphic representation with origin in the intersection point;
- relative reference system angle equal to the first operator angle ("Angle").

#### Parameters

Only the parameters identifying the two reference locators can be specified.

The dialog box is titled "Intersection locator". It contains a "Name" field with the value "Intersection" and an "Apply" button. Below this is a section labeled "Operatori" containing two dropdown menus: "Operator 1" set to "Edge Locator" and "Operator 2" set to "Composite Locator".



## 6.8. Controls

Control tools are used to determine the inspection specifications.

The control tools supply a numeric result as well as a status, necessary for the successive operating phases.

### 6.8.1. **Edge**

**Description:** similar to the *Edge locator*, as it tracks an edge position. Refer to *Edge locator* for further details.

The result obtained however can not be used as a relative reference system for the successive phases (differently from *Edge locator* result).

#### Parameters

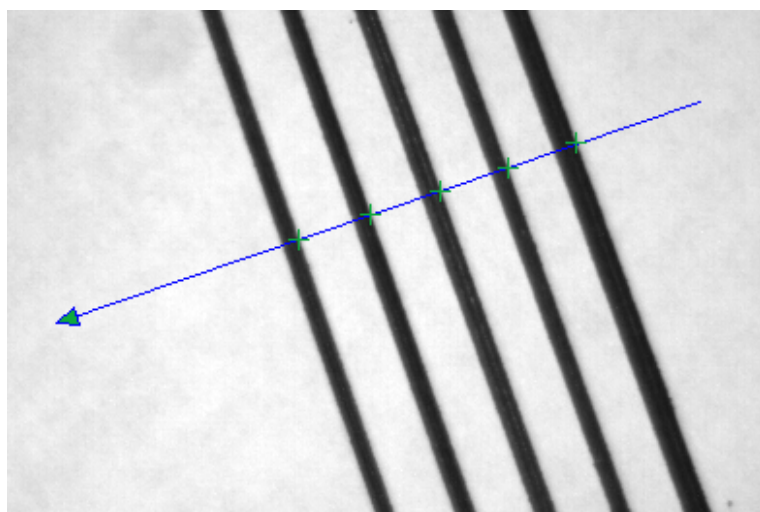
Refer to *Edge locator*.

### 6.8.2. **Edge count**

**Description:** similar to the *Edge locator*, but this tool is able to detect all edges present in the control direction (refer to *Edge locator* at page **Errore. Il segnalibro non è definito.** for more detailed information).

The result obtained however can not be used as a relative reference system for the successive phases (differently from *Edge locator* result).

Specified the edge type to track (positive or negative or both), the tool elaborated the ROI detected by the line and tracks all edges respecting the specifications.



#### Results:

- number of edges tracked in the control direction.

Nome operazione	Tipo	Stato	Risultato
Trigger Mode	Continuo		
Oper00008	Edge count	OK	Number = 5

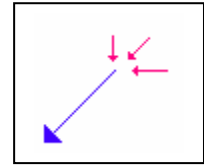


## Parameters

**Reference system:** absolute or relative.

**X:** coordinate that identifies the horizontal position of the first locator end (opposite to the arrow point) with respect to the selected reference system. Corresponds to the pixel elaboration beginning.

*Accepted values: 0-640*



**Y:** coordinate that identifies the vertical position of the first locator end with respect to the selected reference system.

*Accepted values: 0-480*

**Length:** ROI length.

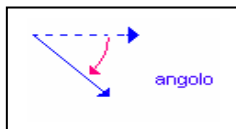
*Accepted values: 0-640*

**Thickness:** elaboration area thickness.

The edge is located along a line (linear ROI) but, to make the result more reliable, the brightness values of the surrounding pixels are taken in account by means of a projection operation.

Please refer to "Projection" (Appendix A.1.6) for further details.

*Accepted values: 0-640*



**Angle:** expressed in degrees, calculated according to the tool rotation respect to the X axis.

**Variation type:** two edge types can appear: one due to the transition from a dark to a bright area (positive edge) and one due to the transition from a bright to a dark area (negative edge).

- "from white to black": only negative edges are considered;
- "from black to white": only positive edges area considered;
- "both": both edges are considered;
- automatic": tool selects automatically which variation to distinguish.

**Direction:** determines the direction of the elaboration:

- "from start to end": elaboration starts from the first end and continues towards the arrow
- "from end to start": elaboration starts from the arrow and continues towards the first end (opposite to the arrow's direction).

**Count:** sets the control on the number of edges that have to be counted inside the range delimited by the minimum and maximum values.



The screenshot shows the 'Edge count' software interface with the following parameters and labels:

- Tool name:** Name (Oper00002)
- ROI:** X (251), Y (87), Angle (67.2)
- Elaboration area length:** Length (222)
- Elaboration area thickness:** Thickness (15)
- Ref. system:** (Origin)
- Variation type considered:** White to black
- Elaboration direction:** From start to end
- Filter on edge count:** Count (Edge, Min 0, Max 0)

### Advanced parameters

**Search threshold:** refers to the highest value, "maximum peak"; (in case of positive edge) or of the minimum value (negative edge) of the brightness derivative. The threshold that determines if a brightness variation can be considered as an edge or not is calculated in percentage of this value. Variations inferior to this threshold are not considered edges.

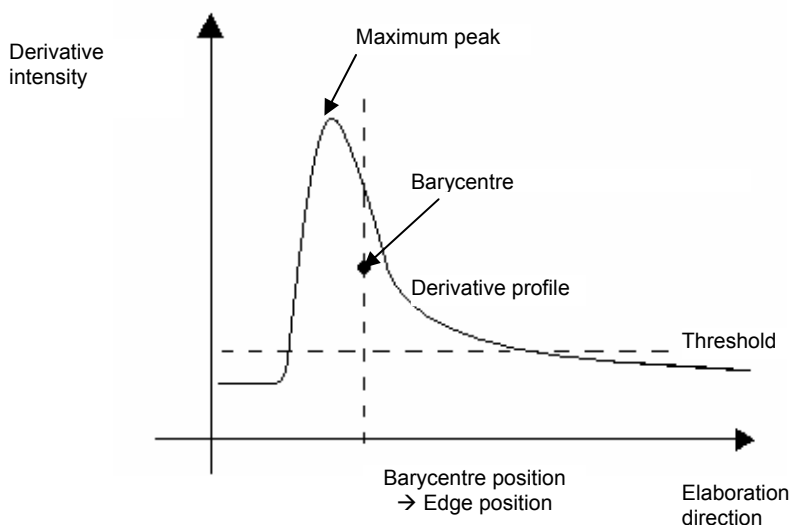
*Accepted values: 0-100. Default value: 60*

**Absolute threshold:** refers to the brightness derivative. The threshold sets a criteria to filter small noise variations during the edge's search. It is expressed in grey levels and establishes the minimum variation value.

*Accepted values: 0-255. Default value: 5*

**Barycentre threshold:** the reference barycentre is referred to the area under the curve individuated by the peak value of the brightness derivative profile. The barycentre position determines the edge position.

The threshold is expressed in percentage of the peak value and determines the area considered to calculate the barycentre. According to the threshold value, a line parallel to the abscissa axis is drawn. The area considered for the barycentre calculation is included between the line and the curve.



E.g. peak value =100, barycentre threshold=40%. A line parallel to the X axis at a 40 height is drawn. The area included between the line and the curve is used to calculate the barycentre.

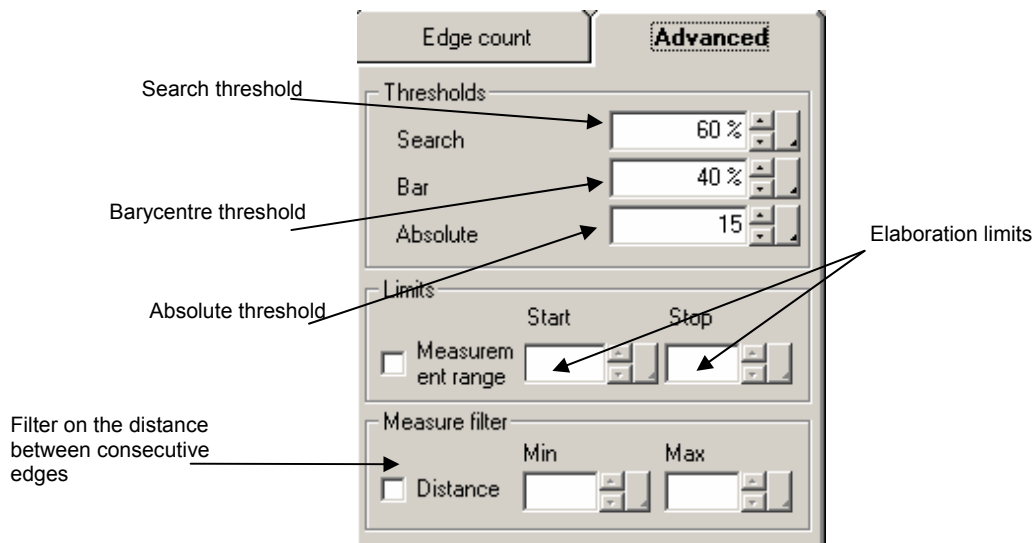


**Constraints:** the *Edge locator* is characterised by a specific ROI.  
The elaboration can be limited to a certain ROI part:

- Beginning: first elaboration starting point.
- End: first elaboration ending point.

*Note:* the threshold refers to the highest value that remains the entire *Edge locator* area.

**Distance:** sets the measurement filter (control) on the distance value between the edges. The tool status depends (if selected) also by the distance between the different edges tracked by the control. In particular, if the distance between consecutive edges is inside the minimum and maximum values set in this filter, the control will then present an OK status, otherwise a FAILURE status.



### 6.8.3. Intersection

**Description:** similar to the *Intersection locator*, as it tracks the intersection determined by the result of two previously used tools. Refer to *Intersection locator* for further details.  
The result obtained however can not be used as a relative reference system for the successive phases (differently from *Intersection locator* result).

#### Parameters

Refer to *Intersection locator*.

### 6.8.4. Inclinometer

**Description:** similar to the *Inclinometer locator*, as it evaluates the object angle respect to the X axis. Refer to *Inclinometer locator* for further details.  
The result obtained however can not be used as a relative reference system for the successive phases (differently from *Inclinometer locator* result).

#### Parameters

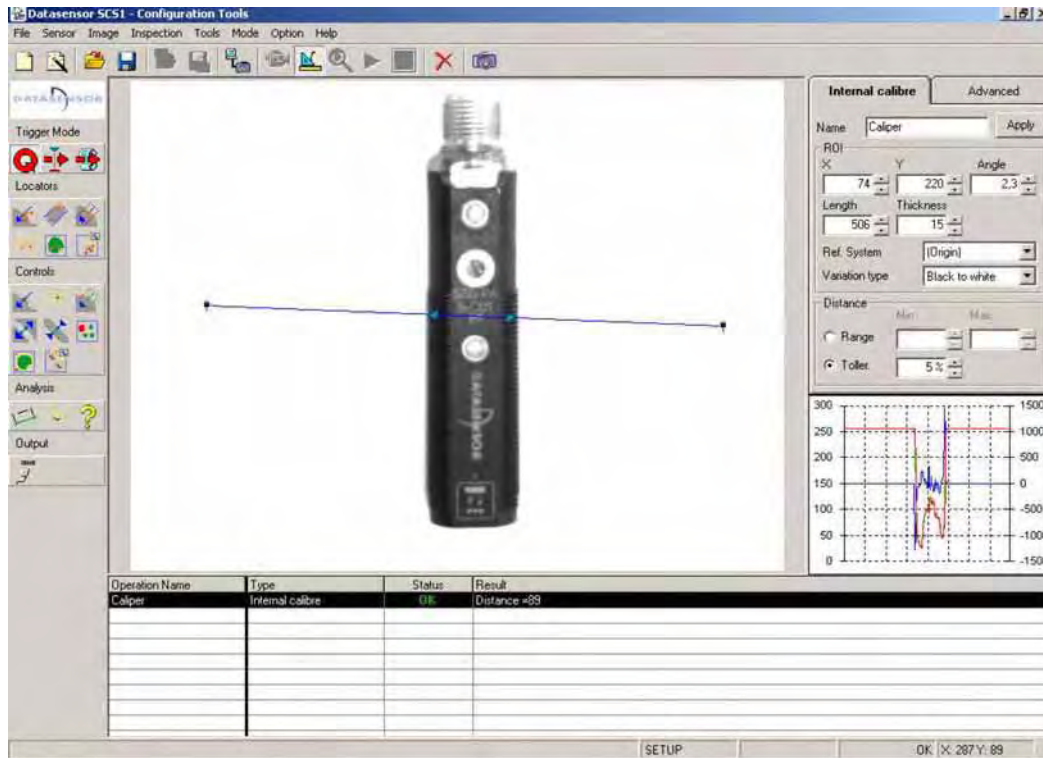
Refer to *Inclinometer locator*.



### 6.8.5. Internal caliper

**Description:** the internal caliper measures distances between two edges.

The tool is represented graphically by a line and carries-out measurements with the following criteria: starting from the middle point of the line, it tracks an edge in both directions (towards end). The elaboration result is the distance between the two edges.



*Note:* ensure that the caliper's medium point is included between the edges of the object to measure.

#### Positioning:

- click on the left mouse button to position the first segment end;
- drag to mouse to set tool dimension;
- release left button to end operation.

#### Results:

- distance;
- status.

#### Parameters

**X:** coordinate that identifies the horizontal position of the first locator point.

**Y:** coordinate that identifies the vertical position of the first locator point.

**Length:** ROI length.

**Thickness:** elaboration area thickness.

**Angle:** expressed in degrees, calculated according to the tool rotation respect to the X axis.

#### Variation type:

- "from white to black": only negative edges are considered;
- "from black to white": only positive edges are considered;
- "both": both edges are considered;
- "automatic": tool selects automatically which variation to distinguish.



**Direction:** determines the direction of the elaboration:

- "from start to end": elaboration starts from the first end and continues towards the arrow;
- "from end to start": elaboration starts from the arrow and continues towards the first end (opposite to the arrow's direction).

**Minimum / maximum distance:** minimum/maximum caliper measurement value accepted. The range of accepted measurement is set by specifying these values and according with them the caliper returns **OK/failure** status.

**Tolerance:** parameter that can be used as an alternative to the distance constraints.

The tool is positioned graphically on the image during Setup and measures the distance between detected edges.

The tolerance is a percentage value that determines, on the calculated distance basis, the range of values accepted during inspection.

*Example:*

Position the *Internal caliper* on the reference image during Setup.

The tool carries-out the elaboration and calculates a distance. In our example the distance is 100 pixel and the tolerance is 10%. During inspection, the range of accepted distance values is:

Value range = [(Setup distance – 10% Setup distance), (Setup distance + 10% Setup distance)]

Value range = (100 – 10), (100 + 10) = [90, 110]

#### 6.8.6. External caliper

**Description:** similar to the *Internal caliper*, but with one difference. Edge tracking begins from the end and continues towards the caliper centre.

**Positioning:** refer to *Internal caliper*.

**Results:**

- distance;
- status.


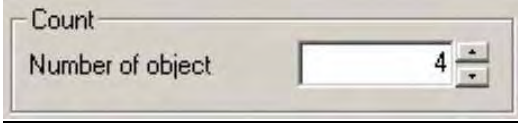
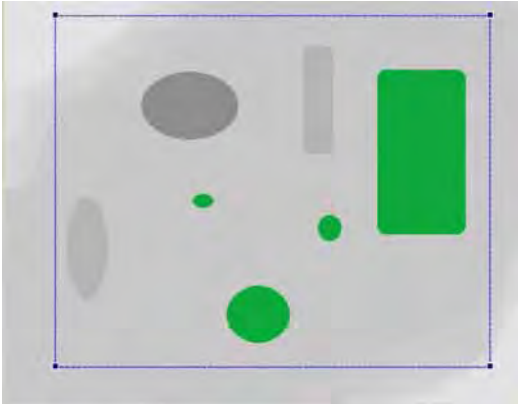
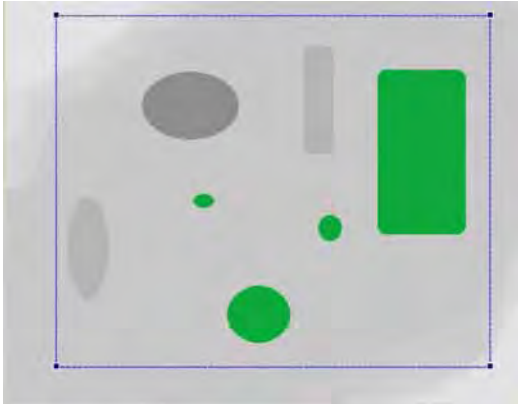
**Parameters**

Refer to *Internal caliper*.



### 6.8.7. **Blob count**

**Description:** tool used for blob counting inside a specific area. The tool is represented graphically by a rectangle that delimits the ROI where the blobs are tracked. Setting the number of expected blobs (parameter “Number of object”) the **OK/failure** status is therefore obtained.

<p>Specification:</p> 																	
<p>Image:</p> 																	
<p>Results:</p> <table border="1"> <thead> <tr> <th>Operation Name</th> <th>Type</th> <th>Status</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>Blob</td> <td>Blob count</td> <td>FAIL</td> <td>Number = 4</td> </tr> </tbody> </table>	Operation Name	Type	Status	Result	Blob	Blob count	FAIL	Number = 4	<table border="1"> <thead> <tr> <th>Operation Name</th> <th>Type</th> <th>Status</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>Blob</td> <td>Blob count</td> <td>OK</td> <td>Number = 4</td> </tr> </tbody> </table>	Operation Name	Type	Status	Result	Blob	Blob count	OK	Number = 4
Operation Name	Type	Status	Result														
Blob	Blob count	FAIL	Number = 4														
Operation Name	Type	Status	Result														
Blob	Blob count	OK	Number = 4														

#### Results:

- number of blobs found;
- status.

#### Parameters

**X, Y:** coordinates of the top left rectangle vertex that delimits the ROI.

**Height:** rectangle height delimiting the ROI.

**Width:** rectangle width delimiting the ROI.


**Object type:** specifies the blob type to track:

- black object;
- white object.

**Binarization threshold:** value of the threshold used to distinguish light-coloured pixels from the dark ones during image passage from greyscale in black and white.

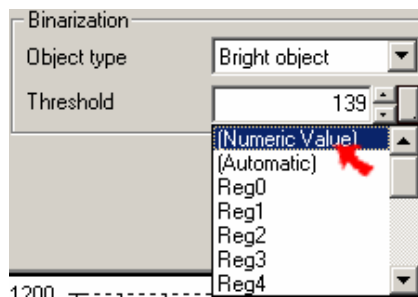
Pixels with a value superior to the threshold result to be white after binarization, those with an inferior value black.

The threshold is calculated automatically when the tools are inserted.

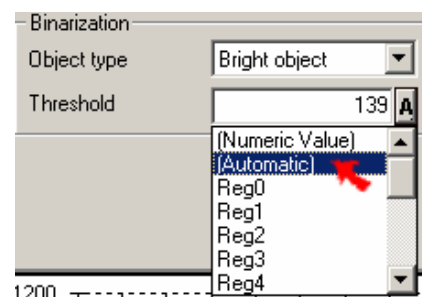
To fix it manually, the user has to click on the icon on the left of the numeric threshold value (  ), then click on “Numeric Value”, and the “A” disappears. The value of the “Threshold” can be now changed.



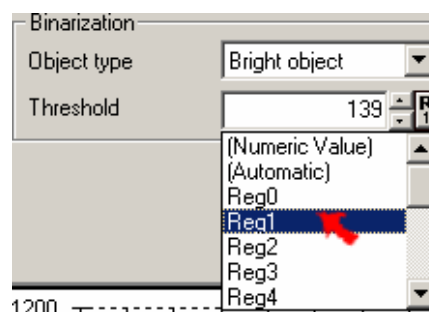
Choosing again



Manual threshold selection



Automatic threshold selection



Threshold/register association

Moreover, as already explained in the “Regist” section, the threshold is a parameter and so it can be associated to a sensor register. The threshold can be thus recalled and modified using the device integrated interface.

**Object number:** specifies the number of blobs to track inside the ROI. If the blob number is different, the status is a **Failure**. Default parameter value is 1.

#### Advanced parameters

**Minimum and maximum areas:** refer to the pixel minimum/maximum number constraint that form a blob. Smaller and larger objects are discarded.

The minimum value is used, for example, to filter noise and other non-relevant object characteristics, while the maximum value can be used to filter the background.

**Perimeter:** minimum and maximum accepted perimeters. The parameter is calculated as follows:

Transitions along X axis + transitions along Y axis + (diagonal transition  $\times \sqrt{2}$ )

The perimeter supplies additional information on the target-object. Objects with the same area may have different parameters.

**Area / perimeter:** minimum and maximum value accepted in the area/perimeter ratio.

**Grey level:** minimum and maximum value of the medium grey level of the blob.



Blobs are tracked on binary images and consequently tone information is lost. Everything is either black or white. This information is recovered, calculating the medium grey blob level on the original greyscale image.

Maximum and minimum grey level values can be specified in order to distinguish objects with the same shape but with a different colour.

*Example:* a green and a blue pills have the same shape and dimensions but have two different grey levels. Binarization may convert all blobs in black.

The blobs can be distinguished by setting the maximum and minimum medium grey levels.

*Accepted values:* 0-255.

**Bounding box width/height (BB):** this is a rectangle with a top left vertex has coordinates that coincide with the minimum X and minimum Y, and a bottom right vertex has coordinates that coincide with the maximum X and maximum Y.

It is practically a rectangle with the smallest area possible able to contain the entire blob.

A control on the object position is made by setting the acceptance constraints on these coordinates.

*Accepted values:* 0-255.



### 6.8.8. Contour match

**Description:** compares a template with a target-object according to the contour characteristics. The *Contour match* is represented graphically by a rectangle that identifies a ROI where to track the target-object. The target-object is tracked through a *Blob analysis* aimed at finding the largest inside the ROI (similar to *Blob locator*). The contour characteristics of the blob tracked are saved and used during inspection to compare the template and target-object. The contour found during Setup is highlighted in yellow. The inspection tolerance can be increased or reduced setting the parameter “Difference”.

**Positioning:** refer to *Blob locator*.

**Results:**

- comparison reliability (“Difference”);
- status.

**Parameters**

**X, Y:** coordinates of the top left rectangle vertex that delimits the ROI.

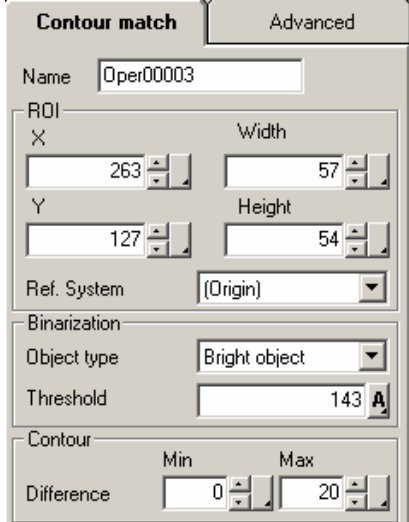
**Height:** rectangle height delimiting the ROI.


**Width:** rectangle width delimiting the ROI.

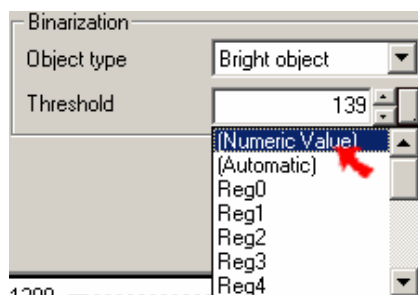
**Object type:** specifies the blob type to track:

- black object;
- white object.

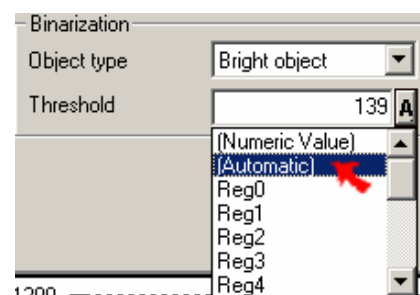
**Binarization threshold:** value of the threshold used to distinguish light-coloured pixels from the dark ones during image passage from greyscale in black and white. Pixels with a value superior to the threshold result to be white after binarization, those with an inferior value black. The threshold is calculated automatically when the tools are inserted.



To fix it manually, the user has to click on the icon on the left of the numeric threshold value (  ), then click on “Numeric Value”, and the “A” disappears. The value of the “Threshold” can be now changed. Choosing again

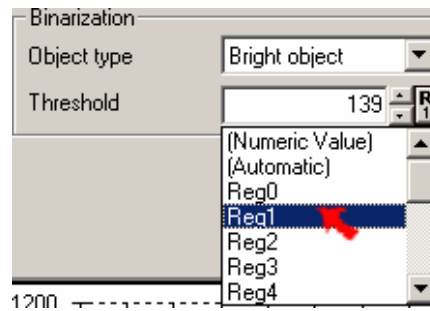


Manual threshold selection



Automatic threshold selection





**Register/threshold association**

Moreover, as already explained in the "Regist" section, the threshold is a parameter and so it can be associated to a sensor register. The threshold can be thus recalled and modified using the device integrated interface.

**Difference threshold:** parameter that takes in account the difference between the template and target-object. The lower this value, the more the two objects have to resemble each other to guarantee a positive result.

Difference = 0 → the two objects have to be identical;

Difference = 1000 → differences are tolerated.

*Note:* if the difference constraints are both zero, the result is always **OK**.



### 6.8.9. Pattern match count

**Description:** counts objects inside a specific area. Similar to the *Pattern match locator*, the *Pattern match count* is formed by two rectangles, an orange one that encloses the template and an external blue one that forms the ROI. During elaboration all objects similar to the template are counted.

**Results:**

- number of objects found;
- status.

**Parameters**

**Counting:** specifies the number of objects to track in the ROI.

If the blob number is different, the status is a **Failure**.

**X, Y:** coordinates of the top left rectangle vertex that delimits the ROI.

**Height:** rectangle height delimiting the ROI.

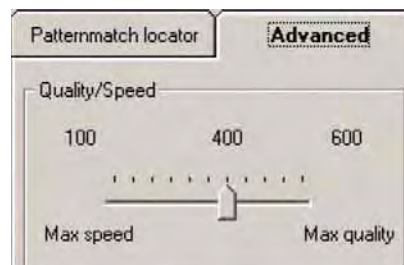
**Width:** rectangle width delimiting the ROI.

**Reference system:** selects the reference system.

**Advanced parameters**

**Quality / speed:** inspection precision and timing can be changed by using the mouse cursor.

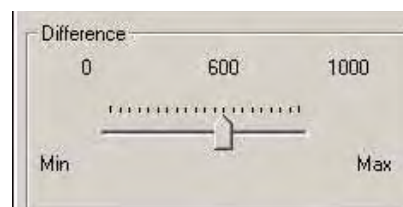
Higher the quality, the higher the inspection precision, but the longer the inspection time. Higher the speed and shorter the elaboration time, but the lower the inspection precision.



**Difference threshold:** parameter that takes in account the difference between the template and target-object to inspect. The lower this value, the more the two objects have to resemble each other to guarantee a positive result.

Difference = 0 → the two objects have to be identical;

Difference = 1000 → differences are tolerated.



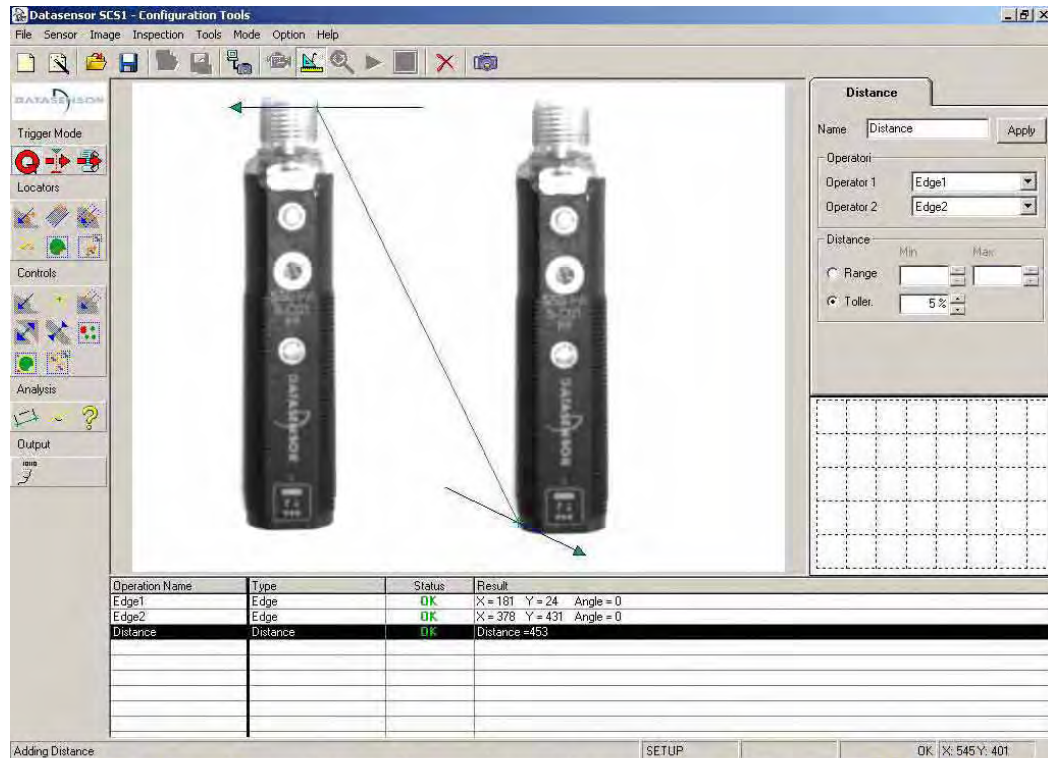


## 6.9. Analysis

Tools used to set further inspection specifications.

### 6.9.1. Distance

**Description:** calculates the distance between two points, using two tools as reference. The reference tools must have as result the coordinates of a point. All locators, some controls such as *Edge*, *Intersection* and *Inclinometer*, can be used as operators.



#### Positioning:

- none, the user just has to assign the name of the two reference tools.

#### Results:

- distance;
- status.

#### Parameters:

**Operator1:** first reference tool;

**Operator2:** second reference tool.

*Note:* one operator can be the origin of the absolute reference system. In this manner position controls can be carried-out using for example a distance locator and tool, and setting accordingly the minimum and maximum distance parameters.

**Minimum / maximum distance:** minimum/maximum value accepted. If the distance is bigger the operating status is **Failure**.

**Tolerance:** parameter that can be used as an alternative to the distance constraints. The tool is positioned graphically on the image during Setup and a distance is calculated. The tolerance is a percentage that determines the range values accepted during inspection.



### 6.9.2. Angle

**Description:** calculates the angle between two lines. Similar to the *Intersection locator*, but provides an angle as the result instead of a point.

**Positioning:**

- none, the user just has to assign the name of the two reference tools.

**Minimum / maximum angles:** minimum/maximum value accepted. If the angle is bigger the operating status is **Failure**.

**Tolerance:** parameter that can be used as an alternative to the angle constraints.

The tool is positioned graphically on the image during Setup and an angle is calculated.

The tolerance is a percentage that determines the range values accepted during inspection.

### 6.9.3. Comparatore

**Description:** compares the results of the other tools (e.g. *Caliper*) or the numeric values through a *major*, *minor* or *equal* comparison control.

**Positioning:**

- none, the user just has to assign the name of the two reference tools.

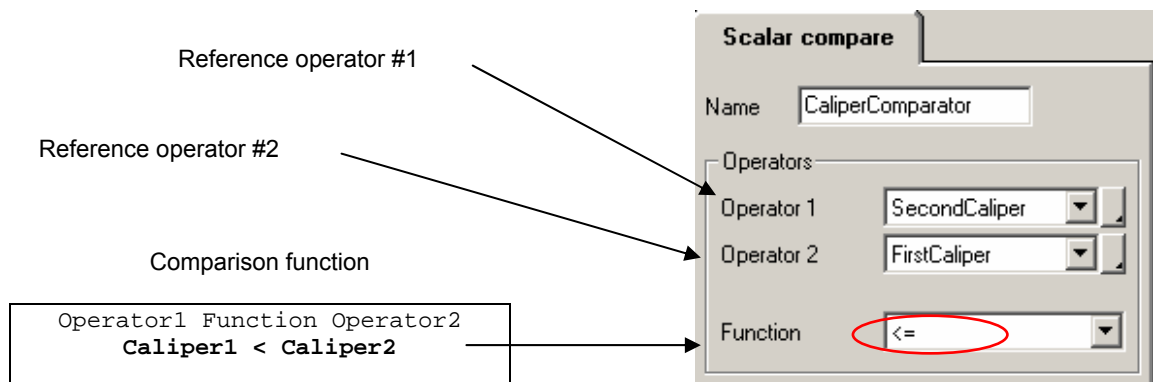
**Parameters:**

**Operator1:** first reference tool/value;

**Operator2:** second reference tool/value

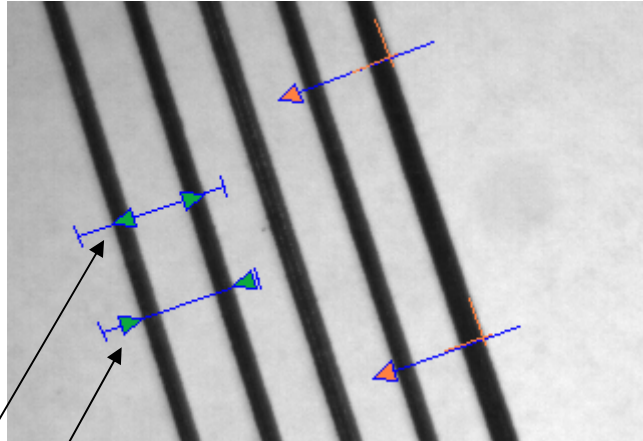
**Function:**

Comparison of operator1 respect to operator2.



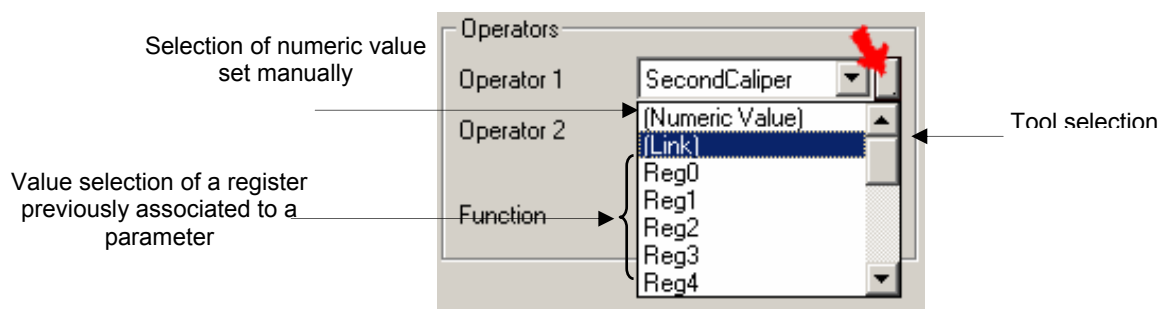


Example: the comparator can be used as follows:



Operation Name	Type	Status	Result
Trigger Mode	Continuous		
FirstCaliper	Internal caliper	OK	Distance = 60
SecondCaliper	External caliper	OK	Distance = 40
CaliperComparator	Scalar compare	OK	
edge1	Edge locator	OK	X = 240 Y = 252 Angle = 117
edge2	Edge locator	OK	X = 324 Y = 257 Angle = 111
EdgeDistance	Distance	OK	Distance = 84
CaliperDistanceComparator	Scalar compare	FAIL	

To set the numeric value of a tool, the user has to click on the grey box on the left side of the parameter tools and choose “**NumericValue**”. The values of parameters not associated to registers can be also compared. In this case, the user must choose the specific “**register**”. If a tool has to be replaced in one of the two previous alternatives, the user has to click on “**Link**”.



#### 6.9.4. Operatore logico

**Description:** carries-out logic operations on results obtained from the tools used in the previous steps. The foreseen functions are: *and*, *or*, *nand*, *nor*, *xor* e *xnor*.

The *Logic tool* replaces the *Test tool* and extends the *and* / *or* logic functions offering upto 5 inputs.

Differently from the *Test tool*, this tool doesn't consent the association of the result with one of the sensor outputs, due to the presence of a specific function (refer to section relative to the digital output).

**Positioning:**

- none, the user just has to assign the name of the reference tools.



The “truth tables” relative to 2 tool inputs *not*, *and*, *or*, *xor*, *nand*, *nor* :

**Not:** unary function that restores the complementary status (opposite) to the associated tool status.

Not status	Tool status
OK	Failure
Failure	OK

**And:** restores an OK status only if all the tool status are OK.

And status	Operator I status	Operator II status
OK	OK	OK
Failure	Failure	OK
Failure	OK	Failure
Failure	Failure	Failure

**Or:** restores an OK status only if one tool has an OK status.

Or status	Operator I status	Operator II status
OK	OK	OK
OK	Failure	OK
OK	OK	Failure
Failure	Failure	Failure

**Xor:** restores an OK status only if one tool has an OK status.

XOR status	Operator I status	Operator II status
Failure	OK	OK
OK	Failure	OK
OK	OK	Failure
Failure	Failure	Failure

**Nand:** restores an OK status when not all tools has an OK status.

Nand status	Operator I status	Operator II status
Failure	OK	OK
OK	Failure	OK
OK	OK	Failure
OK	Failure	Failure



**Nor:** restores an OK status when all tools have a FAILURE status.

Nor status	Operator I status	Operator II status
Failure	OK	OK
Failure	Failure	OK
Failure	OK	Failure
OK	Failure	Failure

**Xnor:** restores an OK status when the tools all have an OK status or all have a FAILURE status.

Xnor status	Operator I status	Operator II status
OK	OK	OK
Failure	Failure	OK
Failure	OK	Failure
OK	Failure	Failure

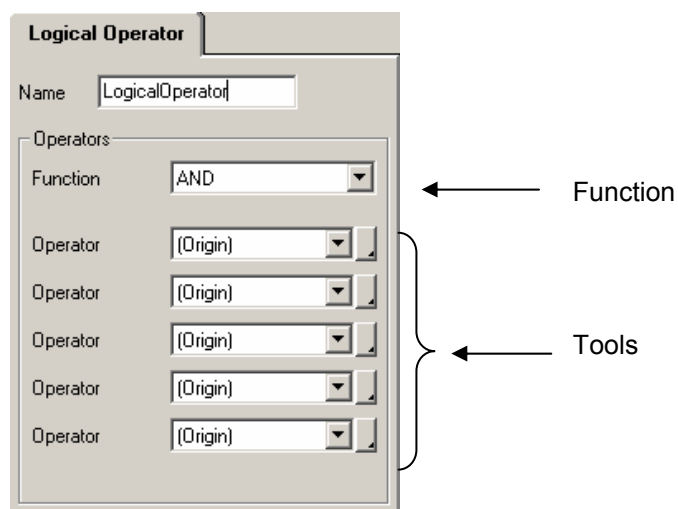
**Positioning:**

- none, the user just has to assign the name of the reference tools and the operation to carry-out.

**Parameters:**

**Operator1-5:** first reference tool/s.

**Functions:** logic tool selected





### 6.9.5. Esposimetro

**Description:** corrects the image light that depends on the light variations in a sample area. The light correction used an index denominated *Correction Factor*, defined as follows:

$$\text{Correction factor} = \frac{\text{Light value of sample image}}{\text{Light value of currently acquired image}}$$

The *correction factor* is used as a *multiplicative factor* to calculate the **reference threshold**<sup>2</sup> for the binarization algorithm of the *Blob Count*, *Blob Locator*, *Contour Match* tools (see pages **Errore. Il segnalibro non è definito.**, **Errore. Il segnalibro non è definito.**, **Errore. Il segnalibro non è definito.**).

**Reference threshold** = correction factor × threshold value calculated on the current image

The tool is represented by a rectangle representing the ROI where the current light value is calculated referred to the image present.

The *light value calculation* can be carried-out in the following modes (refer to *Figure 18*):

- *Median value*: the intermediate value (central) of the grey ROI tool levels when ordered in an ascending/descending order.
- *Average value*: the value obtained from the mathematic average of the grey ROL tool levels.
- *Minimum value*: the inferior value of the grey ROL tool levels.
- *Maximum value*: the highest value of the grey ROL tool levels.
- *Average percentage value*: the value obtained from the arithmetic average of the grey levels once eliminated the grey levels around the *minimum* and *maximum* values per a percentage value set as parameter.
- *Automatic threshold*: threshold value obtained from the binarization algorithm used by the device.

<sup>2</sup> The *Threshold* terms defines the light value, i.e. the grey level value calculated with the methods explained in this section.



**Parameters:**

**X:** *abscissa* on the top left side of the ROI.

**Y:** *ordinate* on the top left side of the ROI.

**Width:** ROI width.

**Height:** ROI height.

**Reference system:** absolute (origin) or relative (other tool) reference system.

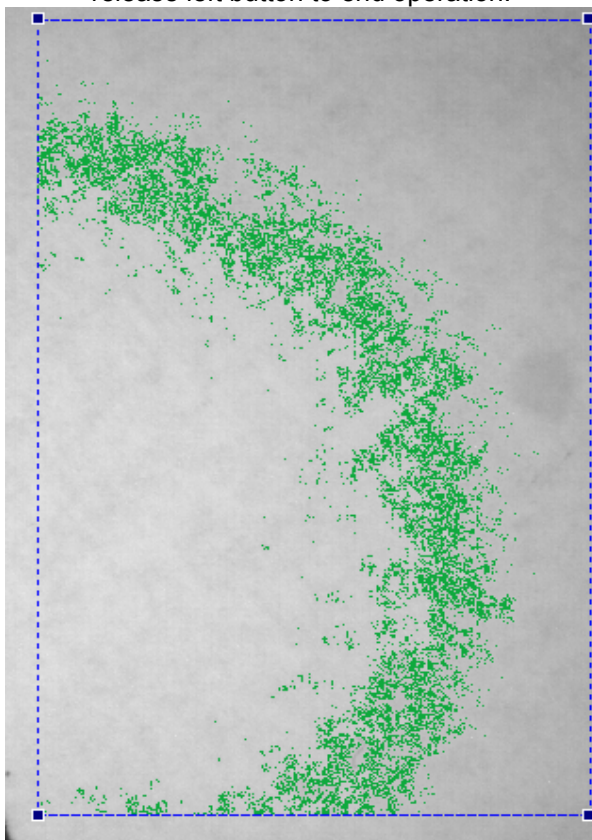
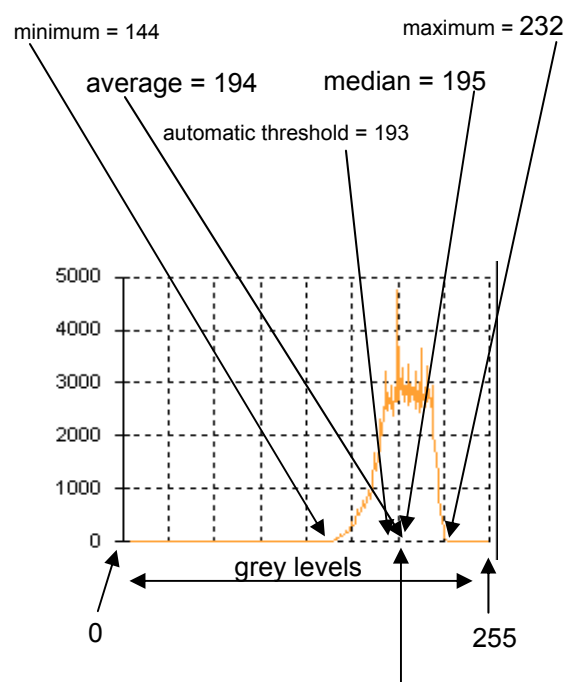
**Method:** calculation mode of the reference threshold

**Threshold:** range filter to set the control on the acquired image threshold value.

**Percentage:** percentage used in the *average percentage* mode of the threshold value.

**Positioning:**

- click on the left mouse button to position the top left peak of the tool ROI;
- drag to mouse to set tool dimension;
- release left button to end operation.

5% average (min - 5% / max - 95%) = 194

**Figure 18 – Calculation mode of the ROI light value**

This tool is a valid support in applications characterised by environmental conditions with a high light variation on the controlled area. These situations, tools using binarization algorithms are not very effective as they function in absence of light variation. Consequently if this tool is used together with the *Blob Count*, *Blob Locator*, *Contour Match* tools, the inspections become more reliable.

*Note:* the use of this tool becomes useless if used with tools unconditioned by light variations, such as *Edge locators* or similar.



## 6.10. Output

Tools used to configure result transmission on output interface.

### 6.10.1. Serial output

**Description:** consents message transmission on the serial interface of the sensor output<sup>1</sup> used containing the associated tool results.

**Positioning:** no graphic positioning available. The reference tool name has to be simply specified in the “Operator” field.

#### Parameters

**Operators:** reference tool. The results are transmitted to the serial port.



The results which are transmitted to the serial port can be of different typologies (please refer to section 7.5 to check message grammar and parameters of the response string). Different result typologies can be obtained for each tool configurable in the “Operator” parameter:

Locators		
	Response string	Examples
<b>Edge locator</b>	:nome_op,result,X,Y,angle;	:edge_locator,0,282.96,335.55,326.81;
<b>Composite locator</b>	:nome_op,result,X,Y,angle;	:composite_locator,0,275.35,155.50,357.52;
<b>Inclinometer locator</b>	:nome_op,result,X,Y,angle;	:inclinometer_locator,0,245.04,265.38,272.66;
<b>Intersection locator</b>	:nome_op,result,X,Y,angle;	:intersection_locator,0,154.05,184.97,271.10;
<b>Blob locator</b>	:nome_op,result,X,Y,angle;	:blob_locator,0,202.00,255.00,181.00;
<b>Pattern Match locator</b>	:nome_op,result,X,Y,angle;	:pattern_match,0,158.00,221.00,00.00 <sup>3</sup> ;
Controls		
<b>Edge</b>	:name_op,result,X,Y,angle;	:edge,0,325.00,359.94,90.00;
<b>Intersection</b>	:name_op,result,X,Y,angle;	:intersection,0,432.52,242.81,68.20;
<b>Inclinometer</b>	:name_op,result,X,Y,angle;	:inclinometer,0,160.32,223.79,283.98;
<b>Internal caliper</b>	:name_op,result,distance;	:internal_caliper,0,91.22;
<b>External caliper</b>	:name_op,result,distance;	:external_caliper,0,113.64;
<b>Blob Count</b>	:name_op,result, number of objects found;	:blob_count,0,001;
<b>Contour Match</b>	:name_op,result,difference;	:contour_match,0,02.00;
<b>Pattern Match Count</b>	:name_op,result,number of objects found;	:pattern_match_count,0,001;

Analysis		
	Results	Examples
<b>Distance</b>	:name_op,result,distance;	:distance,0,64.56;
<b>Angle</b>	:name_op,result,angle;	:angle,0,358.80;
<b>Test</b>	:name_op,result,out1,out2;	:test_and,0,1,0

*Note:* please consider that the operator result is given in a *negative logic*, i.e. **0** for **positive** results, and a number **different from 0** for **negative** results.

<sup>3</sup> The angle must not be considered and is zero by default.



### 6.10.2. Digital output

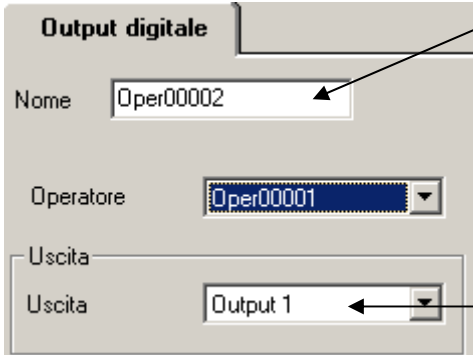
**Description:** Activates the digital outputs (output 1 or output 2) according to the status (OK/FAIL) of the associated tool.

**Positioning:**

None, the user just has to assign the name of the reference tool.

**Parameter**

**Tool:** reference tool whose status commands the activation of the associated digital output.



### 6.10.3. Image saving

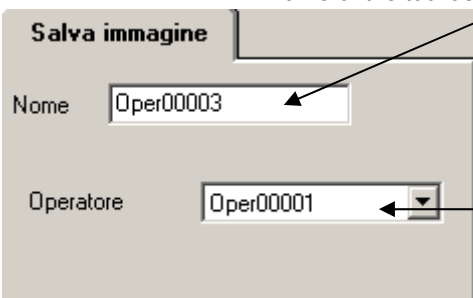
**Description:** saves on the SCS the image currently acquired according to the status (OK/FAIL) of the associated tool.

**Positioning:**

None, the user just has to assign the name of the reference tool.

**Parameter**

**Tool:** reference tool whose status commands image saving.



*Note:* this tool can be used together with the logic tool in order to condition the image saving of an inspection according to particular events (refer to section 6.2.5).



## 7. COMMUNICATION PROTOCOLS

### 7.1. Ethernet

The SCS1 smart camera presents a 100 Mb/s communication port that can be used for HOST PC and configuration interface communication.

Please contact DATASENSOR for further technical specifications.

### 7.2. Serial protocols and digital inputs

The SCS1 series offers three models presenting different communication protocols and supplementary digital inputs:

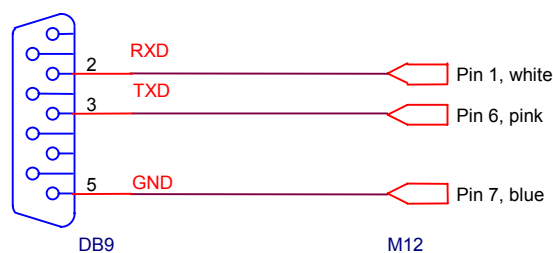
1. RS232 serial protocol
2. RS485 serial protocol
3. Digital input protocol

### 7.3. Electrical connections

- *SCS1 with RS232 serial protocol*: pin 1 of the 8-pole M12 connector dedicated to the transmission of data from SCS1 to PC and pin 6 to the data receipt.



Typical PC-SCS1 connection scheme:

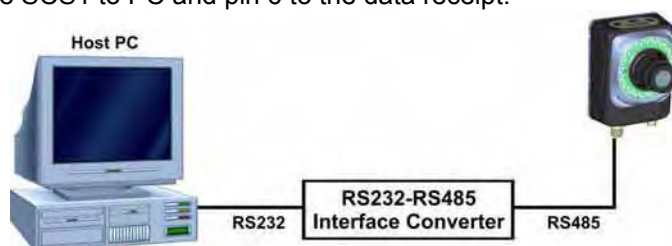


Host PC (DB9)	SCS1
TXD (3)	RS232 RX (pin 6, pink)
RXD (2)	RS232 TX (pin 1, white)
GND (5)	SCS1 GND (pin 7, blue)

Host PC (DB25)	SCS1
TXD (2)	RS232 RX (pin 6, pink)
RXD (3)	RS232 TX (pin 1, white)
GND (7)	SCS1 GND (pin 7, blue)

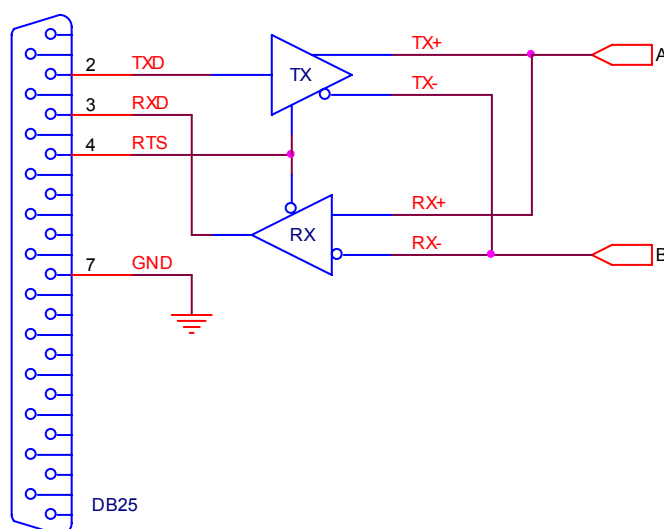


- *SCS1 with RS485 serial protocol:* pin 1 of the 8-pole M12 connector dedicated to the transmission of data from the SCS1 to PC and pin 6 to the data receipt.



The standard protocol implemented on the PC is RS232. If the SCS1 smart camera has to be used with the RS485 serial protocol, a RS232/RS485 serial adapter has to be used to guarantee interfacing with the sensor's receiver.

The following figure shows a typical RS232/RS485 conversion scheme:



Host PC (DB9)	Converter (RS232-DB25 side)
TXD (3)	TXD (2)
RXD (2)	RXD (3)
RTS (7)	RTS (4)
GND (5)	GND (7)

Converter (RS485 side)	SCS1
A	D+ (pin 6, pink)
B	D- (pin 1, white)

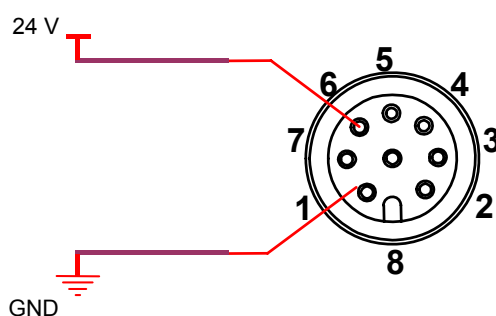
Host PC (DB25)	Converter (RS232-DB25 side)
TXD (2)	TXD (2)
RXD (3)	RXD (3)
RTS (4)	RTS (4)
GND (7)	GND (7)

Converter (RS485 side)	SCS1
A	D+ (pin 6, pink)
B	D- (pin 1, white)



- *SCS1 with supplementary digital inputs*: pin 1 and 6 are used as supplementary digital inputs able to select one of the four available inspections that can be saved on the sensor.

The figure represents one of the four possible input configurations:

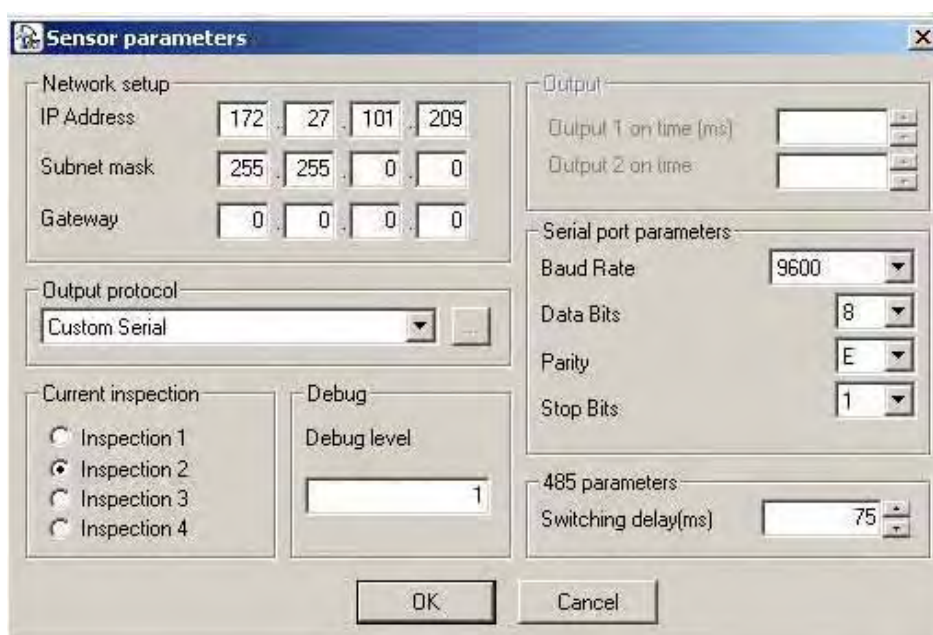


#### 7.4. Sensor serial communication parameters

The sensor serial communication parameters are configured using the SCS1 graphic interface. Ensure PC and sensor connection and select "Parameters" from the sensor menu:



The following window appears and the user can configure the sensor parameters for serial communication.





The parameters can be set both for the RS232 serial port as well as for the RS485 serial port:

- ✓ **Baud Rate:** transmission speed in bit/s (1200 ÷ 115200 bit/s).
- ✓ **Number of Data Bits:** bit number that form a word (7,8 bit).
- ✓ **Type of Parity used:** optional bit used to verify the integrity of the word transmitted (even, odd, none).
- ✓ **Number of Stop Bits:** number of bits used to separate two consecutive words (1,2 bits).

The RS485 serial port consents the configuration of only the following parameter:

- ✓ **Switching Delay:** switching delay of the *tx/rx* or *rx/tx* communication direction



## 7.5. Serial communication protocols (RS232/ RS485)

The serial communication protocol is based on a Master-Slave structure and the information exchange is carried-out in a query-response mode.

### 7.5.1. RS485 serial communication protocol

The RS485 serial protocol is a *half-duplex*.

At powering, the SCS1 (Slave) awaits the Master's commands (PC-PLC). The Master sends to the Slave a command according to a syntax described in Figure 24 (refer to paragraphs 7.5.4 and 7.5.5 for a detailed description of these messages). The Slave (SCS1) interprets the command received, carries-out the action requested and sends a *success* response to the Master, always according to the syntax described in Figure 25 and Figure 26. If the Slave is unable to interpret the command received or if the request needs an action that cannot be activated, the Slave will reply with a *failure* notice as described in Figure 25 and in Figure 26. *In particular*, the inspection results have to be explicitly requested to the SCS1 through the GETINSPRESULT- and GETNEXTINSPRESULT-command messages.

### 7.5.2. RS232 serial communication protocol

The RS232 serial protocol is a *full-duplex*.

At powering, the SCS1 (Slave) sends to the Master (PC-PLC) the inspection results, selected through the graphic interface (refer to paragraph 6.10.1), on the serial output. Differently from the RS485 serial protocol, the inspection result do not have to be explicitly requested to the SCS1 using the following command messages: GETINSPRESULT- and :GETNEXTINSPRESULT-.

Due to the RS232's full-duplex specifications, SCS1 continuously awaits all other command messages. Also in this case, the Master in order to communicate with the Slave sends a command according to the syntax described in Figure 24 (refer to paragraph paragraphs 7.5.4 and 7.5.5 for a detailed description of the command messages). The Slave (SCS1) interprets the command received, carries-out the action requested and sends a *success* response to the Master, always according to the syntax described in Figure 25 and Figure 26. If the Slave is unable to interpret the command received or if the request needs an action that cannot be activated, the Slave will reply with a *failure* notice as described in Figure 25 and in Figure 26.

### 7.5.3. Application examples of serial protocol command messages<sup>4</sup>

The following protocol commands are available (sent from Master to Slave):

The protocol command messages are divided into **two categories**:

1. Command messages for **information request**
2. Command messages for sensor **configuration**

The following command messages belong to the first category: CHKSERLINE-, :GETCURINSP-, :GETINSPRESULT-, :GETNEXTINSPRESULT-.

The following to the second category: SETCURINSP<inspection number>-<sup>5</sup>

<sup>4</sup> The images presented in this section, concerning the serial protocol application examples, show the graphic interface of a typical application used for serial communication between a PC with the SCS1.

<sup>5</sup> For a more detailed description please refer to sections 7.5.4 and 7.5.5.



#### 7.5.4. Command messages for information request

##### :CHKSERLINE-

Answer = **:OK**

Function = consents the status control of the serial communication between Master (PC-PLC) and Slave (SCS).

Action = none

##### Example:

Transmission of command message :CHKSERLINE-

Receipt of answer message :OK

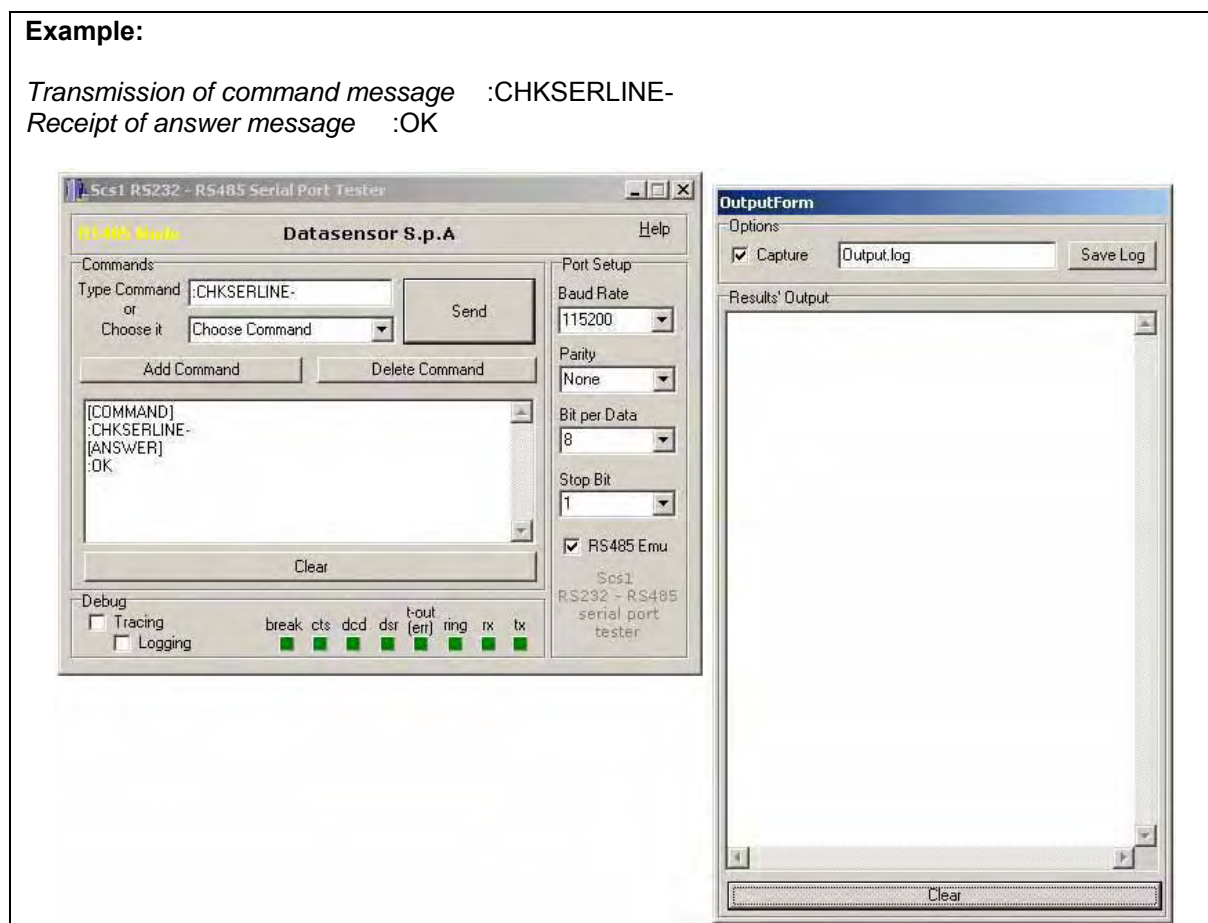


Figure 19 – CHKSERLINE- application example

##### :GETCURINSP-

Answer = **OK& <Inspection index being carried-out>**

<Inspection index being carried-out> = **[1 – 4]**

or Answer = **ERR&<Error Code>**

<Error Code> = **PARAMNOTVALID**

Function = identifies the inspection number (1-4) currently being carried-out.

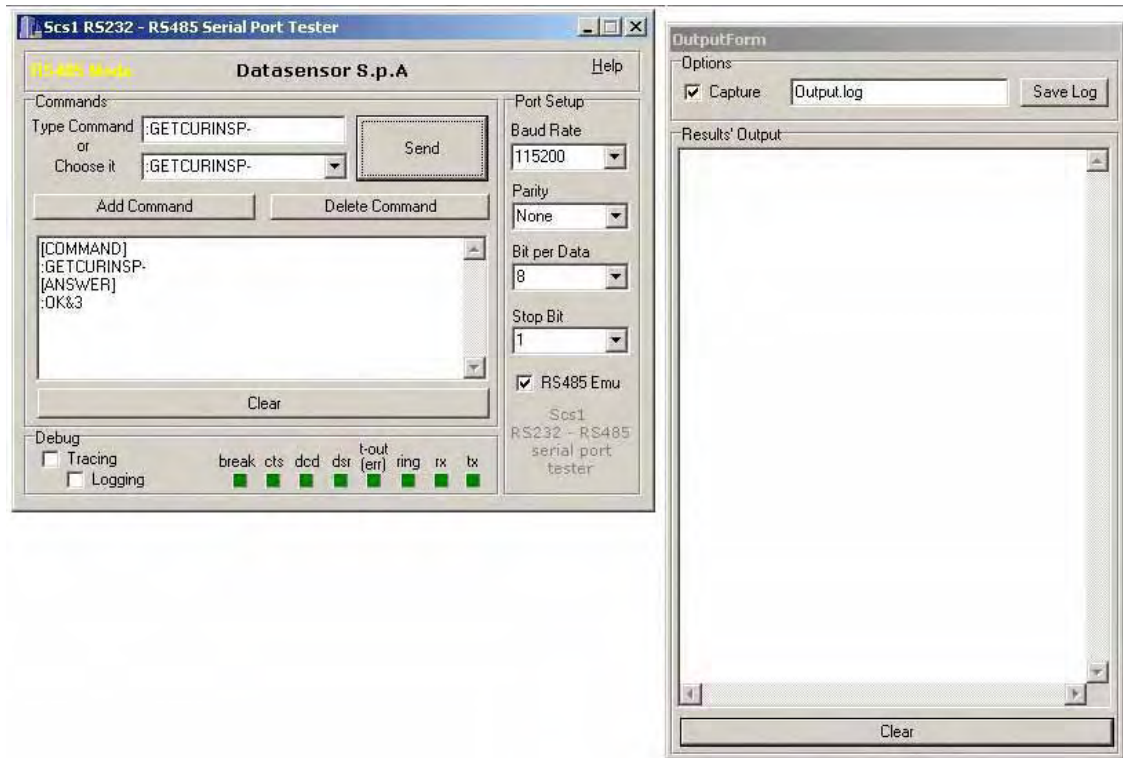
Action: none



**Example:**

*Transmission of command message* :GETCURINSP-

*Receipt of answer message* :OK&3 (hypothetically considering that SCS1 is carrying-out inspection 3 during the receipt of the aforementioned message)



**Figure 20 - GETCURINSP- application example**

**:GETINSPRESULT-**

*Function* = requests to SCS1 the results of the last inspection carried-out.

*Answer* = returns the results of the configured operators to the serial output according to the syntax previously described.

or Answer = **ERR&<Error Code>**

<Error Code> = **INVALIDSTATUS**  
**NOSERIALOUTPUT**

*Action* = none

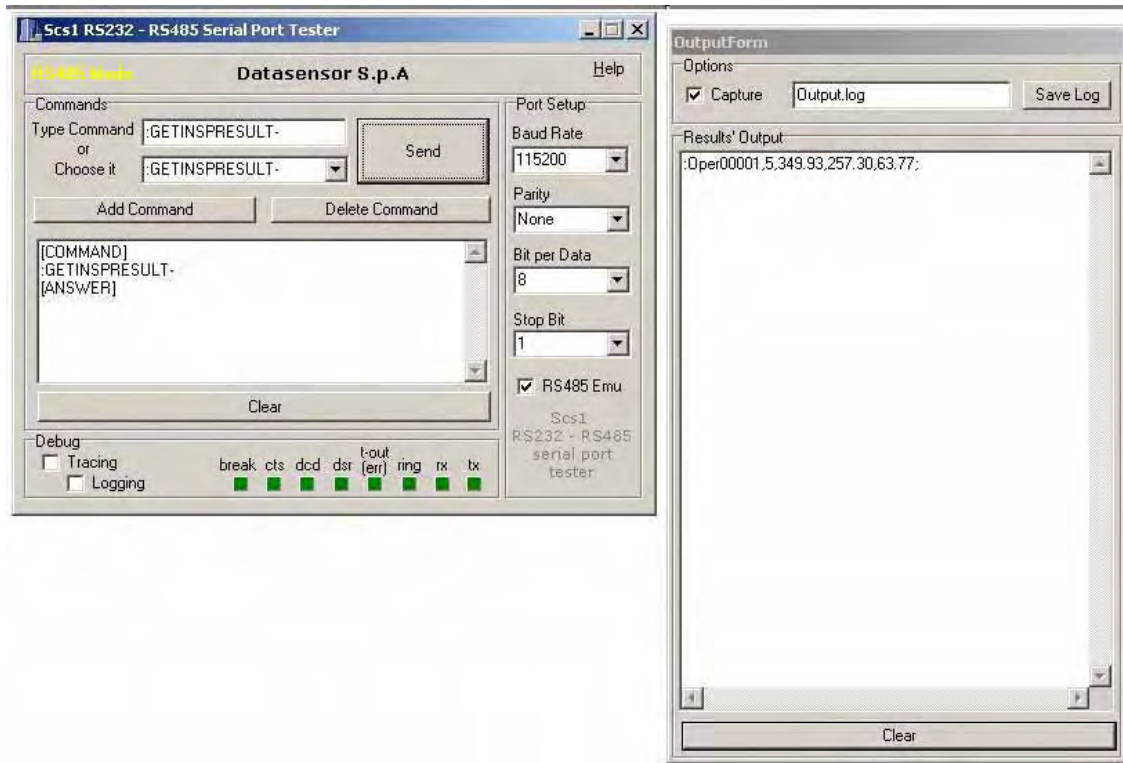


**Example:**

*Hypothesis:* consider an inspection configured with an *Edge locator* and the result on a serial output (refer to paragraph 6.7.1 and 6.10.1).

*Transmission of command message* :GETINSPRESULT-

*Receipt of answer message* :Oper00001,5,349.93,257.30,63.77;



**Figure 21 - GETINSPRESULT- application example**

### :GETNEXTINSPRESULT-

*Function* = requests to SCS1 the results of the next inspection to carry-out.

*Answer* = returns the results of the configured operators to the serial output according to the syntax previously described.

or Answer = **ERR&<Error Code>**

<Error Code> = **INVALIDSTATUS**  
**NOSERIALOUTPUT**

*Action* = none

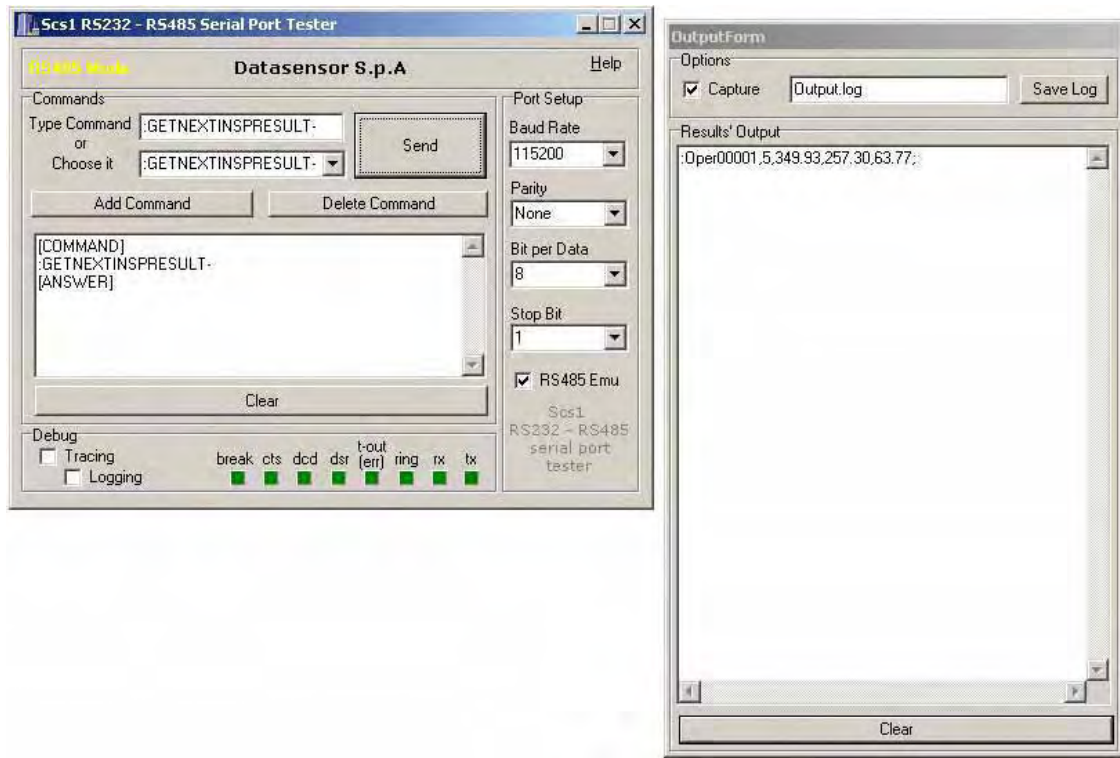


**Example:**

*Hypothesis:* consider an inspection configured with an *Edge locator* and the result on a serial output (refer to paragraph 6.7.1 and 6.10.1).

*Transmission of command message* :GETNEXTINSPRESULT-

*Receipt of answer message* :Oper00001,5,349.93,257.30,63.77;



**Figure 22 - GETNEXTINSPRESULT- application example**

#### 7.5.5. Command messages for sensor configuration

**:SETCURINSP&<Inspection index to carry-out >-**

< Inspection index to carry-out > = **[1 – 4]**

Answer = **:OK**

or Answer = **ERR&<Error Code>**

<Error Code> = **PARAMNOTVALID**

**INVALIDSTATUS** (sensor already connected via Ethernet to another device, incompatible status)

*Function* = consents the configuration of the inspection to carry-out (1-4).

*Action* = inspection currently carried-out is configured according to command parameter request.

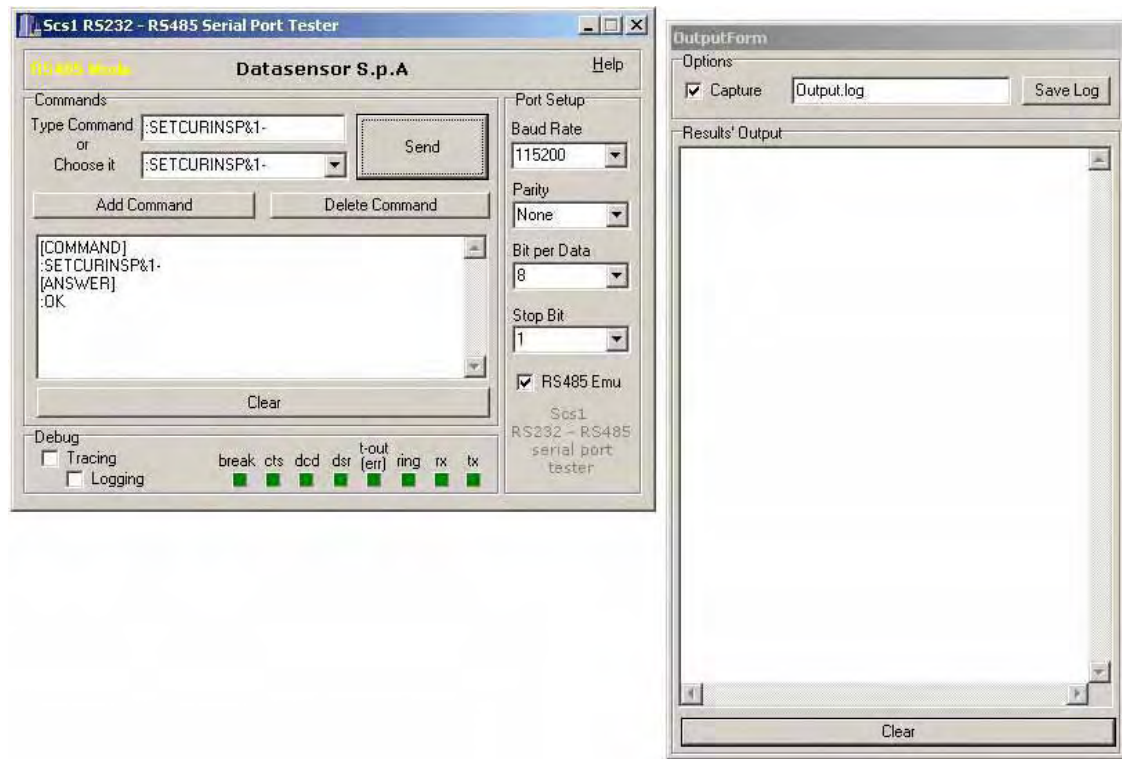


**Example:**

Transmission of command message :SETCURINSP&1-

Receipt of answer message :OK

Action SCS1 carries-out inspection 1 after command and response receipt



**Figure 23 -SETCURINSP&1- application example**

#### 7.5.6. Grammar legend–BNF note

The command and response messages are written in a BNF (Backus and Naur form) syntax. The different meanings are given below:

- ::= means “is defined as”
- | means logic “or”
- <> used to enclose the name of a macrocategory or a *non-terminal* symbol
- [] used to enclose optional terms
- {} used to enclose repeated terms
- “” used to enclose *terminal* symbols

Example: The grammar of command messages given in Figure 24 can generate the following command message:

**:GETNEXTINSPRESULT-**

format from <prologue> “:”, from’ <idCommandWithoutParameters> **GETNEXTINSPRESULT**, and from <epilogue> “-”; or the following command message

**:SETCURINSP&4-**



format from <prologue> “:”, from’ <idCommandWithParameters> **SETCURINSP**, from’<ampersand> &, from <inspectionnumber> 4 and from <epilogue> “-”. While for example, the following command message cannot be generated

~~:SETNEXTCURINSP-~~

as it would be formed from the <prologue> “:”, from’ <idCommandWithoutParameters>, and from <epilogue> “-”; but as shown in Figure 24, the <id command> used doesn’t exist in the grammar.

#### 7.5.7. Command message format

The *message* format foresees a command beginning character (*prologue*, ASCII character “:” (0x3A)) and a command end character (*epilogue*, ASCII character “-” (0x2D)). A generic command presents the following form:

*Message example* = **:Command-**

Where Command represents a null-terminated string of ASCII characters composed of a command *identifier* and “&” *separation character* (0x26) followed by the parameters to be transmitted.

*Command example* = **SETCURINSP&2**

The general syntax of the command message is given below:

```

<Command message> ::= <prologue><command><epilogue>

<prologue> ::= ":" (0x3A in hexadecimal)

<command> ::= <CommandWithParameters> | <CommandWithoutParameters>

<CommandWithParameters> ::= <idCommandWithParameters><ampersand><parameter>

<CommandWithoutParameters> ::= <idCommandWithoutParameters>

<idCommandWithParameters> ::= "SETCURINSP"

<idCommandWithoutParameters> ::= "CHKSERLINE" | "GETCURINSP" |
                                   "GETINSPRESULT" | "GETNEXTINSPRESULT"

<ampersand> ::= "&" (0x26 in hexadecimal)

<parameter> ::= <inspection number>

<inspection number> ::= "1" | "2" | "3" | "4"

<epilogue> ::= <minus><NULL>

<minus> ::= "-" (0x2D in hexadecimal)

<NULL> ::= "\0" (0x00 in hexadecimal)

```

Figure 24 – Command message grammar



### 7.5.8. Response Message format

SCS1 responds positively or negatively to each command, if it can or can not satisfy the received request. An error notice is given in case of a negative response. Possible responses are given below:

- **:OK** = command carried-out correctly
- **:ERR&BADCOMMAND** = command string not identified correctly
- **:ERR&PARAMNOTVALID** = command string identified, but operating parameter is outside range
- **:ERR&INVALIDSTATUS** = command correct, but not compatible with the current system status
- **:ERR&NOSERIALOUTPUT** = results of an inspection without *serial output* configuration are requested to the SCS1.

The *response* format varies according to the command transmitted. In all cases the *response* is a null-terminated string of ASCII characters and foresees a response beginning character (*prologue*) (ASCII character ":" (0x3A)).

If the response requires a return value, this value is transmitted as a parameter in the same command message format (refer to paragraph 7.5.7), i.e. the response contains an error *identifier* and a "&" (0x26) *separation character* (0x26) followed by the parameters of the results requested or an error code.

*Response example without errors* = **:OK&1**

*Response example with errors* = **:ERR&INVALIDSTATUS**

The syntax of the :CHKSERLINE-, :GETCURINSP- and :SETCURINSP- response – command messages is the following:

```

<response message> ::= <prologue><type><epilogue>

<prologue> ::= ":" (0x3A in hexadecimal)

<type> ::= <success> | <failure>

<success> ::= <SuccessID>[<ampersand><SuccessParameter>]

<failure> ::= <FailureID><ampersand><FailureParameter>

<SuccessID> ::= "OK"

<FailureID> ::= "ERR"

<ampersand> ::= "&" (0x26 in hexadecimal)

<SuccessParameter> ::= <InspectionNumber>

<FailureParameter> ::= <ErrorCode>

<InspectionNumber> ::= "1" | "2" | "3" | "4"

<ErrorCode> ::= "BADCOMMAND" | "INVALIDSTATUS" |
                "PARAMNOTVALID" | "NOSERIALOUTPUT"

<epilogue> ::= "\\0" (0x00 in hexadecimal)

```

**Figure 25 – Grammar of the :CHKSERLINE-, :GETCURINSP-, :SETCURINSP- response-command messages**



The response messages to the :GETINSPRESULT- and :GETNEXTINSPRESULT- command messages are slightly different as the SCS1 has to return the results of the current inspection, which depend on the inspection type. In particular, the responses of these commands depend on the number and type of operators configured in an inspection (refer to section 6.10.1).

The syntax of the :GETINSPRESULT- and :GETNEXTINSPRESULT- response – command messages is the following:

```

<response> ::= <result>{<result>}<epilogue>

<result> ::= <prologue><OperatorName><comma>
              <OperatorResult><OperatorParameter><semicolon>

<prologue> ::= ":" (0x3A in hexadecimal)

<OperatorName> = NULL-terminated string

<comma> ::= "," (0x2C in hexadecimal)

<OperatorResult> = "0" | "1"

<OperatorParameter6> ::= <OperatorParameter>{<OperatorParameter>}

<OperatorParameter> ::= <digit>{<digit>}{<dot><digit><digit>}

<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<digit > ::= "." (0x2E in hexadecimal)

<semicolon> ::= ";" (0x3B in hexadecimal)

<epilogue> ::= <line feed><NULL>

<line feed> ::= 0x0A (in hexadecimal)

<NULL> ::= "\0" (0x00 in hexadecimal)

```

**Figure 26 – Grammar of the :GETINSPRESULT-, :GETNEXTINSPRESULT- response-command messages**

*Note:* A time-out period is available. Consequently, if the terminating message character (-) is not read by the Slave within 5 seconds from the reading of the beginning message character (:), the partially received command is aborted and the Slave is put in stand-by for other Master requests.

Moreover, a standard response message is available to signal a command which correctly formatted according to the syntax previously described but unidentified by the serial protocol. In this case the following message is obtained as the command response: **:ERR&BADPARAM.**

*Example of command which is syntactically correct but unidentified = :CHKBAUDRATE-*



### 7.6. Digital input communication protocol


The models, presenting this function, allow the selection of the inspection to be carried-out by correctly setting the electric levels of the pin 1 and 6 of the 8-pole connector. These models do not have a serial interface.

The correspondence between the input status and the inspection carried out is summarised in the following table:

Input 1 (pin 1)	Input 2 (pin 2)	Selected inspection
0 V	0 V	1
0 V	24 V	2
24 V	0 V	3
24 V	24 V	4



## 8. PERIODICAL CHECKS AND MAINTENANCE


-  For correct system maintenance, the hardware must be kept free from dust and the SCS1 software must be updated when new releases are issued.

Remove dust or dirt accumulated on the sensor using a soft cloth, dampened with a neutral detergent solution if necessary.

-  Avoid dirt on the imager (area behind lenses).

Use anti-static compressed air to remove dust for periodical maintenance or when necessary.

The latest SCS1 software release is available on the website: [www.datasensor.com](http://www.datasensor.com).


-  The Smart Camera Sensor SCS1 does not require particular maintenance, with the exception of the cleaning of the protective surfaces of the optics.  
Use a cotton cloth dampened with water for cleaning.

**Do not use under any circumstances:**

- alcohol or solvents;
- wool cloths or synthetic fabric.

### 8.1. Warranty

The Smart Camera Sensor SCS1 has a 36 month guarantee from the manufacturing date. DATASENSOR will not be liable for any damages to persons and things caused by the non-observance of the correct installation modes and device use.

-  The warranty will not cover damages caused by incorrect installation, incorrect use and accidental causes such as bumps or falls.

In presence of problems, please contact the DATASENSOR SpA Technical Support:

**Technical Support**

Tel.: 0039 051/6765611

Fax: 0039 051/6759324

E-mail: [service@datasensor.com](mailto:service@datasensor.com)



## 9. TECHNICAL DATA

### SCS1:

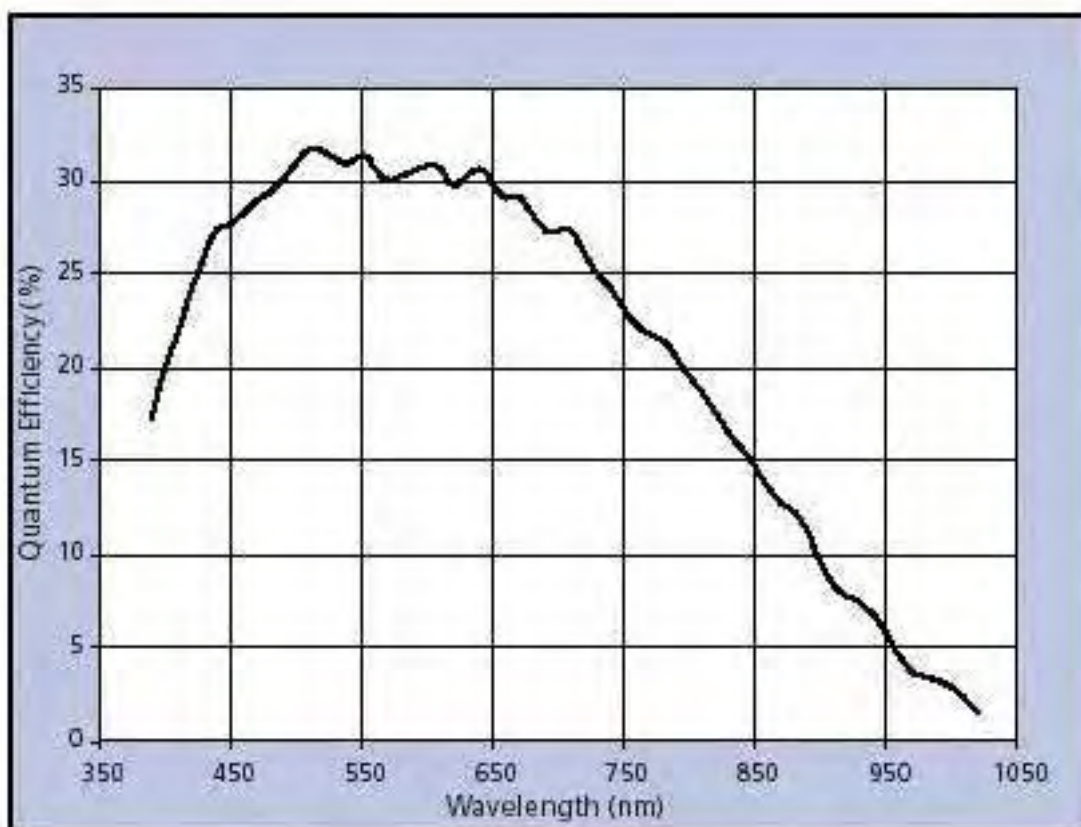
Power supply:	24 Vdc $\pm$ 10%
Ripple:	1 Vpp max with illuminator / 2 Vpp max without illuminator
Consumption (output current and illuminator excluded):	120 mA at 24 Vdc
Outputs:	2 PNP outputs (short-circuit protection)
Network interface:	Ethernet RJ45 - 10/100 Mbs
Serial interface:	RS232 / RS485
External illuminator interface:	M8 4-pole connector, 24 Vdc illuminator power supply, strobe signal (TTL)
Output current:	100 mA max
Saturation voltage:	< 2 V
Image sensor:	CMOS 6.61mm x 4.97mm (1/2") format, 640 x 480 pixel resolution (VGA) 9.9 micrometer pixel dimension
Lenses:	Standard CS-mount or C-mount with adapter
Dimensions:	112.4 mm x 75 mm x 40 mm (without illuminator), 112.4 mm x 75 mm x 55 mm (with illuminator)
Indicators:	4-digit display, 3 green LEDs, 1 yellow OUTPUT LED
Setting:	SET push-button, + and - push-buttons Auxiliary Ethernet PC graphic user interface
Data retention:	Non-volatile FLASH memory
Operating temperature:	-10 ... +55°C
Storage temperature:	-25 ... +70°C
Vibrations:	
Shock resistance:	
Housing material:	Aluminium alloy
Mechanical protection:	IP40
Connections:	RJ45, M12 8-pole, M8 4-pole
Weight (lenses excluded):	300 g without illuminator, 385 g with illuminator

### Image sensor (Micron MT9V403):

Technology:	CMOS
Power supply:	3,3 V
Format:	1/2-inch (6,61 mm x 4,97mm)
Array format:	659 x 494 with windowing capability
Pixel size:	9.9 $\mu$ m x 9.9 $\mu$ m
Maximum frame rate:	200 fps
Shutter exposure time:	20 $\mu$ s - 1,3 sec with 66 Mhz clock
Gain:	1x-18x
SNR:	45 dB
Responsivity:	2.0 V/lux-sec con illuminazione a 550 nm



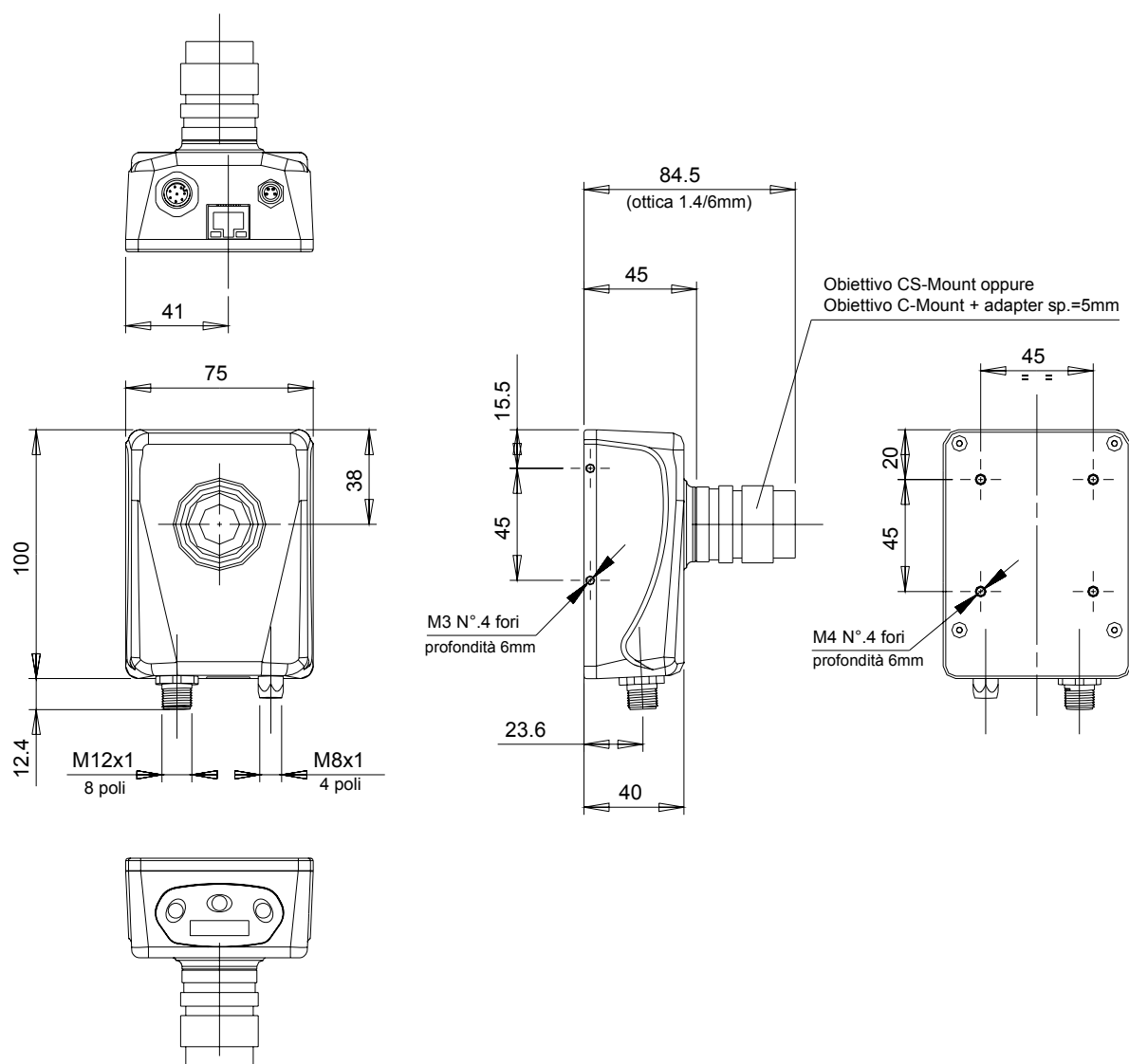
Quantum efficiency





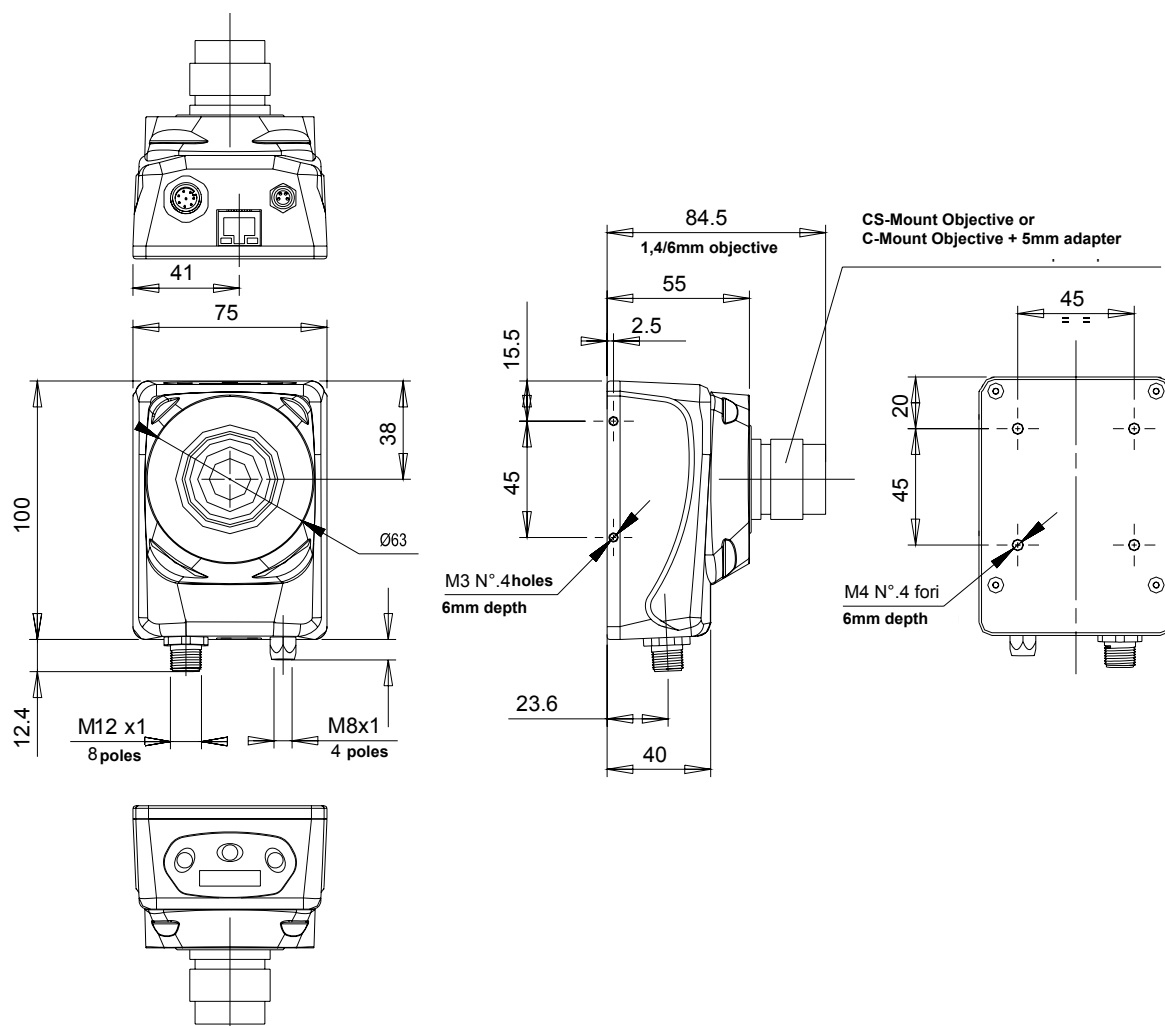
## 10. DIMENSIONS

### 10.1. SCS1 dimensions without illuminator





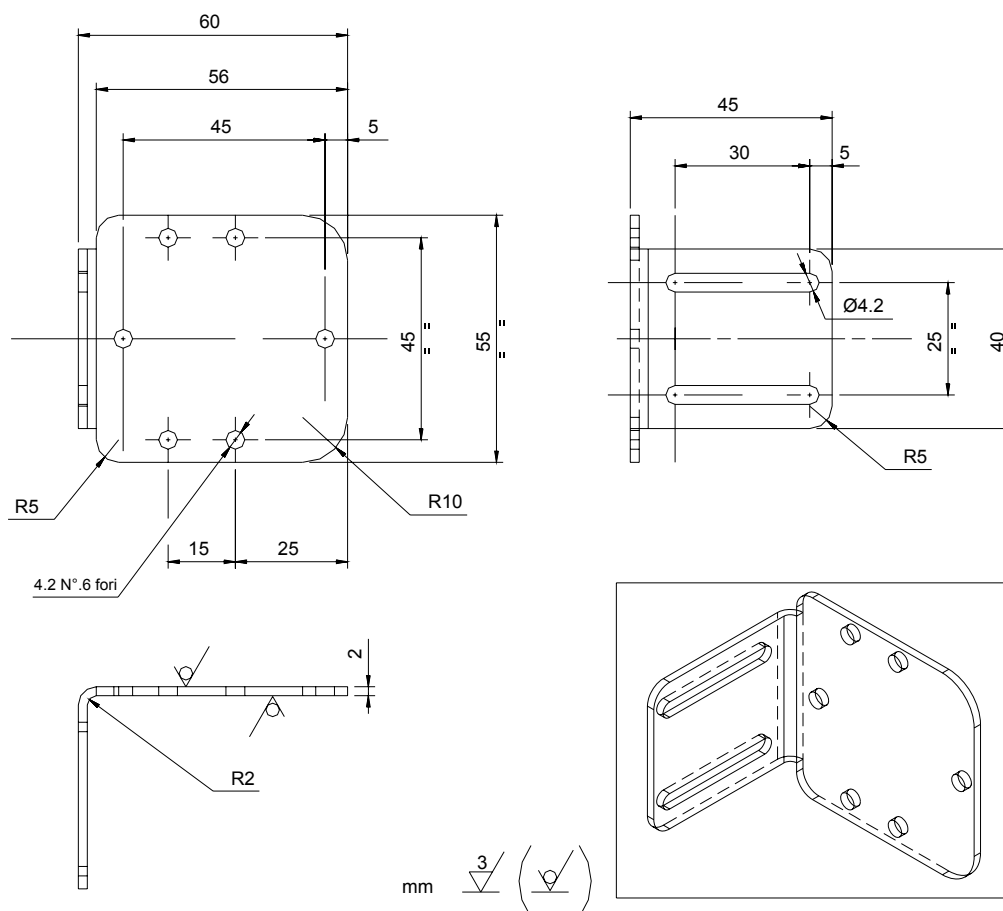
## 10.2. SCS1 dimensions with illuminator



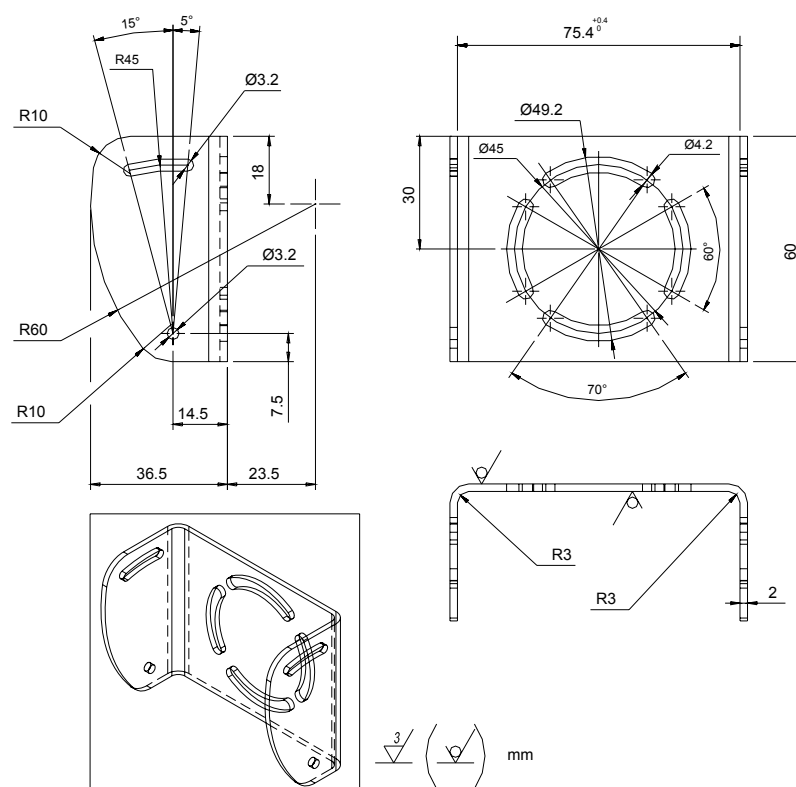


## 10.3. Fixing bracket dimensions

## ST-5047



## ST-5048





## 11.ACCESSORIES

### 11.1. Illuminators

Model	Description
SCS-IL-R635-22C-024-1012	back illuminator, 24 Vdc, continuous red light
SCS-IL-R635-22P-024-1012	back illuminator, 24 Vdc, pulsed red light
SCS-IL-R635-31C-024-1012	top illuminator, 24 Vdc, continuous red light
SCS-IL-R635-31P-024-1012	top illuminator, 24 Vdc, pulsed red light
SCS-IL-R660-12C-024-0007-T	ring illuminator, 24 Vdc, continuous red light
SCS-IL-R660-12P-024-0007-T	ring illuminator, 24 Vdc, pulsed red light
SCS-PS-IL-024	strobe power supply unit 24 Vdc

**Ring light (SCS-IL-R660-12C-024-0007-T / SCS-IL-R660-12P-024-0007-T):**

provides diffused continuous or pulsed red light over a small area in front of the sensor. The illuminator has a circular shape with a hole in middle. The sensor lens is mounted directly on the optics.



**Back light (SCS-IL-R635-22C-024-1012 / SCS-IL-R635-22P-024-1012):**

provides low-intensity continuous or pulsed light behind the target-object. Object contour and shape are highlighted, but not the surface characteristics.



**Top light (SCS-IL-R635-31C-024-1012 / SCS-IL-R635-31P-024-1012):**

provides uniform continuous or pulsed light in a concentrated area. The illuminator is positioned behind the sensor and, if oriented correctly, very precise inspections can be carried-out.



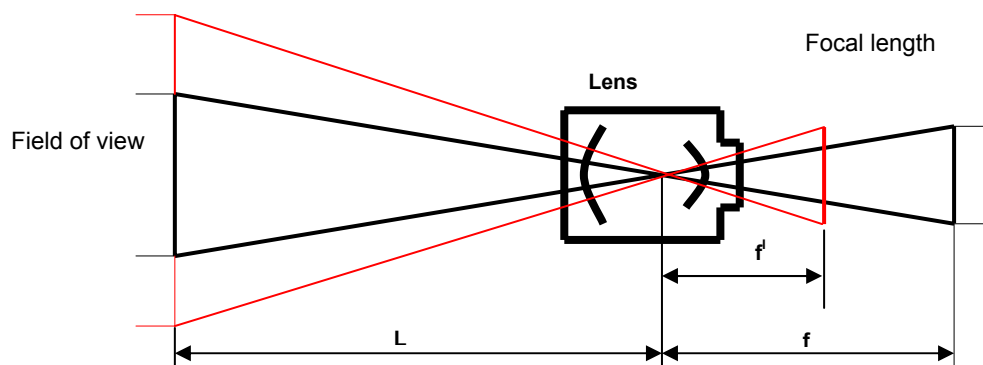


## 11.2. Lenses

The lenses, available as accessories, present a shutter and adjustable focus. The different focal lengths (6 mm, 8 mm, 12 mm, 16 mm) guarantee flexibility in terms of operating distance and field of view dimensions.

Smaller the focal length, the smaller the operating distance allowed and the larger the field of view.

For a more details refer to “Appendix B”.



The following table lists the available DATASENSOR lenses:

Model	Description
SCS-LE-V06-C-V	6 mm lens
SCS-LE-V08-C-V	8 mm lens
SCS-LE-V12-C-V	12 mm lens
SCS-LE-V16-C-V	16 mm lens



### 11.3. Fixing brackets

According to the application and sensor support, the user can use different rigid brackets available as accessories.

Two types of fixing brackets are available:

- *Linear adjustable fixing bracket.*

This is a rigid fixing bracket composed of two parts rotated at 90° respect to each other.

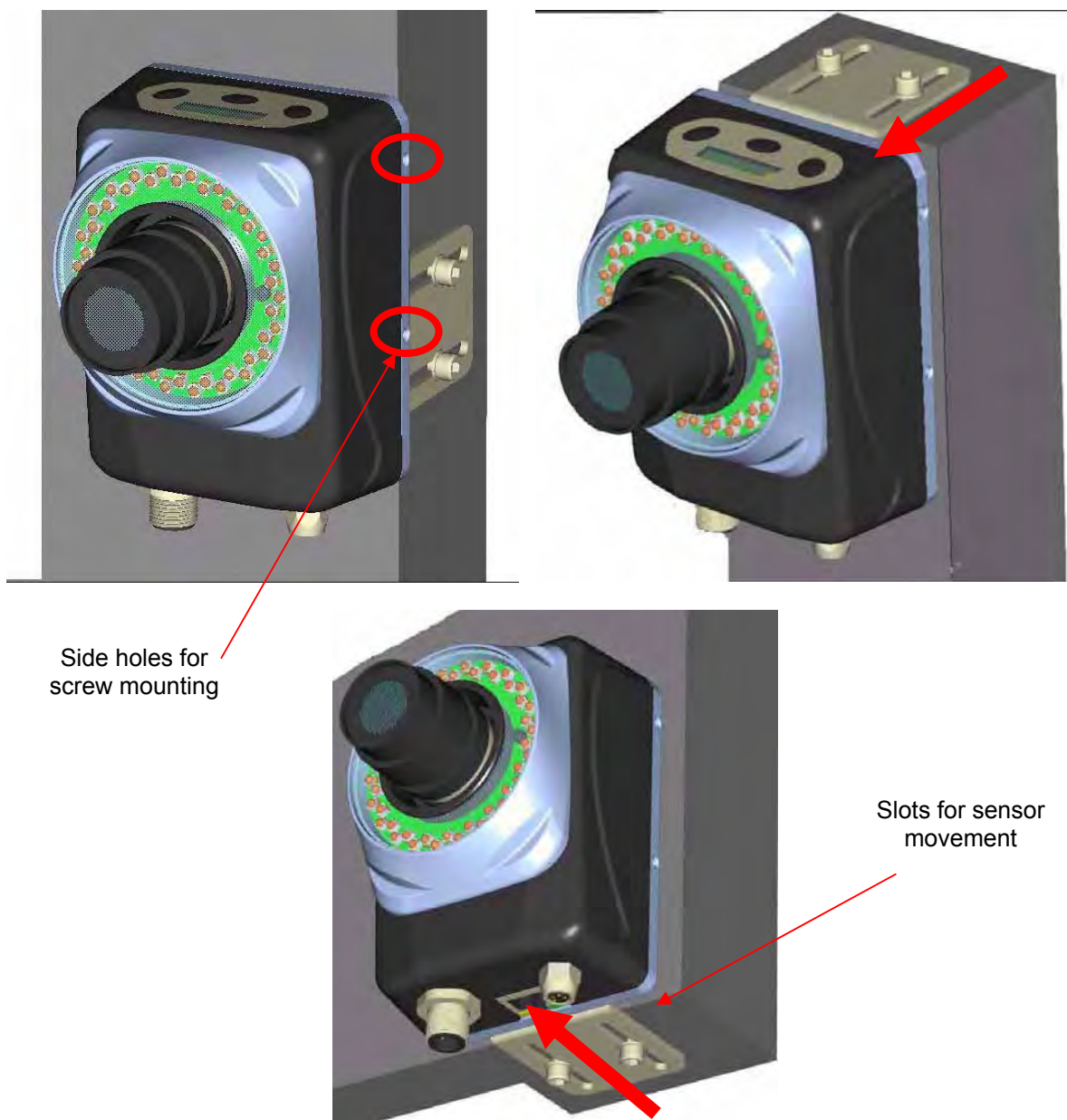
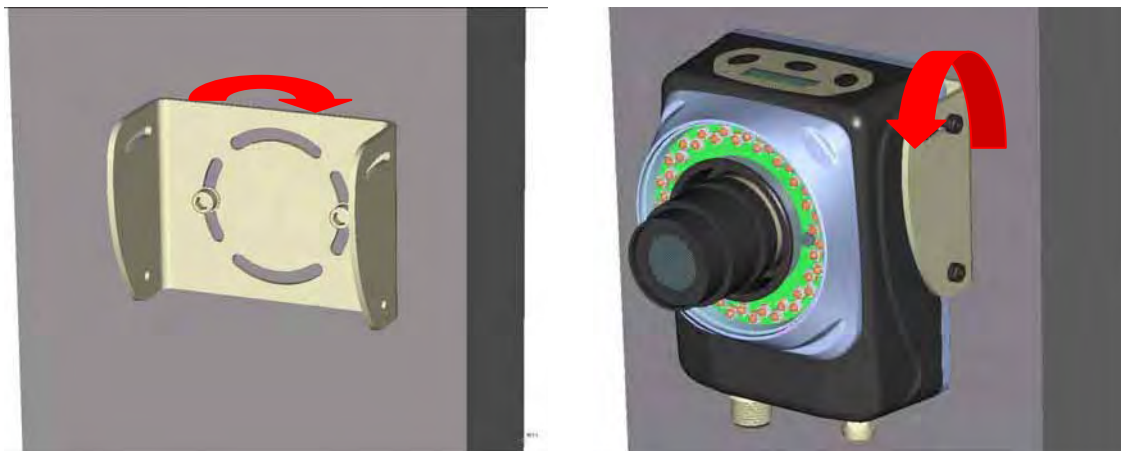


Figura 27



- *Angled adjustable fixing bracket.*

Fixing bracket that allows sensor rotation and inclination.



Note : consider very carefully the use of fixing brackets in presence of strong vibrations.



## 12. APPENDIX A: IMAGE ELABORATION TECHNIQUES

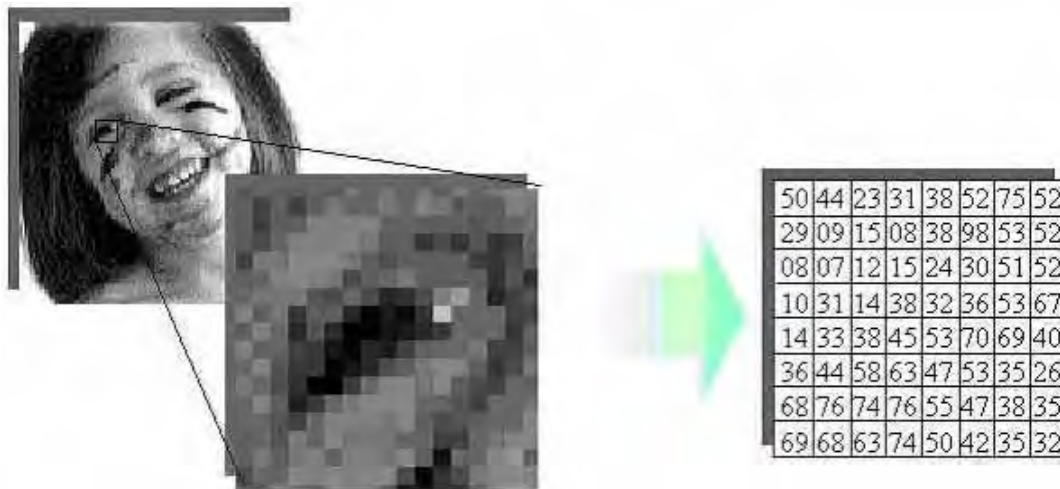
### 12.1. General concepts: digital image

**What is a digital image?** A digital image is simply a matrix of numbers.

Each matrix element is called picture's element or pixel. ( e.g.: SCS1 resolution is 640x480 and consequently the image is composed of 640 columns and 480 lines, forming a total of more than 300000 pixels).

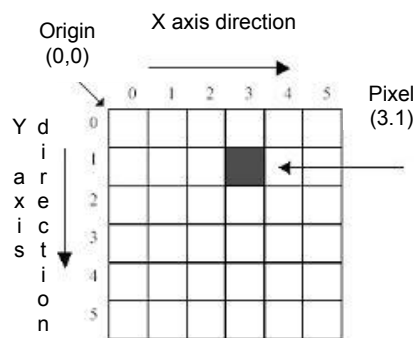
Each pixel can be considered a source of information. In greyscale systems, each pixel provides a numeric value included between 0 and 255, corresponding to the brightness of the selected pixel. 0 corresponds to black, 255 to white. In colour systems, each pixel supplies three different brightness values: one relative to red, one to blue and one to green. These three values form the RGB value (Red, Green, Blue) of the pixel.

Each colour can be represented theoretically by a specific combination of these three basic colours.



The pixel part inside the image is identified by the coordinates (X, Y).

The system origin is positioned conventionally in the top right angle above the image.



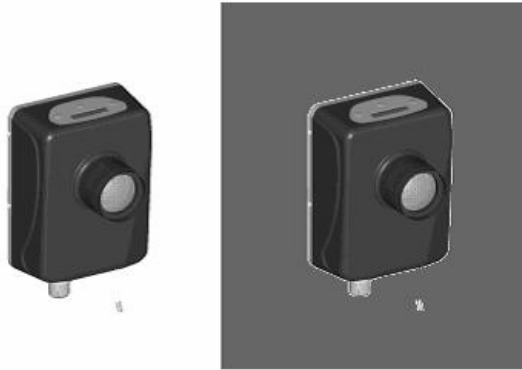
### 12.2. Machine vision

“Machine vision” refers to a machine equipped with a set of visible tools. These machines, exploiting their “sight”, inspect an object tracking defects and missing parts as well as making measurements, reading codes and other similar applications. The machine eye is a sensor that, according to the user instructions, acquires object images and gathers all the information necessary to carry-out the elaboration.



### 12.3. Binarization

Operation that converts a greyscale image into a monochrome one. An image formed by black and white pixels is obtained. Only the most essential information of the initial image is maintained. In most cases, pixels belonging to the same object have very similar grey values and after the binarization operation are gathered together in a white or black area. Different operations can be thus carried-out with only a reduced memory load. The setting of the most appropriate threshold is fundamental.



The figure given above shows the same object on two different backgrounds. In the first image, the image-background contrast is very evident, whereas in the second image the distinction is less evident.

Let's consider an histogram, with an abscissa indicating the 256 possible grey tones (0 corresponds to black, 255 to white) and an ordinate with the corresponding pixel number of each value. In the first case, two distinct distributions will be generated as far as the object and the background pixels are concerned, whereas in the second case the pixel distributions will be overlap partially. The best solution is to carefully set the threshold between the peak values of both distributions.

### 12.4. Inspection timing

To determine the elaboration time necessary for an inspection, three factors have to be considered. These factors correspond to the three inspection phases: exposure, acquisition and inspection time.

The exposure time indicates the period of time the device remains exposed to light during image acquisition. Longer the exposure time, the more light enters into the device. Exposure time is determined by three factors:

- *Overall speed of the parts to inspect:* in presence of high speed, a low exposure time is recommended to avoid obtaining a blurred image.
- *Overall inspection rhythm:* a high inspection rhythm requires necessarily a low exposure time.
- *Available light:* optimum lighting requires a low exposure time and vice versa.

At times the reduction of the exposure time may be required, but the overall image brightness has to be maintained. If necessary, proceed as follows:

- Increase the light aimed on the inspection area.
- Increase the CMOS gain: the gain is the output/input ratio and in this situation coincides with the overall brightness increase. Please acknowledge that a high gain can generate a decrease of image quality.



- *Use lenses with variable slots:* the slot is a hole that allows light passage. The light passes through the lens and hits the surface and acquires the image. Larger the hole, the more light passes and the minor image depth. The focussing of an object with a variable lens distance may result to be difficult.
- *Acquisition time:* the time necessary for the sensor to acquire an image. After the exposure of the CMOS to light for the set exposure time, the image has to be stored in the sensor memory. The transfer of a complete image requires approximately 30 ms. The time can be reduced selecting the acquisition of only a certain part of the image.
- *Inspection time:* the inspection time depends on the operations and the tools used. For example, line tools (e.g. Edge) require a shorter time respect to area tools (e.g. Blob).

## 12.5. Blob analysis

**Blob:** abbreviation for Binary Large Object.

A Blob is a group of adjacent pixels, each with a brightness higher than a specific threshold. Blobs are typically extracted from binary images. Areas with black or white pixels of an image are gathered in connected regions defined blobs.

A blob is recognised examining the image line by line and not each single pixel. At each interval characterised by the same pixel type (white or black), the initial and final coordinates of each line are saved. These objects are formed examining contiguous lines and gathering intervals with the same pixel type. This approach speeds up the operations respect to the algorithm based on each pixel.

The objects examined can have any shape and complexity and can also contain different holes, which are counted as objects.

Different parameters are available to improve performances. Filters that an object has to respect in order to be considered a blob, can be introduced. For example, a threshold can be set concerning the minimum number of pixels an object has to contain, and so excluding small pixels groups due to noise. Objects touching the border of the inspected area can be suppressed. Moreover, a maximum pixel number belonging to a blob can be set to filter the background.

Target-object parameters can be calculated when the necessary information has been gathered: area (pixel number), perimeter ( $X \text{ transitions} + Y \text{ transitions} + \text{diagonal transitions} \times \sqrt{2}$ ), bounding box (Figure 30), moments (necessary to describe asymmetric object orientation), convex perimeter (Figure 31 – approximates the real perimeter nearer to the bounding box).

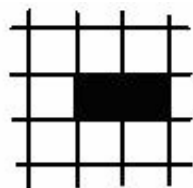


Figure 28

area=2, perimeter =6

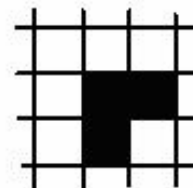


Figure 29

area=3, perimeter=  $6 \cdot \text{SQRT}(2)$

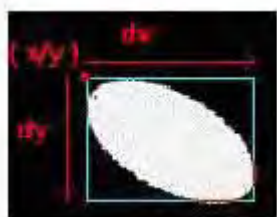


Figure 30

bounding box: rectangle with object height and width

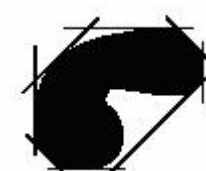


Figure 31



The advantages offered by the *Blob analysis* consist in high elaboration speed and possibility of tolerating and measuring object orientation and dimension variations.

While the constraints consist mainly in the difficulty of distinguishing objects from backgrounds with low contrasts, objects in touch with each other or overlapped objects as well as the difficulty of recognising similar objects.

## 12.6. Contour match

*Contour match* recognised objects comparing the contour characteristics. The analysis is carried-out on binary images with a specific threshold.

The group of black pixels defines the image (BinaryImageSet, BIS).

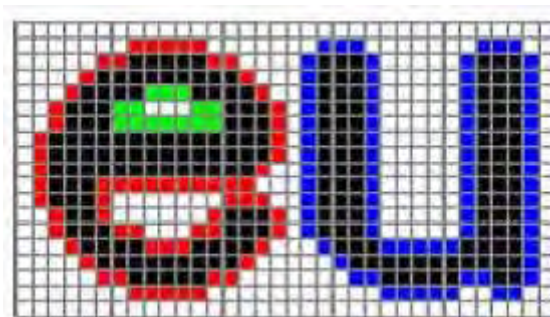


Figure 32

The contour is defined as a pixel sequence that follows the binary image border. This sequence is obtained setting relations between adjacent pixels. The contour does not belong to the BIS. The algorithm can detect both external contours (indicated in red) as well as internal contours (indicated in green).

A template and some relative information (position, pixel number, dimensions, etc.) has to be acquired before recognising a target-object. This information is normalised and entered in a vector and then stored in a database. Object recognition is obtained comparing the target-object's vector and the template's vector previously stored. The distance between the vectors has to be defined to carry-out the comparison. The vector (in the database) closer to the target-object is selected.

This method guarantees a discrete elaboration speed and can tolerate and measure object orientation and dimension variations. Moreover, it offers a higher accuracy respect to *Blob analysis*, but has difficulty in distinguishing objects from backgrounds with low contrasts as well as objects in touch with each other or overlapped objects.

## 12.7. Edge detection

A brief introduction to "projection" is necessary. *Edge detection* functions on 256 grey tone values and not on binary images. Considering that a specific image area can be represented by a matrix of lines and columns. Each matrix element contains a brightness value associated to each pixel. The following example has three lines and ten columns.

69	75	78	85	98	120	128	135	133	135
66	69	76	81	88	102	118	129	134	136
68	74	81	87	101	119	128	136	134	135

The sum of each column value is:

203	218	235	253	287	341	374	400	401	406
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

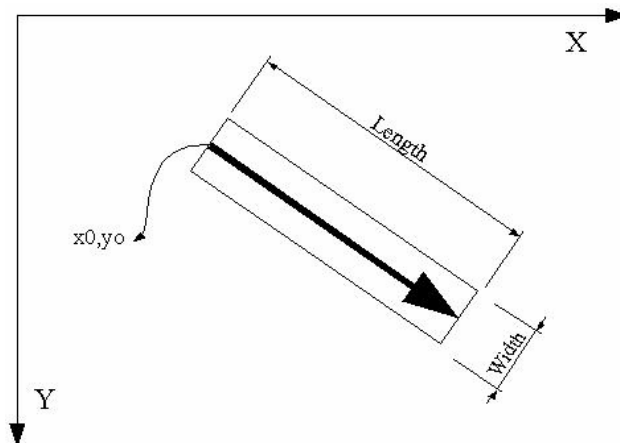


Dividing per three these values and placing them on one single line we will obtain the projection of the area:

68	73	78	84	96	114	125	133	133	135
----	----	----	----	----	-----	-----	-----	-----	-----

The projection consists practically in obtaining an average of the brightness values of an area and listing them on a line. More image information is thus obtained, respect to considering only the pixels and a more accurate analysis can be made.

The area is represented as an arrow (and so determining the projection direction) of a specific length (corresponding to the number of columns) associated to a specific width (number of lines).



When all values are listed along the line, the average intensity flow profile is obtained and the edges can be tracked.

**What is an Edge:** a variation of the brightness between adjacent pixels, that presents higher values respect to a specific threshold.

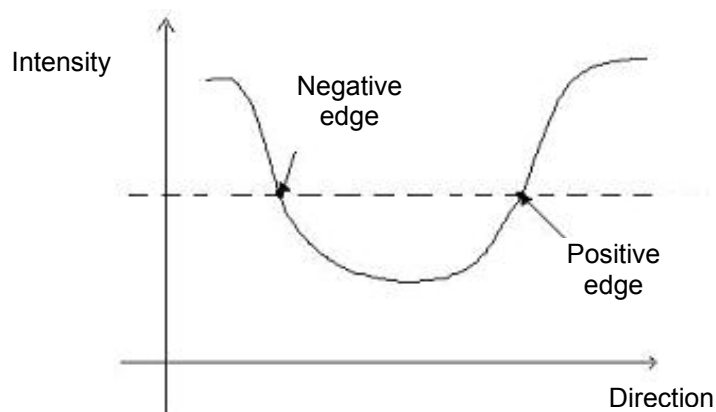
Two different edge types can be found: one due to the transition from a white to a black pixels (negative edge) and one due to the transition from a black to a white pixel (positive edge).

The following paragraphs offer two different criteria to track and edge and determine the position.

#### 12.7.1. Threshold detection

The first point is detected following the intensity profile along a specific direction and the intensity value varies. The point detected will track the edge and its position.

This method is simple and fast but it is very dependent on the lighting used and thus must be avoided in presence of non-constant lighting, while representing an advantage with back light sources.





### 12.7.2. Contrast detection

The detection is not based on the intensity flow but on its derivative. The edge is tracked analysing the brightness difference between two adjacent pixels. The edge is associated to the point where the derivative curve assumes a maximum (positive edge) or a minimum (negative edge) value. The following figures indicate an object and graphs relative to the intensity flow (red) and its derivative (blue).

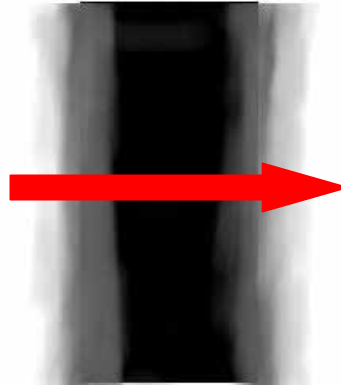
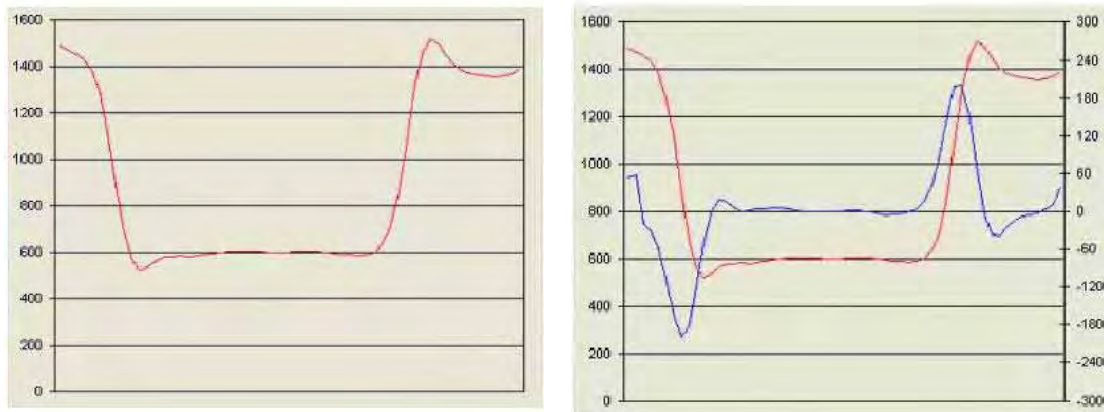


Figure 33



absolute values indicated on the y axis on the left,  
derivative values on the right

The advantage offered by this method is that it is relatively independent from the lighting used. Lighting changes generate an offset on the absolute values maintaining the same contrast. This method is slower and require major memory. It is recommended when the lighting can not be maintained constant.

The threshold values are lower as they are referred only to variation and not to the absolute value.

*Edge detection* is generally used to track the position of an object and make different measurements. It is a very fast and accurate elaboration.

#### Applications:

- 1) Accurate measurement of keys, labels, mechanical parts, etc.
- 2) Object tracking.
- 3) Search of elements inside complex objects.
- 4) Quality control on surfaces (with dust, flaws).
- 5) Position control.



## 12.8. Pattern match

*Pattern match* consists in locating similar patterns between a template and a target-object to inspect. The light characteristics inside a specific area have been saved.

Each template pixel corresponds to a specific brightness value represented by a grey level (0-255).

This value is saved in order to form a numeric matrix corresponding to a specific image region.

During elaboration, a rectangle with dimensions similar to the template passes over the area searching the target-object.

At each rectangle position, the distance between the enclosed target-object and the template is calculated as follows: the brightness difference between pixels in different positions is calculated and the sum of the values provides the difference between the two images.

The image will be considered valid only if the distance results to be inferior to the set threshold.



## 13. APPENDIX B: LENSES

### 13.1. C and CS mounting

The focal length is the only difference between the C and CS mounting. In C mounting the distance is 17.526 mm and 12.5 mm in the CS. The mounting border and opening are the same.

A CS-mount camera can function with CS-mount lenses, but also with C-mount lenses. The C-mount lenses form a longer image (5 mm) respect to CS-mount lenses and so the user just has to add a 5 mm spacer between the camera and lens to achieve complete compatibility.

On the contrary C-mount cameras can function only with C-mount lenses and not with CS-mount lenses.

**The SCS1 device has CS-mounting and thus guarantees maximum flexibility in choosing the lens.**

	C-mount camera	CS-mount camera
C mount lens	OK	OK with adaptor
CS mount lens	NO	OK

**Table 2: different mounting compatibility**

### 13.2. Shutter and focus

The lenses purchased as SCS1's accessory present two rings: one for shutter regulation and the other for focus.

Between the rings there is a white notch that acts as reference point.

The focus ring is characterised by "*near*" text and " $\infty$ " symbol. Turning the ring making closer the reference notch and *near*, the sensor will progressively focus a closer plane. Turning the ring making closer the notch and  $\infty$ , the sensor will progressively focus a further plane.

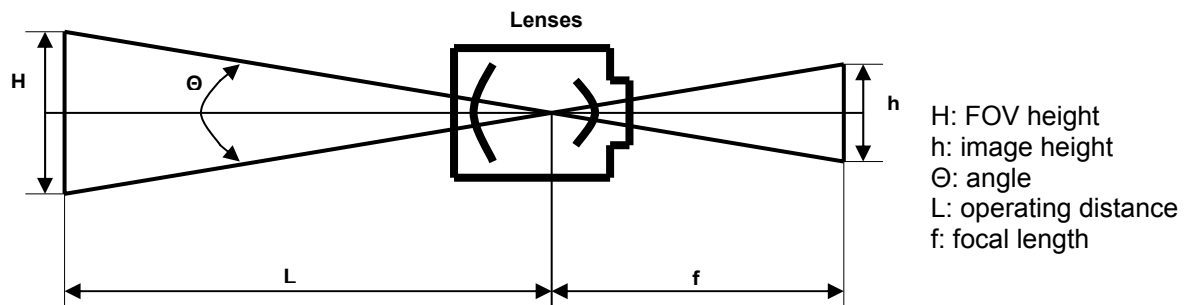
The shutter is characterised by "C" and "O" letters. The shutter opens turning the ring making closer the reference notch and "O" and closes turning making closer the notch and "C". The larger the opening, the more light passes through the lens during acquisition and the image appears brighter (depending on the exposure time) and smaller the depth of field.





### 13.3. Field of view

The field of view (FOV) is the area acquired by the image sensor which forms the image to inspect. The FOV and image sensor dimensions condition directly the resolution.



### 13.4. Operating distance

The operating distance is the distance between the lens' back-side and the object. The operating distance is limited by:

- lenses used have a minimum functioning distance;
- lenses may have to be positioned close to the object to avoid light variations;
- physical sensor positioning constraints may exist.

### 13.5. Focal length

The focal length is the distance between the rear nodal point of the lens (the point where the light beams leave the rear of the lens) and the image sensor.

If more than one focal length is required for the application, the user has to consider the following:

- Cost: a lens with a shorter focal length is less expensive.
- Depth of focus: the depth of focus (focus tolerance) is the area in front of and beyond the best focus point in which the image quality remains acceptable. A longer focal length provides less depth of focus; a shorter focal length provides more depth of focus.
- Distortion: a lens with larger focal length provides a less distorted image.

### 13.6. FOV inspection / resolution

The inspection area has to be measured, considering a margin for object movements, in order to determine the FOV dimension.

To check if the FOV is sufficient, the user can proceed as follows:

- estimate the necessary resolution;
- use the FOV formula to determine the maximum vertical and horizontal FOV dimensions able to
- maintain the necessary resolution.

FOV formula: necessary resolution x 640 pixel = Max horizontal FOV  
 Necessary resolution x 480 pixel = Max vertical FOV

*Example:*

Inspection area: 300 x 350 mm. Necessary resolution: 1 mm

Max. horizontal FOV = 1 mm x 640 = 640 mm  
 Max vertical FOV = 1 mm x 480 = 480 mm

The FOV constraint calculated (640x480) is the largest inspection area and so a 350x300 FOV can be accepted.

If the FOV calculated is inferior to the desired inspected area, the user can proceed as follows:

- use more telecameras;
- inspect a smaller area;
- reduce resolution requirements.



## 13.7. Operating distance table

Operating distance in millimetres							
FOV in mm (height x width)	4 mm lens	8 mm lens	12 mm lens	16 mm lens	25 mm lens	50 mm lens	75 mm lens
5x4	0	0	0	0	0	69	77
10x8	0	0	0	0	0	104	153
15x11	0	0	0	0	52	139	190
20x15	0	0	0	0	69	173	247
25x19	0	0	0	55	87	209	283
30x23	0	0	50	67	104	243	340
35x26	0	39	59	78	121	278	397
40x30	0	45	67	89	139	313	427
45x34	0	50	75	100	156	347	464
50x38	0	55	83	111	173	382	515
60x45	33	67	100	133	209	451	619
70x53	39	78	117	155	243	521	721
80x60	45	89	133	178	278	590	825
90x68	50	100	150	200	313	660	9328
100x75	55	111	167	222	347	729	1031
110x83	61	122	183	245	382	799	1134
120x90	67	133	200	267	417	868	1237
130x98	72	145	217	289	451	937	1340
140x105	78	155	233	311	486	1007	1443
150x113	83	167	250	333	521	1077	1547
160x120	89	178	267	355	555	1146	1649
170x128	95	189	283	378	590	1215	1752
180x135	100	200	300	400	625	1285	1855
190x143	105	211	317	422	660	1354	1959
200x150	111	222	333	445	695	1423	2062
225x169	125	250	375	500	781	1597	2319
250x188	139	278	417	555	868	1771	2577
275x206	153	305	459	611	955	1945	2835
300x225	167	333	500	667	1042	2118	3093
325x244	181	365	542	722	1129	2292	3351
350x263	195	389	583	778	1219	2465	3608
375x281	209	417	625	833	1302	2639	3866
400x300	222	445	667	889	1389	2813	4124
425x319	236	472	709	945	1476	2986	4381
450x338	250	500	750	1000	1563	3160	4639
475x356	264	528	792	1055	1649	3333	4897
500x375	278	555	833	1111	1736	3507	5155
525x394	292	583	875	1167	1823	3681	5413
550x413	239	611	917	1222	1910	3854	5670
575x431	319	639	959	1278	1997	4028	5928
600x450	333	667	1000	1333	2083	4201	6185
625x469	347	695	1042	1389	2170	4375	6443
650x488	361	722	1083	1445	2257	4549	6701

Table 3: Focal length/FOV/Operating distance (in millimetres)



FOV in inches (height x width)	Operating distance in inches						
	4 mm lens	8 mm lens	12 mm lens	16 mm lens	25 mm lens	50 mm lens	75 mm lens
0.2 x 0.2	0.00	0.00	0.00	0.00	0.00	2.73	3.33
0.3 x 0.2	0.00	0.00	0.00	0.00	1.07	3.47	4.67
0.4 x 0.3	0.00	0.00	0.00	0.00	1.40	4.13	5.33
0.5 x 0.4	0.00	0.00	0.00	1.13	1.73	4.87	6.00
1.0 x 0.8	0.00	0.00	1.67	2.20	3.47	8.33	11.33
1.5 x 1.1	0.00	1.67	2.53	3.33	5.20	11.80	17.33
2.0 x 1.5	0.00	2.20	3.33	4.47	6.93	15.33	22.67
2.5 x 1.9	0.00	2.80	4.20	5.53	8.67	18.67	26.67
3.0 x 2.3	0.00	3.33	5.00	6.67	10.40	22.00	30.67
3.5 x 2.6	1.93	3.87	5.87	7.80	12.13	26.00	36.00
4.0 x 3.0	2.20	4.47	6.67	8.87	14.00	29.33	41.33
4.5 x 3.4	2.53	5.00	7.53	10.00	15.33	32.67	46.67
5.0 x 3.8	2.80	5.53	8.33	11.13	17.33	36.00	51.33
5.5 x 4.1	3.07	6.13	9.20	12.20	19.33	39.33	56.67
6.0 x 4.5	3.33	6.67	10.00	13.33	20.67	43.33	62.00
6.5 x 4.9	3.60	7.20	10.87	14.67	22.67	46.67	67.33
7.0 x 5.3	3.87	7.80	11.67	15.33	24.00	50.00	72.00
7.5 x 5.6	4.20	8.33	12.53	16.67	26.00	53.33	77.33
8.0 x 6.0	4.47	8.87	13.33	18.00	28.00	56.67	82.67
8.5 x 6.	4.73	9.47	14.00	18.67	29.33	60.67	80.67
9.0 x 6.8	5.00	10.00	15.33	20.00	31.33	64.00	92.67
9.5 x 7.1	5.27	10.53	16.00	21.33	32.67	67.33	98.00
10.0 x 7.5	5.53	11.13	16.67	22.00	34.67	70.67	103.33
10.5 x 7.9	5.87	11.67	17.33	23.33	36.67	74.00	108.00
11.0 x 8.3	6.13	12.20	18.67	24.67	38.00	78.00	113.33
11.5 x 8.6	6.40	12.80	19.33	25.33	40.00	81.33	118.67
12.0 x 9.0	6.67	13.33	20.00	26.67	42.00	84.67	124.00
12.5 x 9.4	6.93	14.00	20.67	28.00	43.33	88.00	128.67
13.0 x 9.8	7.20	14.67	22.00	28.67	45.33	98.00	134.00
13.5 x 10.1	7.53	15.33	22.67	30.00	46.67	95.33	139.33
14.0 x 10.5	7.80	15.33	23.33	31.33	48.67	98.67	144.00
14.5 x 10.9	8.07	16.00	24.00	32.00	50.67	102.00	149.33
15.0 x 11.3	8.33	16.67	25.33	33.33	52.00	105.33	154.67
15.5 x 11.6	8.60	17.33	26.00	34.67	54.00	109.33	160.00
16.0 x 12.0	8.87	18.00	26.67	35.33	55.33	112.67	164.67
16.5 x 12.4	9.20	18.67	27.33	36.67	57.33	116.00	170.00
17.0 x 12.8	9.47	18.67	28.67	38.00	59.33	119.33	175.33
17.5 x 13.1	9.73	19.33	29.33	38.67	60.67	122.67	180.67
18.0 x 13.5	10.00	20.00	30.00	40.00	62.67	126.67	185.33
18.5 x 13.9	10.27	21.33	30.67	41.33	64.00	130.00	190.67
19.0 x 14.3	10.53	21.33	32.00	42.00	66.00	133.33	196.00
19.5 x 14.6	10.87	22.00	32.67	43.33	68.00	136.67	201.33
20.0 x 15.0	11.13	22.00	33.33	44.67	69.33	140.00	206.00

Table 4: Focal length/FOV/Operating distance (in inches)



## 14. APPENDIX C: ILLUMINATORS

### 14.1. General lighting considerations

- Keep lighting constant.
- Keep lighting consistent (avoid shadows or very bright spots).
- Acquire the shape and form of the target-object with lighting that optimises its contrast but separates it from the background.
- Minimise the variations due to ambient light, sunlight or other ambient light sources.
- Select a lighting technique appropriate for the shape, dimension, colour, reflectivity, transparency and heat tolerance of the target-object.
- Adjust the best distance between target-object and light source.
- Adjust the best sensor exposure time.
- Consider all inspection area constraints.

### 14.2. Ambient light minimisation

To ensure consistent lighting, the amount of ambient light hitting the object has to be minimised. If ambient light is not minimised, external light sources can generate inspection failure. Such failures can be prevented by suppressing ambient light with brighter lights or by shrouding inspection area to block ambient light.

#### **Suppressing ambient light with a brighter, more consistent and more controllable light:**

- *Advantages:* hardware positioning is flexible, as not restricted by light-blocking barriers.
- *Disadvantages:* lighting must be very bright and close to target-object.

#### **Suppressing ambient light with physical barriers:**

- *Advantages:* light brightness is not critical. Lights and camera are protected against physical damages.
- *Disadvantages:* more hardware is required and necessary and inspection area is less accessible.

### 14.3. Lighting options

#### 14.3.1. Ring light

A technique with many application possibilities.

The illuminator is mounted directly on the sensor and illuminates any object in front of it. A ring light provides diffused lighting over a small area.

- *Advantages:* provides correct lighting also for small objects.  
Reduces shadows on images with protrusions. Centres the light on the image.
- *Disadvantages:* with large objects, the corners of the image may lose brightness, creating a halo of black pixels along the external image edge. With highly reflective objects, the image may contain a circular glare of reflected light.

Check of dates, codes on labels or control of label presence are only some of the possible applications.





#### 14.3.2. Top light

This technique generates even light in a concentrated area. The illuminator is positioned behind the sensor and if oriented correctly can carry-out very precise inspections.

- *Advantages:* the light beam is outside the axis respect to sensor optics, allowing the highlight of specific target areas and not others.
- *Disadvantages:* difficulty in lighting evenly the entire object, especially in presence of protruding parts that create unwanted shadows.



#### 14.3.3. Back light

The illuminator is positioned behind the target-object, aimed directly back towards the sensor, and the resulting silhouette can be inspected for dimension and shape control. The brightness is low.

- *Advantages:* does not supply surface characteristics. Facilitates the diameter measurement of round objects. Shows presence or absence of holes.
- *Disadvantages:* illuminator positioning behind object may result difficult. Does not analyse object surfaces. The illumination area must be larger than then inspection area.



Typical applications include object sorting according to shape and dimensions, measurement of spacing between chip leads, inspection of object holes or cracks.



## 15. GLOSSARY

**Acquisition**

The manner in which information is made available to an analysis system, as in image acquisition.

**Algorithm**

A set of rules or procedures used for the solution of problems through the carrying-out of a specific number of steps.

**Ambient light**

Light present in the environment but not generated by the vision system.

**Autoexposure**

Process that allows to automatically adjust the exposition time.

**Backlighting**

Condition where the light reaching the image sensor comes from behind the object to inspect.

**Binarization**

Process that consists in the conversion of a grey-tone image to a two-level image, obtaining an image composed of black and white pixels.

**Blob**

Acronym for Binary Large Object. A Blob è defined as a group of adjacent pixels with brightness higher (or lower) than a specific threshold. Typically blobs are extracted from binary images.

**C-mount /CS-mount**

Two different lens mounting, with the same mounting thread and slots but with different focal lengths. 17.526 mm for C-mount and 12.5 for CS-mount.

**Edge**

Brightness variation between two adjacent pixels with values exceeding a specific threshold.

**Exposure time**

The period of time in which the image-acquiring device is exposed to light. The longer the exposure times, the more light enters into the device and consequently the image is brighter.

**Focal length**

The focal length is the distance between the rear nodal point of the lens (the point where the light beams leave the rear of the lens) and the image sensor.

**Field of View (FOV)**

Visible field focussed by the sensor.

**Greyscale**

Standard grey-tone variations of values from white (255) to black (0).

**Inspection**

A set of operation steps carried-out by the sensor to control the target-objects.

**LED**

Acronym for Light-Emitting Diode.

**Noise**

Irrelevant data that cause disturbance on the data to measure or inspect.

**Operating distance**

The operating distance is the distance between the lens' back-side and the object.



**Pattern matching**

Comparison technique between digital images that considers the brightness value of each single pixel.

**Pixel**

A digital image can be represented by a numeric matrix. Each matrix element is denominated pixel (picture's element). Each pixel can be considered as an information source regarding image greyscale or colour composition.

**Region of Interest (ROI)**

The area analysed by a specific tool during inspection.

**Teach-in**

Process that allows inspection configuration using the sensor's control panel.

**Template**

Image used to for inspection configuration and used as a comparison during elaboration.

**Tolerance**

Established value range upon which to differentiate good and bad products.

**Toolbar**

Panel presenting the push-buttons of the most-frequently used functions of the SCS1 graphic interface.

**Toolbox**

Panel presenting the push-buttons of the available inspection configuration tools of the SCS1 graphic interface

**Tool**

Image elaboration software instrument.

**Trigger**

Mode that defines the image acquisition instant.