Smart-VS™

PRODUCT REFERENCE GUIDE



Smart Vision Sensor

DATALOGIC

Datasensing S.r.l.
Strada S. Caterina 235
41122 Modena
Italy
Tel. +39 059 420411
Fax +39 059 253973

This manual refers to software version 1.3.0 and later.

© 2020-2023 Datasensing S.r.l.

All rights reserved. Without limiting the rights under copyright, no part of this documentation may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means, or for any purpose, without the express written permission of Datasensing S.r.l. Owners of Datasensing products are hereby granted a non-exclusive, revocable license to reproduce and transmit this documentation for the purchaser's own internal business purposes. Purchaser shall not remove or alter any proprietary notices, including copyright notices, contained in this documentation and shall ensure that all notices appear on any reproductions of the documentation. Electronic versions of this document may be downloaded from the Datasensing website (www.datasensing.com).

Disclaimer

Datasensing has taken reasonable measures to provide information in this manual that is complete and accurate, however, Datasensing shall not be liable for technical or editorial errors or omissions contained herein, nor for incidental or consequential damages resulting from the use of this material. Datasensing reserves the right to change any specification at any time without prior notice.

Trademarks

Datasensing and the Datasensing logo are trademarks of Datasensing S.r.l. Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S. and the E.U.

Patents

See www.patents.datasensing.com for patent list.

TABLE OF CONTENTS

PREFACE VI
About this Manual vi
Manual Conventions vi
Technical Support vi
Support Through the Website vi
Warranty vii
COMPLIANCE VIII
General viii
Power Supply viii
EMC Compliance viii
European Declaration of Conformity viii
UKCA Declaration of Conformity viii
FCC Compliance ix
EAC Compliance ix
LED Safety ix
HANDLING X
GENERAL VIEW XII
RAPID CONFIGURATION 1
Connect the System 1
Mount and Position the Device 2
HMI Button Configuration 3
Button Teaching Procedure 3
Button Incremental Teaching Procedure (optional) 4
Button Re-Teaching Procedure (optional) 5
Firmware Rollback and Reset Configurations (optional) 6
Using the Smart-VS WebApp 7
Dashboard Page 8
Current Bank 9
IP Settings 9
Settings Backup 9
Device Information 10
Global settings 10
Hamburger icon 10
Teaching 11
Step 1: Image Setup 11
Step 2: Acquire GOOD 12
Step 3: Acquire NO GOOD 13
Step 4: Learn 13
Monitoring 14
Upper Bar 14
Main area 15
Bottom bar 15
Download the filmstrip 16
I/O Settings 17
INTRODUCTION 18
Product Description 18

Excellent Performance 19
Ease of Setup 19
Ease of Use 19
Flexible Solution 19
Industrial Strength 19
Indicator and Keypad Button 20
Aiming System 21
LED Spots 21
Model Selection and Order Information 22
Internal Lighting System 22
Accessories 22
Application Examples 23
INSTALLATION 24
Package Contents 24
Mechanical Dimensions 25
Mounting And Positioning Smart-VS 27
ELECTRICAL CONNECTIONS 28
Power Supply 29
Inputs 29
Trigger and external push-button connection 30
Outputs 30
Digital I/O diagram 30
Data Valid timing details 31
On-Board Ethernet Interface 32
TYPICAL LAYOUTS 33
Sensor-like Setup 33
WebApp for Initial Configuration and Occasional Monitoring 34
Setup with Telnet Communication for Job Switching 35
READING FEATURES 36
FOV Calculation 36
Detection Diagram 37
MULTI JOB CONFIGURATION 38
Job Switching via Smart-VS WebApp 38
Job Switching via Telnet Communication 39
MAINTENANCE 40
Cleaning 40
TROUBLESHOOTING 41
TECHNICAL FEATURES 43
TCP/IP COMMUNICATION 45
Connection 45
protocol Overview 46
Description of available features 46
Use of synchronous features 46
Use of asynchronous features 47
Delimiters and separator 48
Device control commands 49
Command format 49
Reply format 49
List of available device control functions 50
Return codes 51
BankStatus 51
CRTJB (CReaTeJoB) 52
MDFJB (MoDiFyJoB) 52
EXTJB (EXiTJoB) 53
TRNJB (TRaiNJoB) 53
CNGJB (ChaNGeJoB) 53
BNKST (BaNKSTatus) 54

GTRJB (GeTRunningJoB) 54
GTDVCS (GeTDeViCeStatus) 54
ACQIMG (ACQuireIMaGe) 55
CLRBNK (CLeaRBaNK) 55
CLRJBS (CLeaRJoBS) 55
GTATS (GeTAsyncTaskStatus) 56
FNZJB (FiNaliZeJoB) 56
FNZTRN (FiNaliZeTRaiNing) 57
Example of creating a new job 57
Example of incremental training 57
Binary file upload and download commands 58
Upload command format 58
Upload reply format 58
Download command format 58
Download reply format 59
Device Procedure control Commands 60
Procedure control command format and return codes 60
List of available procedure control functions 60
CRTJBF (CReaTeJoBFile) 61
CRTBCK (CReaTeBaCKup) 61
ULBF (UpLoadBinaryFile) 61
DLBF (DownLoadBinaryFile) 62
STJBF (SToreJoBFile) 62
STBCK (SToreBaCKup) 63
FNZJBF (FiNalaZeJoBFile) 63
FNZBCK (FiNalaZeBaCKup) 64
FNZJST (FiNaliZeJobSTorage) 64
FNZBST (FiNaliZeBackupSTorage) 64
Example Upload of a binary file 65
Scenario 1 65
Scenario 2 66
Example Downloading a binary file 67
Scenario 1 67
Scenario 2 68
Protocol exceptions 68

PREFACE

ABOUT THIS MANUAL

This Product Reference Guide (PRG) is provided for users seeking advanced technical information, including connection, programming, maintenance and specifications. The Quick Reference Guide (QRG) and other publications associated with this product can be downloaded free of charge from the website listed on the back cover of this manual.

Manual Conventions

The following conventions are used in this document:

The symbols listed below are used in this manual to notify the reader of key issues or procedures that must be observed when using the reader:



NOTE: Notes contain information necessary for properly diagnosing, repairing and operating the reader.



CAUTION: This symbol advises you of actions that could damage equipment or property.



WARNING: This symbol advises you of actions that could result in harm or injury to the person performing the task.

TECHNICAL SUPPORT

Support Through the Website

Datasensing provides several services as well as technical support through its website. Log on to (www.datasensing.com).

For quick access, from the home page click on the search icon, and type in the name of the product you're looking for. This allows you access to download Data Sheets, Manuals, Software & Utilities, and Drawings.

WARRANTY

Datasensing warrants that the Products shall be free from defects in materials and workmanship under normal and proper use during the Warranty Period. Products are sold on the basis of specifications applicable at the time of manufacture and Datasensing has no obligation to modify or update Products once sold. The Warranty Period shall be **three years** from the date of shipment by Datasensing, unless otherwise agreed in an applicable writing by Datasensing.

Datasensing will not be liable under the warranty if the Product has been exposed or subjected to any: (1) maintenance, repair, installation, handling, packaging, transportation, storage, operation or use that is improper or otherwise not in compliance with Datasensing's instruction; (2) Product alteration, modification or repair by anyone other than Datasensing or those specifically authorized by Datasensing; (3) accident, contamination, foreign object damage, abuse, neglect or negligence after shipment to Buyer; (4) damage caused by failure of a Datasensing-supplied product not under warranty or by any hardware or software not supplied by Datasensing; (5) any device on which the warranty void seal has been altered, tampered with, or is missing; (6) any defect or damage caused by natural or man-made disaster such as but not limited to fire, water damage, floods, other natural disasters, vandalism or abusive events that would cause internal and external component damage or destruction of the whole unit, consumable items; (7) use of counterfeit or replacement parts that are neither manufactured nor approved by Datasensing for use in Datasensing-manufactured Products; (8) any damage or malfunctioning caused by non-restoring action as for example firmware or software upgrades, software or hardware reconfigurations etc.; (9) loss of data; (10) any consumable or equivalent (e.g. cables, power supply, batteries, etc.); or (11) any device on which the serial number is missing or not recognizable.

THE DATASENSING WARRANTIES ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER WRITTEN, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE. DATASENSING SHALL NOT BE LIABLE FOR ANY DAMAGES SUSTAINED BY BUYER ARISING FROM DELAYS IN THE REPLACEMENT OR REPAIR OF PRODUCTS UNDER THE ABOVE. THE REMEDY SET FORTH IN THE WARRANTY STATEMENT IS THE BUYER'S SOLE AND EXCLUSIVE REMEDY FOR WARRANTY CLAIMS. NO EXTENSION OF THIS WARRANTY WILL BE BINDING UPON DATASENSING UNLESS SET FORTH IN WRITING AND SIGNED BY DATASENSING'S AUTHORIZED REPRESENTATIVE. DATASENSING'S LIABILITY FOR DAMAGES ON ACCOUNT OF A CLAIMED DEFECT IN ANY PRODUCT DELIVERED BY DATASENSING SHALL IN NO EVENT EXCEED THE PURCHASE PRICE OF THE PRODUCT ON WHICH THE CLAIM IS BASED. DATASENSING SHALL NOT BE LIABLE FOR DAMAGES RELATING TO ANY INSTRUMENT, EQUIPMENT, OR APPARATUS WITH WHICH THE PRODUCT SOLD UNDER THIS AGREEMENT IS USED. Further details on warranty coverage, rights and conditions are addressed under and regulated by the Terms and Conditions of Sales of Datasensing available at https://www.datasensing.com/terms_conditions_sales.

COMPLIANCE

GENERAL

For installation, use and maintenance do not open the sensor.

Only connect Ethernet connections to a network which has routing only within the plant or building and no routing outside the plant or building

Power Supply

ATTENTION: READ THIS INFORMATION BEFORE INSTALLING THE PRODUCT

This product is intended to be installed by **Qualified Personnel** only.

This product is intended to be connected to a UL Listed Direct Plug-in Power Unit (rated 10 to 30 V, minimum 1 A) marked LPS or “Class 2”.

EMC Compliance

In order to meet the EMC requirements:

- connect sensor chassis to the plant earth ground by means of a flat copper braid shorter than 100 mm;
- for direct connections, connect your cable shield to the locking ring nut of the connector.

European Declaration of Conformity

Hereby, Datasensing S.r.l. declares that the full text of the European Declaration of Conformity is available at: www.datasensing.com. Select the link from the downloads section of the product page.

UKCA Declaration of Conformity

Hereby, Datasensing S.r.l. declares that the full text of the UKCA Declaration of Conformity is available at: www.datasensing.com. Select the link from the downloads section of the product page.

Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

FCC Compliance

Modifications or changes to this equipment without the expressed written approval of Datalogic could void the authority to use the equipment.

This device complies with PART 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference which may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his/her own expense.

EAC Compliance

Customs Union:

The CU Conformity certification has been achieved; this allows the Product to bear the Eurasian Mark of conformity.

LED Safety

LED emission according to EN 62471.

HANDLING

The Smart-VS is designed to be used in an industrial environment and is built to withstand vibration and shock when correctly installed, however it is also a precision product and therefore before and during installation it must be handled correctly to avoid damage.

- Avoid that the device is dropped (exceeding shock limits).



- Do not fine tune the positioning by striking the device or bracket.



- Do not weld the device into position. It can cause electrostatic, heat or reading window damage.



- Do not spray paint near the sensor. It can cause reading window damage.



GENERAL VIEW



Figure 1 - General View

1. Bracket Mounting Holes (2)
2. Power On LED
3. Ethernet Connection LED
4. 90° Rotating Connector Block
5. HMI Interface
6. Ethernet Connector
7. Power - I/O Connector
8. Lens
9. LED Aiming System
10. Red Spot (NO GOOD)
11. Green Spot (GOOD)
12. White Polarized Illuminators



Figure 2 - HMI Interface Details

HMI CONFIGURATION	
	**NO GOOD object** • in Teach phase: blinking, NO GOOD object teaching • in Run phase: NO GOOD object detected
	• for future use
	**Trigger** • in Teach phase: trigger input status •in Run phase: trigger received
	**GOOD object** • in Teach phase: blinking, GOOD object teaching • in Run phase: GOOD object detected
	**Run** • steady: device in Run phase • blinking: Teaching required

CHAPTER 1
RAPID CONFIGURATION

CONNECT THE SYSTEM

Figure 3 shows an example illustrating how to connect the system. In this layout the result signals are connected to the PLC while it is possible to configure and monitor the device by the Host PC.

The device is activated by an External Trigger (photoelectric sensor) when the object enters its reading zone.

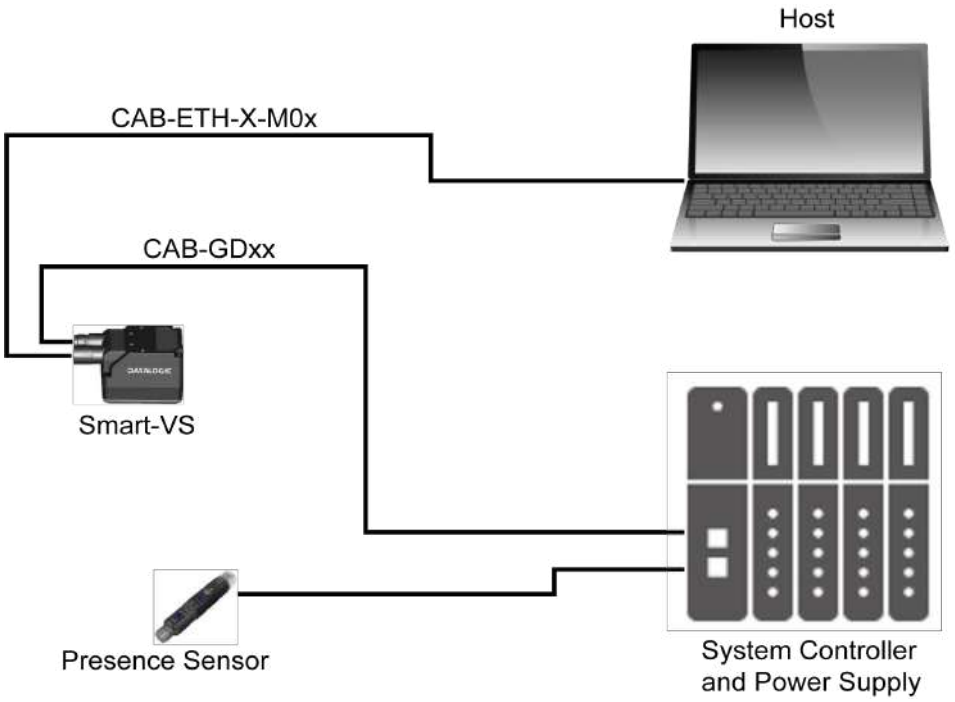


Figure 3 - Smart-VS layout example

MOUNT AND POSITION THE DEVICE

1. To mount the Smart-VS, use the mounting bracket to obtain the most suitable position for the device. The most common mounting configuration is shown in the figure below.

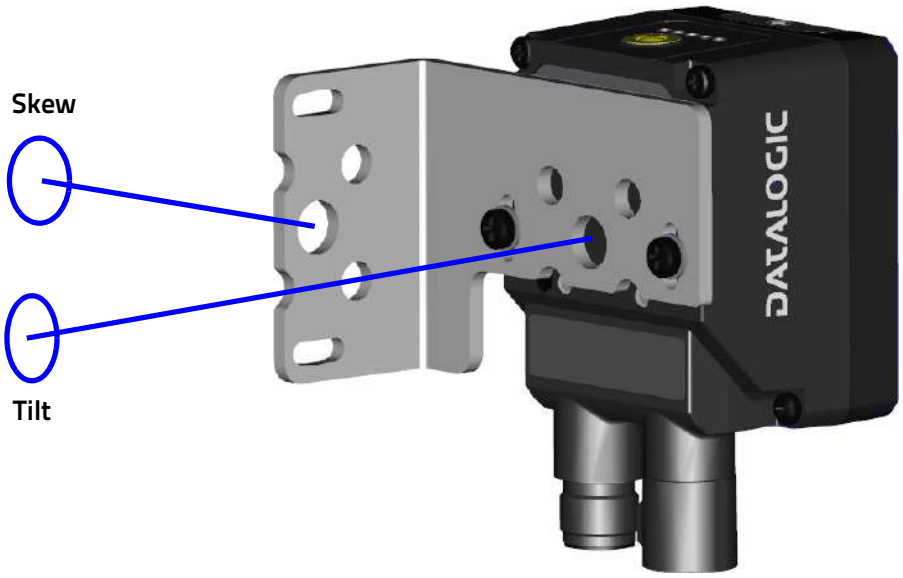


Figure 4 - Positioning with Mounting Bracket

2. When mounting the Smart-VS take into consideration these three ideal label position angles: **Pitch or Skew 10° to 20° and Tilt 0°**, although the device can operate at any tilt and skew angle provided the object feature to be inspected fits into the Field Of View (FOV).



Figure 5 - Max. 20% pitch to avoid direct reflections

NOTE: Rapid Configuration of the Smart-VS sensor can be made either through the HMI interface (see "Button Teaching Procedure" on page 3**) which requires no PC connection, or by using the Smart-VS WebApp (see** "Using the Smart-VS WebApp" on page 7**). Select the procedure according to your needs.**

HMI BUTTON CONFIGURATION

Button Teaching Procedure

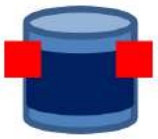
The Run LED will blink until the Teaching procedure is entered (e.g. device factory default).

Long press (> 4s, until the red LED on HMI lights up) the HMI Button to enter the Teaching procedure.

1. GOOD objects required to be taught (green LED and green spot blink).



2. Place the GOOD object in front of the Aiming System.



3. Place the trigger sensor properly. The Trigger LED indicates object detection.



4. Short press (< 1s) the HMI Button to acquire the image. More than one GOOD object can be acquired. It is suggested to acquire one image per GOOD object instance. Camera parameters are auto-adjusted on the first acquisition only.





5. Long press (> 4s, until the red LED on HMI lights up) the HMI Button to start acquiring NO GOOD objects. The red LED and the red spot start blinking.





6. Place the NO GOOD object in front of the Aiming System. Check the Trigger LED.

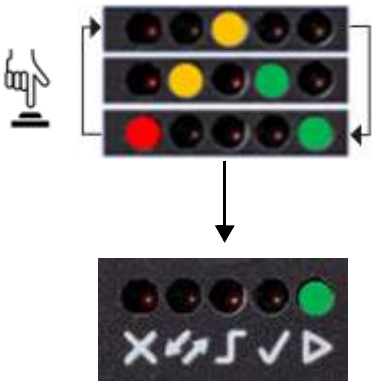




7. Short press (< 1s) the HMI Button to acquire the image. More than one NO GOOD object can be acquired.



8. Long press (> 4s, until the red LED on HMI lights up) the HMI Button to enter the automatic learning procedure (LEDs pattern). In the end, the device will enter the Run Phase



NOTE: The Teaching procedure can also be entered when in Run phase: press the HMI Button for 2s (yellow LED on HMI) for incremental teaching or 4s (red LED on HMI) to cancel and re-teach.

Button Incremental Teaching Procedure (optional)

When in Run mode (Run green LED steady on), training images can be added to the current job.

NOTE: It is recommended to perform this procedure via WebApp to have more control over this operation.

1. Press the HMI Button for approx. 2 seconds and release it as soon as the HMI yellow LED turns on.



2. New GOOD object images can be added to the current job.

   If incremental teaching is needed for NO GOOD objects only, skip to step 3.



   Short press (< 1s) the HMI Button to acquire a GOOD image.



3. Long press (> 4s, until the red LED on HMI lights up) the HMI Button to start acquiring NO GOOD objects. Red LED and red spot start blinking.



4. New NO GOOD object images can be added to the current job.

   If incremental teaching is needed for GOOD objects only, skip to step 5.

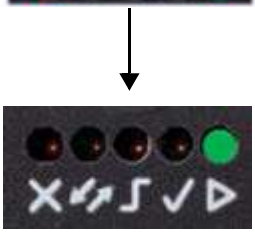
   Short press (< 1s) the HMI Button to acquire the NO GOOD image.





5. Long press (> 4s, until the red LED on HMI lights up) the HMI Button to enter the automatic learning procedure (LEDs pattern). In the end, the device will enter the Run phase.





NOTE: In case of mistakes, the procedure can be aborted before proceeding to step 5 by pressing the HMI Button for approx. 2 seconds until the HMI yellow LED turns on.



Button Re-Teaching Procedure (optional)

When in Run mode (Run green LED steady on), the current job can be erased and re-trained.

Long-press (> 4s, until the red LED on HMI lights up) the HMI Button to enter the Teaching procedure, then follow the "Button Teaching Procedure" on page 3.

If errors occur and the new Teaching is not finalized, the current job will not be overwritten.

NOTE: In case of mistakes, the procedure can be aborted before entering the learning procedure by pressing the HMI Button for approx. 2 seconds until the HMI yellow LED turns on.



Firmware Rollback and Reset Configurations (optional)

It is possible to perform a firmware Rollback to the second to last software version installed or a complete Reset Configurations.



NOTE: It is suggested to close the Smart-VS WebApp before performing the procedures above. Connect to or refresh the WebApp to check the device software version (see "" on page 17).

In order to perform the Rollback or Reset procedure, follow the next steps:

1. Disconnect the device power supply.
2. Press and hold down the HMI button.
3. Power up the device keeping the HMI button pressed until:
   - for firmware **Rollback**, hold down until all LEDs blink simultaneously, then release it when they are steady on (within 4 seconds max).



NOTE: All saved jobs are kept with a Firmware Rollback. Two consecutive Firmware Rollback procedures lead to the factory software version keeping all the saved jobs.

   - for the **Reset** configurations procedure, hold down until all LEDs blink simultaneously, then release and press it again (within 1 second).



NOTE: The Reset Configurations procedure restores the factory default configurations (global settings and IP address 192.168.3.100) and erases all saved jobs, while keeping the last software version installed on your device.

USING THE SMART-VS WEBAPP

Before using a Smart-VS device, a Teaching procedure must be performed. Teaching can be completed either using the embedded HMI (refer to the Smart-VS Quick Reference Guide) or the Smart-VS WebApp graphic user interface.



NOTE: To access the Smart-VS WebApp, Google Chrome is the recommended Internet browser.

To access the Smart-VS WebApp, connect to the device IP address (factory default: 192.168.3.100) via Ethernet.

For systems that support the Link-Local Multicast Name Resolution (LLMNR) protocol, you can connect to the Smart-VS by typing **smart-vs/** in the address bar (if there is only one device on the network) or **smart-vs-[Serial Number]/** (e.g. smart-vs-c12345p/) if there are multiple devices on the same network.



NOTE: In case of a new device, the user is automatically redirected to the Dashboard Page (see "Dashboard Page" on page 8**).**



NOTE: In case of an already trained device, the Monitoring page is displayed (see "Monitoring" on page 14**).**



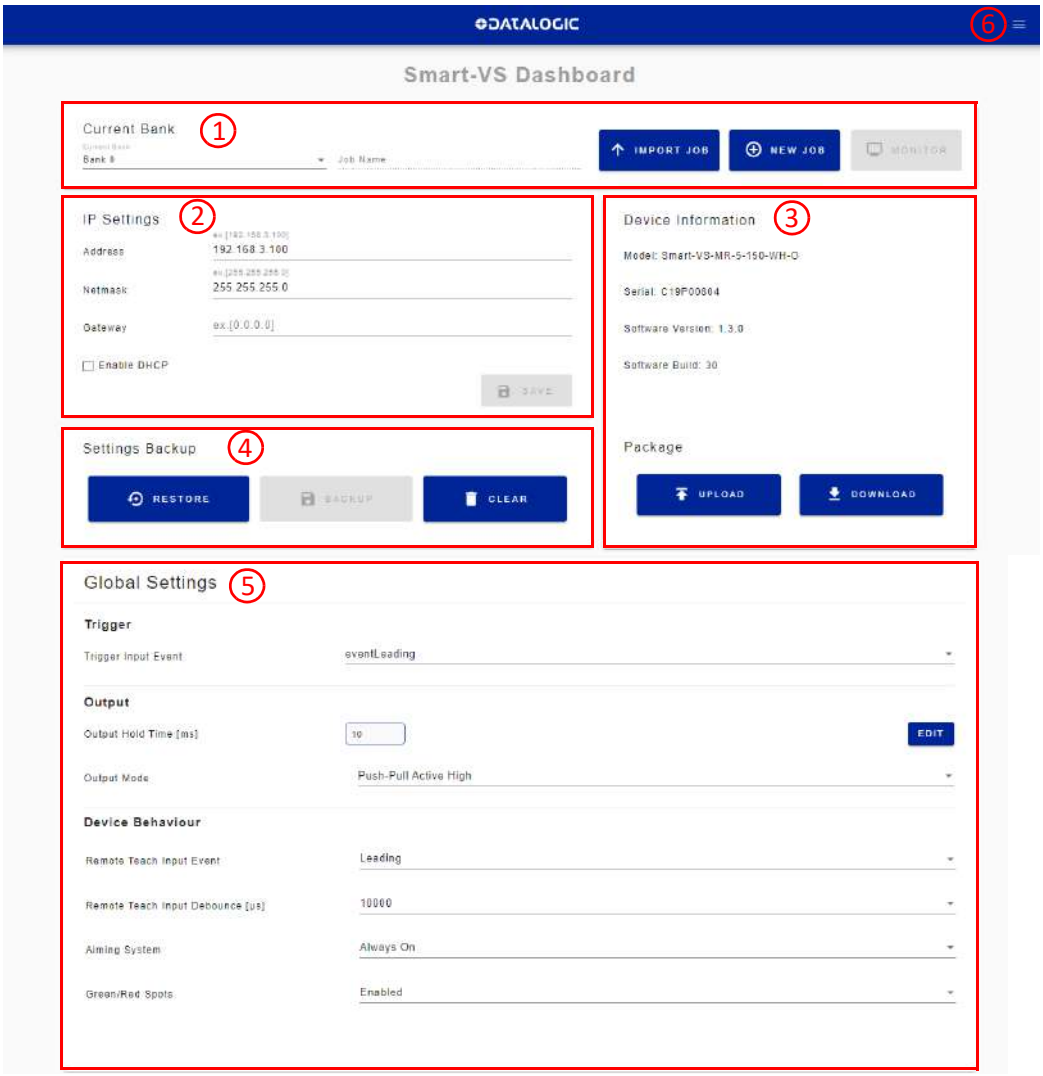
NOTE: A I/O Settings page is also available to adjust the device settings (see "I/O Settings" on page 17**).**



NOTE: Device information and the software update feature are available on the Dashboard Page**.**

DASHBOARD PAGE

When connecting a new device, the “Smart-VS Dashboard” page is displayed showing:



AREA	DESCRIPTION
1	**Current Bank** (see "Current Bank" on page 9)
2	**IP Settings** (see "IP Settings" on page 9)
3	**Settings Backup** (see "Settings Backup" on page 9)
4	**Device Information** (see "Device Information" on page 10)
5	**Global Settings** (see "Global settings" on page 10)
6	**Hamburger icon** (see "Hamburger icon" on page 10)

Current Bank

Allows to change the bank in use.

Import Job

To import a job previously downloaded on PC from another Smart-VS.

New Job

To create a new job through the Teaching wizard. Refer to "Teaching" on page 11.

Monitor

Allows to go to the monitoring page if the selected bank is configured.

IP Settings

Under IP Settings, the Address and Netmask fields can be edited to configure any static IP address, while the Gateway field is optional.

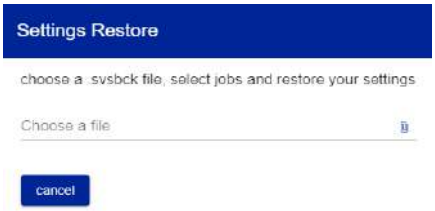
Furthermore, by enabling DHCP, the device automatically acquires the IP address. There must be a DHCP server in the network. If there is no DHCP server, the device will assign itself an IP address in the local link range (169.254.0.0/16).

Settings Backup

When a Settings Backup of a device has been completed, this button allows importing all jobs and global settings of that device.

Restore

Opens the following window:



After selecting the backup file (`.svsbck` format), select one or more job you want to restore and confirm. The global settings will be restored automatically. At the end of the procedure, the Monitoring page is displayed.

Backup

Only available if there is at least one job on the device. Allows to save all the jobs on the device in a `.svsbck` file.

Device Information

Show device information (model, serial number, software versions).

Upload

To upload a firmware package.

Download

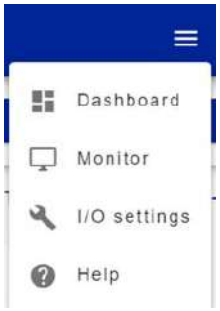
To download the device firmware package.

CAUTION: Do NOT perform the Settings Restore and Upload Package operations while the device is receiving trigger signals.

Global settings

Refer to "I/O Settings" on page 17.

Hamburger icon



AREA	DESCRIPTION
Dashboard	Opens the Dashboard page
Monitor	Opens the Monitor page
I/O Settings	Opens the I/O Settings page
Help	Opens the Smart-VS WebApp User's guide

TEACHING

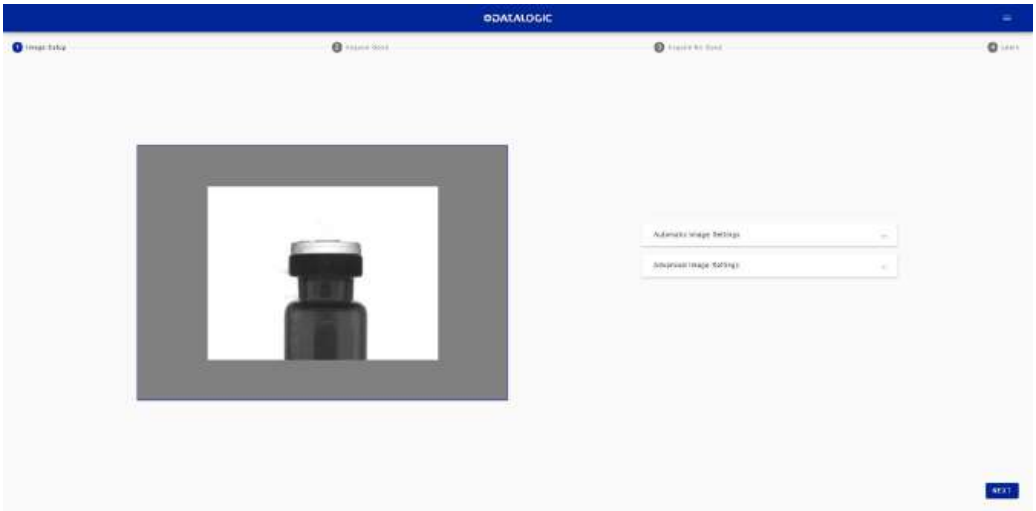
To perform Teaching on your Smart-VS device, enter a job name, select the bank where to store it, click on the Create Job button, and follow the procedure described below.



Step 1: Image Setup

On the **Image Setup** page, select *Start Automatic Setup* to automatically set the Focus Distance, the Exposure Time, and the Sensor Gain parameters (suggested). Alternatively, select *Advanced Image Settings* to manually set the parameters.

It is also possible to move the Field of View based on the application needs, which can be useful when objects of different batch formats must be inspected on the same line. To do this, simply click on the image and drag it throughout the gray area.



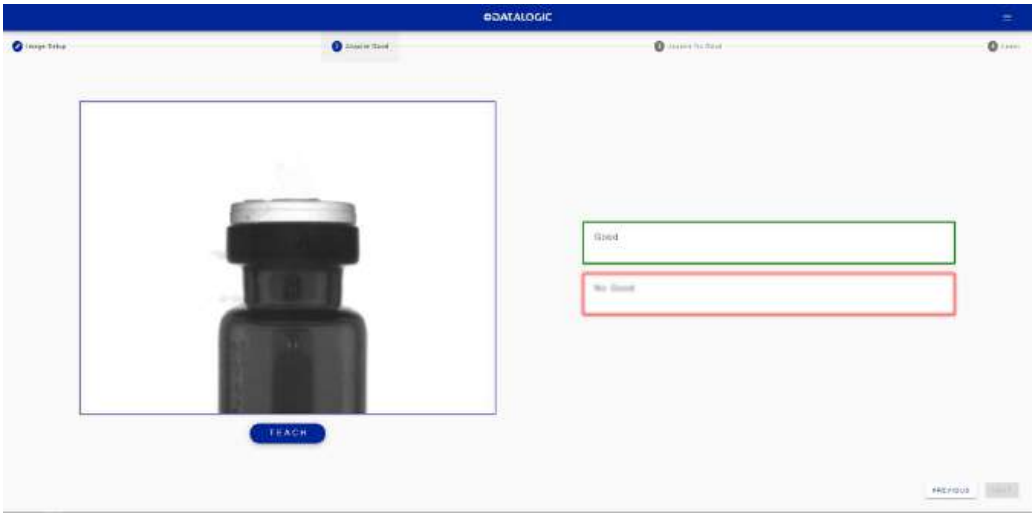
After completing image setup, click *NEXT*.

Step 2: Acquire GOOD

On the **Acquire GOOD** page, one or more GOOD objects can be acquired by clicking on the *Teach* button.

It is recommended to acquire as many GOOD images as the known number of instances to be treated as GOOD.

NOTE: The GOOD and NO GOOD boxes can store a total of 6 images, of which at least one per box. This means that each box can have a maximum of 5 images.



NOTE: Click on an acquired object in the GOOD box and select DELETE to delete it.

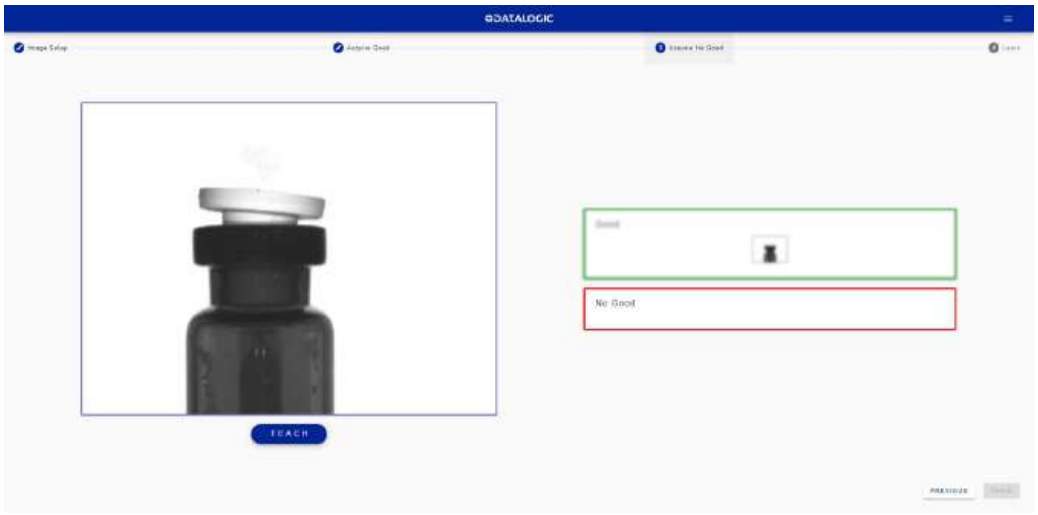
After adding at least one object, you can click *Next* and go to the Acquire NO GOOD page.

Step 3: Acquire NO GOOD

On the **Acquire NO GOOD** page, one or more NO GOOD objects can be acquired by clicking on the *Teach* button.

It is recommended to acquire as many NO GOOD images as the known number of instances to be treated as NO GOOD.

NOTE: The GOOD and NO GOOD boxes can store a total of 6 images, of which at least one per box. This means that each box can have a maximum of 5 images.

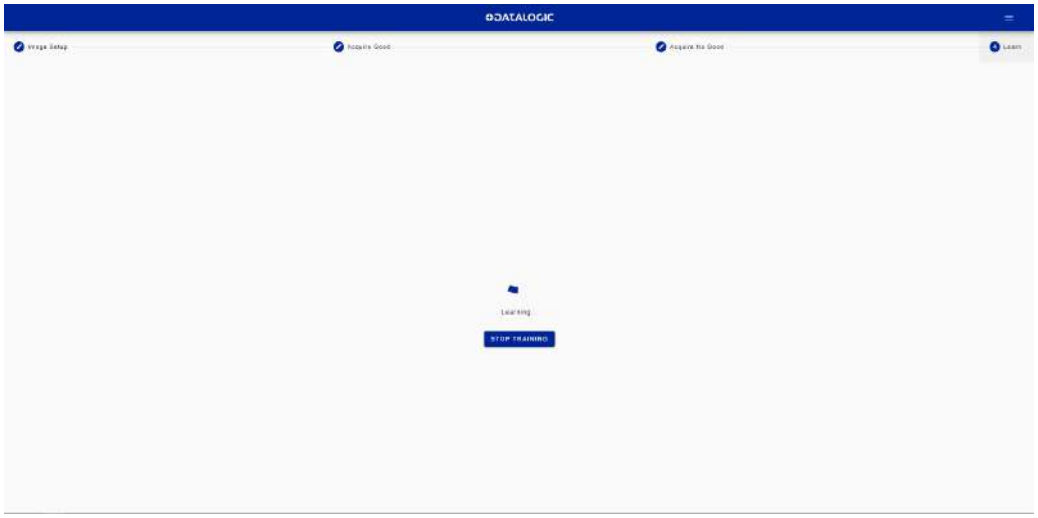


NOTE: Click on an acquired object in the NO GOOD box and select DELETE to delete it.

After adding at least one NO GOOD object, you can click on the *Teach* button.

Step 4: Learn

This step is completed automatically. If it takes too long, you can stop the procedure clicking on the *Stop training* button.



Once the device is trained, the Smart-VS WebApp switches to the Monitoring page.

MONITORING

The **Monitoring** page is divided into an upper bar, a main area, and a bottom bar.



AREA	DESCRIPTION
1	**Upper bar** (see "Upper Bar" on page 14)
2	**Main area** (see "Main area" on page 15)
3	**Bottom bar** (see "Bottom bar" on page 15)

Upper Bar

The **upper bar** contains the job name to the left. Use the drop-down menu to switch to another job (if present).

NOTE: If you want the selected job to be the running and startup job, click on the *Play* button before leaving the page.

Next to the job name, the following icons are present:

ICON	DESCRIPTION
	**Upload job**: uploads a job from your PC (`.svscfg` file)
	**Download job**: downloads the current job on your PC
	**Add new job**: switches to the Training wizard to create a new job
	**Delete job**: deletes current job
	**Save Widget:** save current configuration of the widget
	**Add Widget:** open a new windows to select the widget to be added and where

Main area

The **main area** includes:

- the statistics box to the left. This shows the statistics concerning the ongoing acquisitions. These are active by default, but can be changed or removed;
- the cropping area at the center;
- empty area at the right available for custom widget.

Bottom bar

The **bottom bar** displays an archive of acquisitions (filmstrip).



NOTE: When the device receives no trigger event, the Monitoring page only displays the statistics box to the right and a blank filmstrip.

To start acquiring objects, the following options are available:

ICON	DESCRIPTION
	Save and Play with External Trigger
	Play with Self-Trigger
	Single shot
	Show all images or just one category between GOOD, NO GOOD, and OVERRUN
	Download the filmstrip (refer to "Download the filmstrip" on page 16)

The *Play* button is also a *Save* button: click on it before leaving the page to store any change.



NOTE: The *Play with Self-Trigger* button is not an actual operating mode (output signals are not driven). It is only intended for demonstration purposes or preliminary checks in static condition.

Next to the *Play* icons, the acquisition results are displayed as green bars for GOOD objects, red bars for NO GOOD objects, and gray bars for OVERRUN[1] occurrences.

Click on the *Pause* icon to stop acquisition and enable all available functions.

Dragging the gray box along the bottom bar displays the acquisitions included in that interval. You can now perform **Incremental Training**: click on an acquired image and select GOOD or NO GOOD to add it to the corresponding acquisition box. It is also possible to drag and drop acquired images to the GOOD / NO GOOD box or from the GOOD box to the NO GOOD box and vice versa.

After clicking on one of the *Play* icons, the newly added acquisitions will be used by the device in addition to those stored during the initial Teaching procedure.

1. Overrun occurrences are mostly due to a noisy trigger signal and/or a too fast sequence of trigger events (e.g. more than two events every 50 ms). Choose a proper Debounce filter time and/or adjust the line speed to have 20 pieces per second.

Furthermore, the Smart-VS WebApp allows image saving. Any image shown in the WebApp can be saved clicking on it and selecting *Save*.



NOTE: It is recommended to enable the browser option “Ask where to save each file before downloading”.

Download the filmstrip

The download button allows to download the filmstripe from the archive.

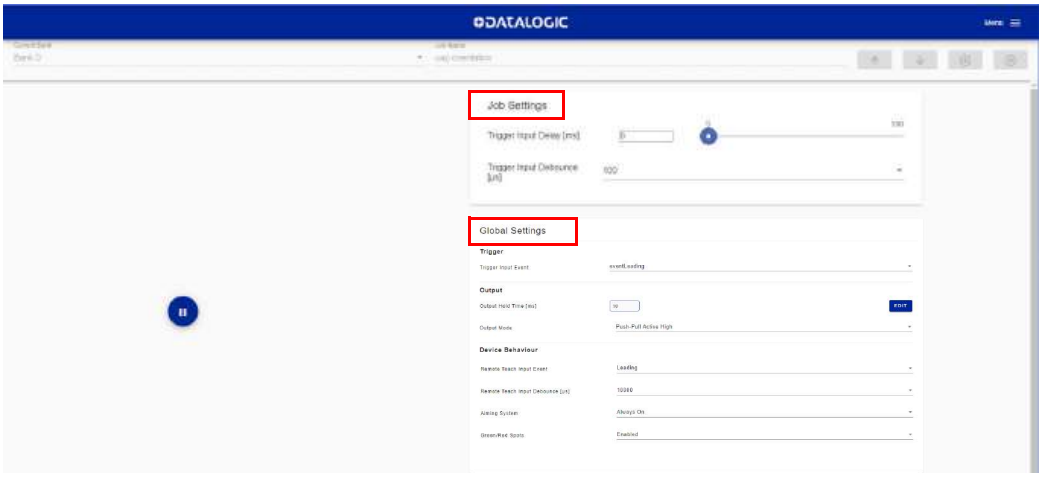


NOTE: Selecting the option “Download the entire filmstrip” downloads the last 50 acquisitions.

I/O SETTINGS

This page contains Job Settings and Global Settings.

It can be reached using the hamburger icon on the upper right corner.



The parameters under Job Settings are variable for each job, which means that any change will only be valid for the current job:

- **Trigger Input Delay**: the delay to apply prior to the acquisition process of the trigger event. It is measured in time (ms). This parameter can be adjusted along with image acquisition: images are shown on the related window.
- **Trigger Input Debounce**: filter debounce time measured in µs.

The parameters under Global Settings are common to all jobs, which means that any change will be valid for all jobs:

- **Trigger Input Event**: can be either Leading or Trailing.
- **Output Hold Time**: the time (ms) during which the output data remains valid.
- **Output Mode**: available selections are NPN, PNP, Push-Pull active High, Push-Pull active Low.
- **Remote Teach Input Event**: available selections are Leading and Trailing.
- **Remote Teach Input Debounce**: filter debounce time measured in µs.
- **Aiming System**: available selections are Always ON, Always OFF, Calibration Only.
- **Green/Red Spots**: available selections are Calibration Only, Enabled, Disabled.

CHAPTER 2
INTRODUCTION

PRODUCT DESCRIPTION

Smart-VS is a sensor with learning ability and easy to set, suitable for any presence and orientation object detection.

Smart-VS can solve all the applications where a very simple installing procedure is requested without the need of particularly smart camera features.

Thanks to its algorithm technology, Smart-VS is the ideal solution for application problems related to machine format adaptability and flexibility that a sensor or sensor system cannot solve as easily.

Smart-VS has been developed for different uses in the secondary packaging and automotive industries, food and beverage applications for bottling, cosmetic packaging, and all those markets where the following detection applications are needed:

Label presence



Cap orientation



Cap presence



Printing on label



Excellent Performance

- Electronic focus control
- 50 ... 150 mm operating distance
- Image Cropping for shorter acquisition time
- Algorithm for fast and simple application setup

Ease of Setup

- Quick installation by using a user-friendly, sensor-like HMI for easy and intuitive setup
- LED pointers for aiming
- Fast mounting accessory
- Visual Feedback to verify exact code positioning in the Field of View and to maximize performance
- Easy-to-use Web Server GUI for maintenance and job setting
- User-defined database of Image Acquisitions

Ease of Use

- HMI interface LEDs provide operational and performance feedback
- Multi Image Acquisition Settings for higher flexibility
- Image saving and storage with buffering capability
- Diagnostic software tools

Flexible Solution

- Complete set of accessory mounting brackets and cables
- Ethernet point-to-point communication available
- Opto-coupled, polarity insensitive inputs
- PP, NPN or PNP short-circuit protected outputs

Industrial Strength

- Industrial compact visual sensor
- Rugged metal construction
- Sealed circular M12 connectors
- IP65 and IP67 protection class
- 50 °C max operating temperature
- Supply voltage ranges from 10 to 30 Vdc

INDICATOR AND KEYPAD BUTTON



Figure 6 - Indicators

The following LED indicators are located on the device:

PWR	blue LED indicates that the device is connected to the power supply (Figure 6, 1)
NET	yellow LED indicates connection to the on-board Ethernet network (Figure 6, 2)

The colors and meaning of the five LEDs are illustrated in the following table:

	**NO GOOD**	red LED indicates a NO GOOD object. Blinking during Teaching (Figure 6, 3)
		for future use (Figure 6, 4)
	**TRIGGER**	yellow LED indicates trigger input status in Teach phase, and trigger received in Run phase (Figure 6, 5)
	**GOOD**	green LED indicates a GOOD object. Blinking during Teaching (Figure 6, 6)
	**RUN**	green LED indicates that the device is in Run phase. If blinking, it indicates that Teaching is required (Figure 6, 7)

During device startup (reset or restart phase), these five LEDs blink for one second.

AIMING SYSTEM

There are two red LED pointers that project two equidistant square patterns onto the target area. The object feature to be detected should be centered between the two squares.



Figure 7 - Smart-VS Aiming LEDs

LED SPOTS

There are two LED spots that can be activated to project light onto the target area to indicate that a GOOD or NO GOOD object has been detected.

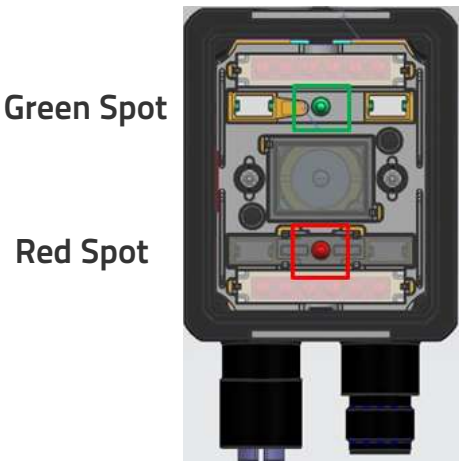


Figure 8 - Smart-VS Good / No Good LED Spots

MODEL SELECTION AND ORDER INFORMATION

Model	Smart-VS-MR-5-150-WH-O
Description	SVS WP 150mm OUT
Optics	7 mm
Illuminator	White polarized
I/O	3 OUT + 2 IN + ETH
Order number	959971320

INTERNAL LIGHTING SYSTEM

The Smart-VS illuminator is composed of 4 white LEDs with polarized filter.



Figure 9 - Smart-VS illuminator

ACCESSORIES

The following accessories can be used with Smart-VS.

Accessory	Order No.
Cables	
CAB-GD03 M12-17P 3M power and I/O cable, isolated wires	95A900052
CAB-GD05 M12-17P 5M power and I/O cable, isolated wires	95A900053
M12-IP67 GIGA Ethernet Cable X-Coded (1M)	93A050122
M12-IP67 GIGA Ethernet Cable X-Coded (3M)	93A050123
M12-IP67 GIGA Ethernet Cable X-Coded (5M)	93A050124
Adapter Cable GIGA Ethernet X-Coded M12 to RJ45	93A050128
Adapter Cable GIGA Ethernet X-Coded to Ethernet D-Coded	93A050129
Bracket	
BK-22-000 Fixing Bracket M220 Body	93ACC0230

APPLICATION EXAMPLES

Smart-VS has been developed for different uses in the secondary packaging and automotive industries, food and beverage applications for bottling, cosmetic packaging, and all those markets where the following detection applications are needed:

Application		Solved cases (OK / NOT OK)	
Label presence			
Cap orientation			
Cap presence			
Printing on label			

CHAPTER 3
INSTALLATION

PACKAGE CONTENTS

Verify that the Smart-VS device and all the parts supplied with the equipment are present and intact when opening the packaging; the list of parts includes:

- Smart-VS (w/connector plug/cover)
- Quick Reference Guide
- Mounting Kit
  - Mounting Screws (2)
  - Washers (2)
  - Mounting Bracket

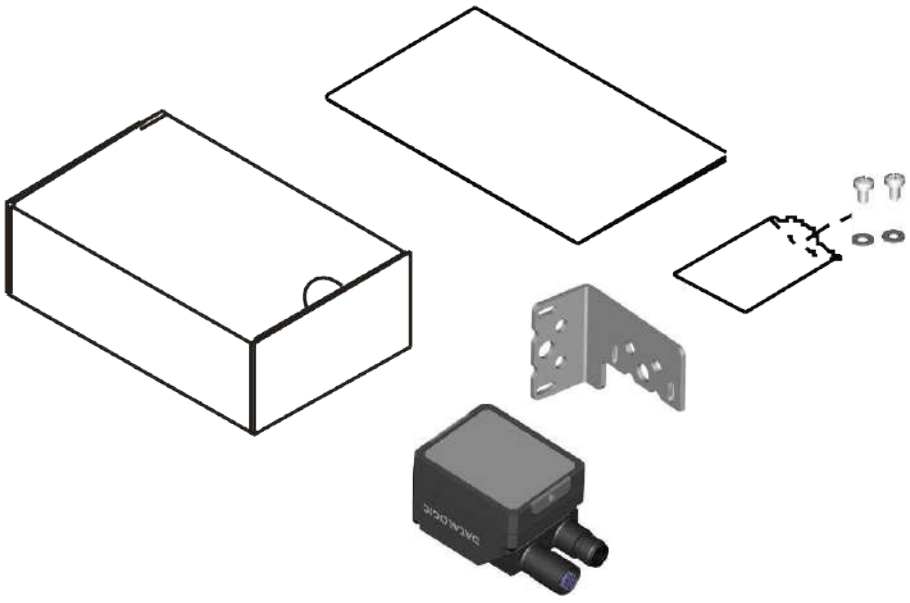


Figure 10 - Package Contents

The Ethernet sealing connector plug provides IP67 protection when Ethernet is not used.

MECHANICAL DIMENSIONS

Smart-VS can be installed to operate in different positions. The two screw holes (M3 x 4mm depth) on the body of the device are for mechanical fixture.

The diagrams below give the overall dimensions of the device and may be used for its installation.

Refer to page 27 for various mounting solutions and correct positioning.

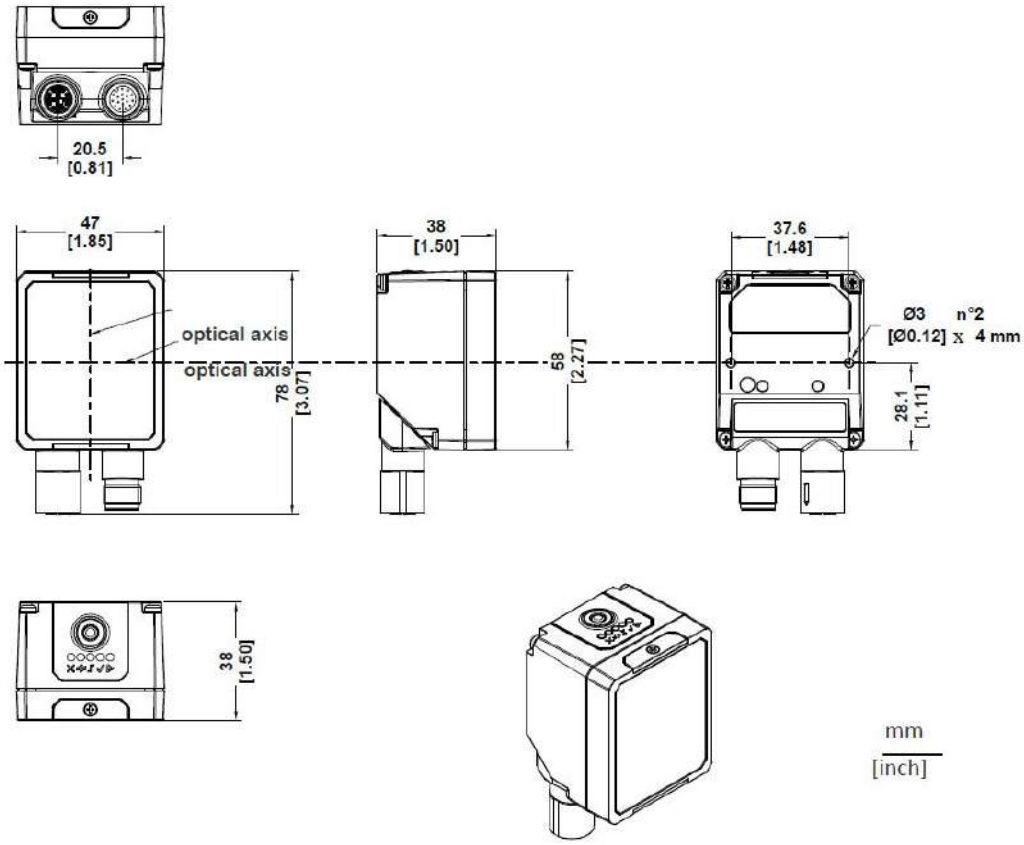


Figure 11 - Overall Dimensions; Connector at 0°

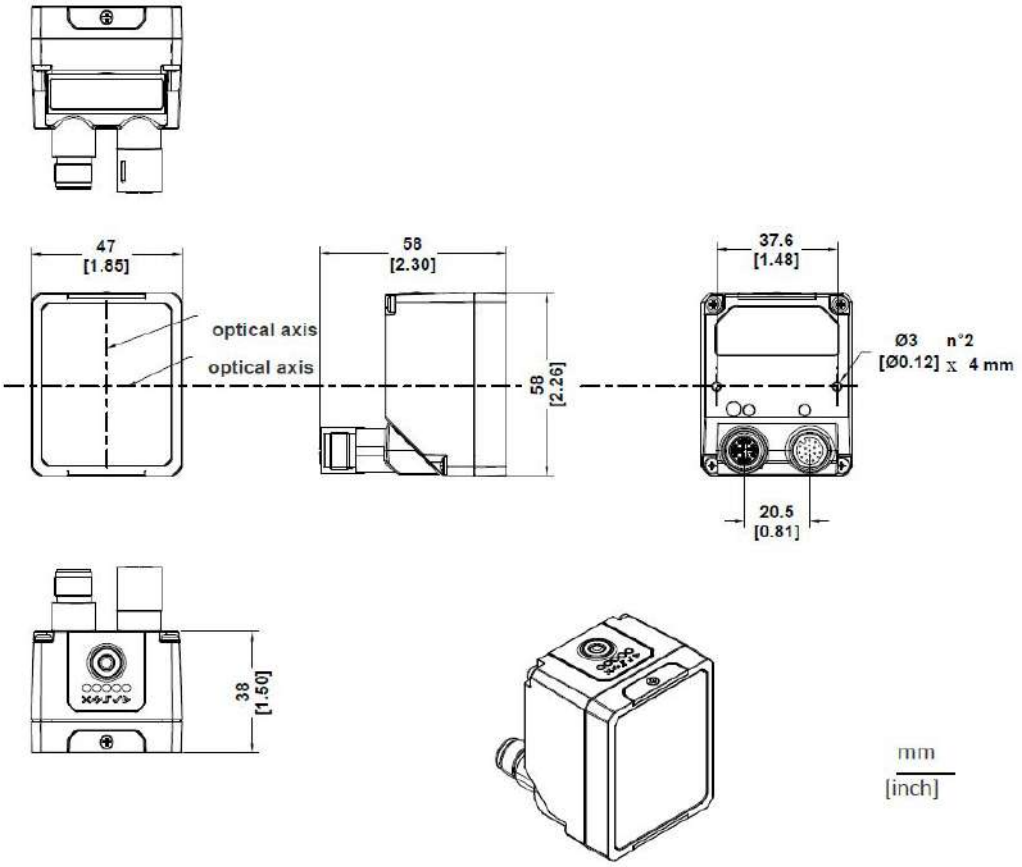


Figure 12 - Overall Dimensions; Connector at 90°

MOUNTING AND POSITIONING SMART-VS

Using the Smart-VS mounting brackets you can obtain rotation on the various axes of the device as shown in the diagram below:



Figure 13 - Positioning with Mounting Bracket

CHAPTER 4
ELECTRICAL CONNECTIONS

Smart-VS can be connected through one of the available CAB-GDxx accessory cables.

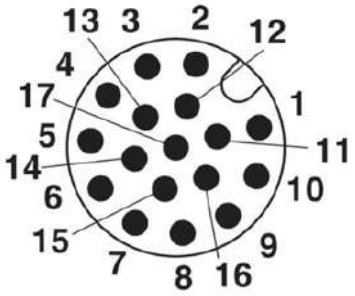


Figure 14 - M12 17-pin Power and I/O Connector

The recommended cables 95A900052 and 95A900053 (refer to "Accessories" on page 22) terminate in an M12 17-pin connector on the Smart-VS side and 9 stripped wires on the other side. These wires have the following functionalities:

POWER AND I/O CONNECTOR PINOUT (cables with 9 stripped wires)			
PIN	**NAME**	**COLOR**	**DESCRIPTION**
1	Vdc	Brown	Power supply input voltage +
2	GND	Blue	Power supply input voltage -
Connector case	Chassis		Connector case provides electrical connection to chassis
6	I1A	Yellow	I1A Trigger Input A (polarity insensitive)
5	I1B	Pink	I1B Trigger Input B (polarity insensitive)
13	I2A	Green	I2A Remote Teach A (polarity insensitive)
3	I2B	White	I2B Remote Teach B (polarity insensitive)
9	O1	Red	Data Valid PP
8	O2	Gray	GOOD Output PP
16	O3	Black	NO GOOD Output PP

WARNING: For proper installation, it is recommended to trim out all unused wires.

Previous Smart-VS versions use CAB-GDxx cables that terminate in an M12 17-pin connector on the Smart-VS side and 17 stripped wires on the other side. These wires have the following functionalities:

POWER AND I/O CONNECTOR PINOUT (cables with 17 stripped wires)			
PIN	NAME	COLOR	DESCRIPTION
1	Vdc	Brown	Power supply input voltage +
2	GND	Blue	Power supply input voltage -
Connector case	Chassis		Connector case provides electrical connection to chassis
6	I1A	Yellow	I1A Trigger Input A (polarity insensitive)
5	I1B	Pink	I1B Trigger Input B (polarity insensitive)
13	I2A	White/Green	I2A Remote Teach A (polarity insensitive)
3	I2B	White	I2B Remote Teach B (polarity insensitive)
9	O1	Red	Data Valid PP
8	O2	Gray	GOOD Output PP
16	O3	Yellow/Brown	NO GOOD Output PP



WARNING: For proper installation, it is recommended to trim out all unused wires.

POWER SUPPLY

The power must be between 10 and 30 Vdc only.



NOTE: Each time the Smart-VS is powered on, the electronic focus motor system does extra work for system checking and setup. It is recommended to limit power cycles to a few times per day.

INPUTS

There are two optocoupled polarity insensitive inputs available on the reader: Input 1 (External Trigger) and Input 2 (Remote Button).

The External Trigger is necessary to start acquiring and processing images.

The Remote Button installation is recommended in cases where the device is not accessible and, in general, to have easier access to the Teaching function.

The electrical features of both inputs are:

V_{AB} = 30 Vdc max.

I_{IN} = 10 mA max.

The active state of these inputs are selected in software.

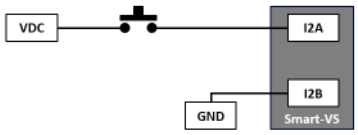
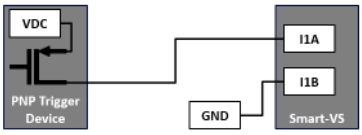
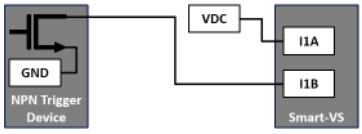
An anti-disturbance filter, by default, is implemented in software on both inputs. The value can be changed through the software parameter Debounce Filter. See "I/O Settings" on page 17 or refer to the Settings section of the Smart-VS WebApp User's Guide for further details on these parameters.

These inputs are optocoupled and can be driven by both NPN and PNP type commands.

NOTE: Polarity insensitive inputs assure full functionality even if pins A and B are exchanged.

Trigger and external push-button connection

The following table shows how to connect the Trigger and Remote inputs.

	RECOMMENDED CONNECTION	NOTES
External Push-Button		A Normally Open push-button is needed. I2A and I2B are polarity insensitive and can be inverted.
Trigger signal from PNP device		Connect the output of the PNP trigger device to I1A and the GND supply rail to I1B. I1A and I1B are polarity insensitive and can be inverted.
Trigger signal from NPN device		Connect the output of the NPN trigger device to I1B and the VDC supply rail to I1A. I1A and I1B are polarity insensitive and can be inverted.

OUTPUTS

The three outputs can be configured as NPN, PNP, and Push-Pull (default setting). They are short-circuit protected and the maximum load is 100 mA.

Output 2 and Output 3 serve as GOOD and NO GOOD Object signals. By default, they stay active for 10 ms.

Output 1 serves as Data Valid signal. It is activated 1 ms after the GOOD or NO GOOD Object signal activation, and deactivated 1 ms before the GOOD or NO GOOD Object signal deactivation.

The output activation length can be set via Smart-VS WebApp on the Settings page.

Digital I/O diagram

Image acquisition starts on the Leading edge of the input trigger signal (default, can be changed to Trailing edge via Smart-VS WebApp).

A Debounce time can be set on the Smart-VS WebApp to reject noisy trigger signals.

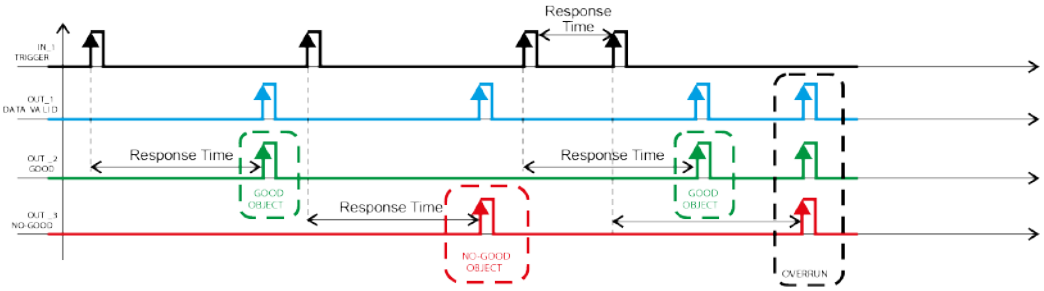
Each trigger event corresponds to a Data Valid signal indicating that the Output signals can be sampled.

The inspection result, i.e. a GOOD or NO GOOD object detected, is always available after 50 ms from trigger (response time).

If a Debounce time greater than zero has been set, the device response time is 50 ms plus the trigger Debounce time.

Overrun conditions (see "Troubleshooting" on page 41) are signaled raising both the GOOD and NO GOOD output signals.

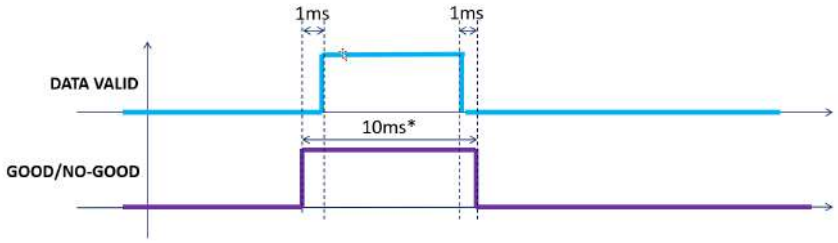
NOTE: Although not strictly necessary, it is recommended to connect and process all the output signals in order to have complete system information for diagnostics and troubleshooting.



Data Valid timing details

The Data Valid signal is raised 1 ms after the GOOD or NO GOOD signal is raised.

It is lowered 1 ms before the GOOD or NO GOOD signal is lowered.



*default, adjustable via Smart-VS WebApp

ON-BOARD ETHERNET INTERFACE

The on-board Ethernet Interface can be used for WebApp access and Telnet communication.

The WebApp is accessible opening an Internet browser (Google Chrome is recommended) and entering the device IP address 192.168.3.100 in the address bar.

Telnet communication can be established by connecting to device IP at port 23.

A **CAB-ETH-X-M0x** cable can be used to connect to a host device.

NOTE: Point-to-point connection is highly recommended.

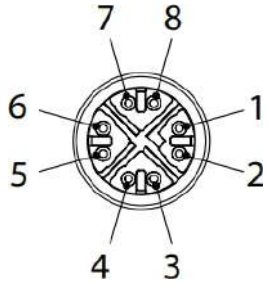


Figure 15 - M12 8-pin Standard Ethernet Network Connector

STANDARD ETHERNET NETWORK CONNECTOR PINOUT		
PIN	NAME	DESCRIPTION
1	TX+	Transmit data (positive pin)
2	TX-	Transmit data (negative pin)
3	RX+	Receive data (positive pin)
4	RX-	Receive data (negative pin)
5	nc	Not connected
6	nc	Not connected
7	nc	Not connected
8	nc	Not connected

CHAPTER 5
TYPICAL LAYOUTS

The following typical layouts refer to system hardware configurations. Dotted lines in the figures refer to optional hardware configurations within the particular layout.

NOTE: All software configurations are made through the Smart-VS WebApp, which connects to the reader through the on-board Ethernet interface.

SENSOR-LIKE SETUP

The Smart-VS is connected directly to the machine control system.

An optional push-button is recommended if Teaching is frequently performed for product batch changes or if the device is not easily accessible.

A presence sensor or the machine electrical phase is necessary to trigger image acquisition.

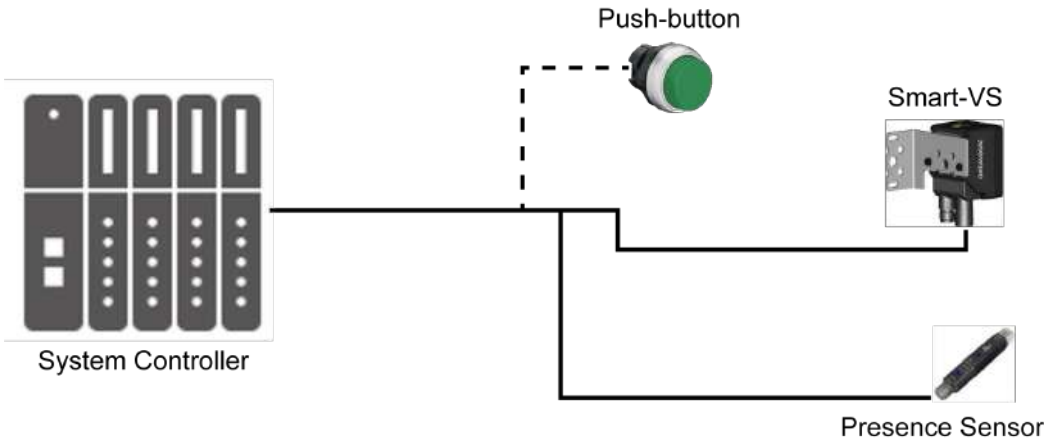


Figure 16 - Sensor-like Setup

WEBAPP FOR INITIAL CONFIGURATION AND OCCASIONAL MONITORING

In addition to the previous setup, a PC can be connected via Ethernet for initial configuration and occasional monitoring.

A point-to-point Ethernet connection is recommended.

A presence sensor or the machine electrical phase is necessary to trigger image acquisition.

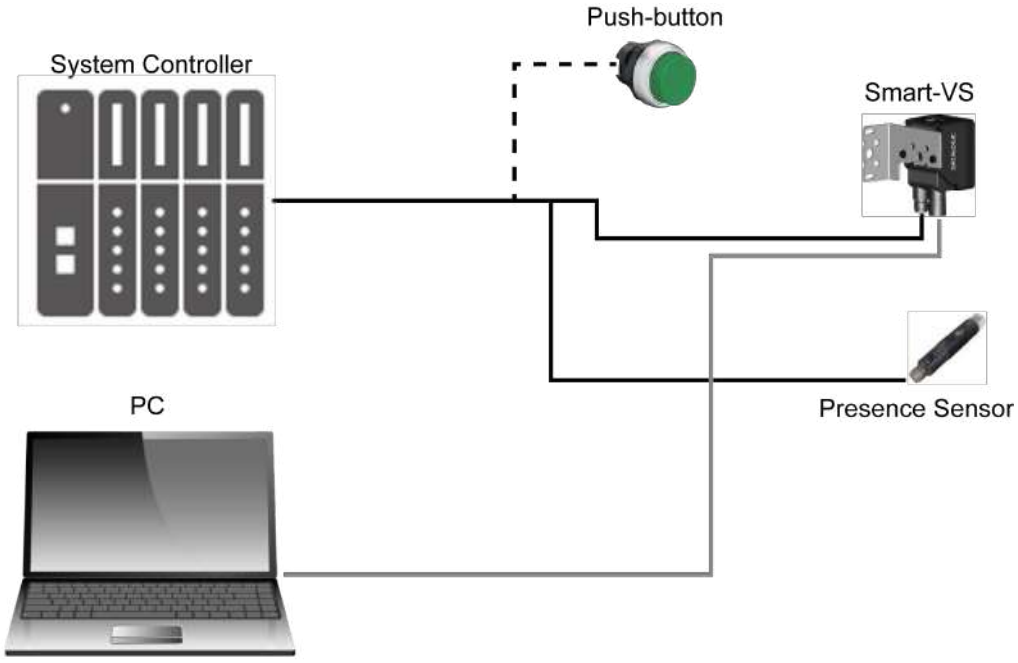


Figure 17 - WebApp Setup

SETUP WITH TELNET COMMUNICATION FOR JOB SWITCHING

If more than one Job is needed, Job switching can be done via Telnet communication.

An Ethernet switch can be used to ease the connection, but a dedicated LAN is still recommended, since the Smart-VS is configured with a fixed IP address.

A presence sensor or the machine electrical phase is necessary to trigger image acquisition.

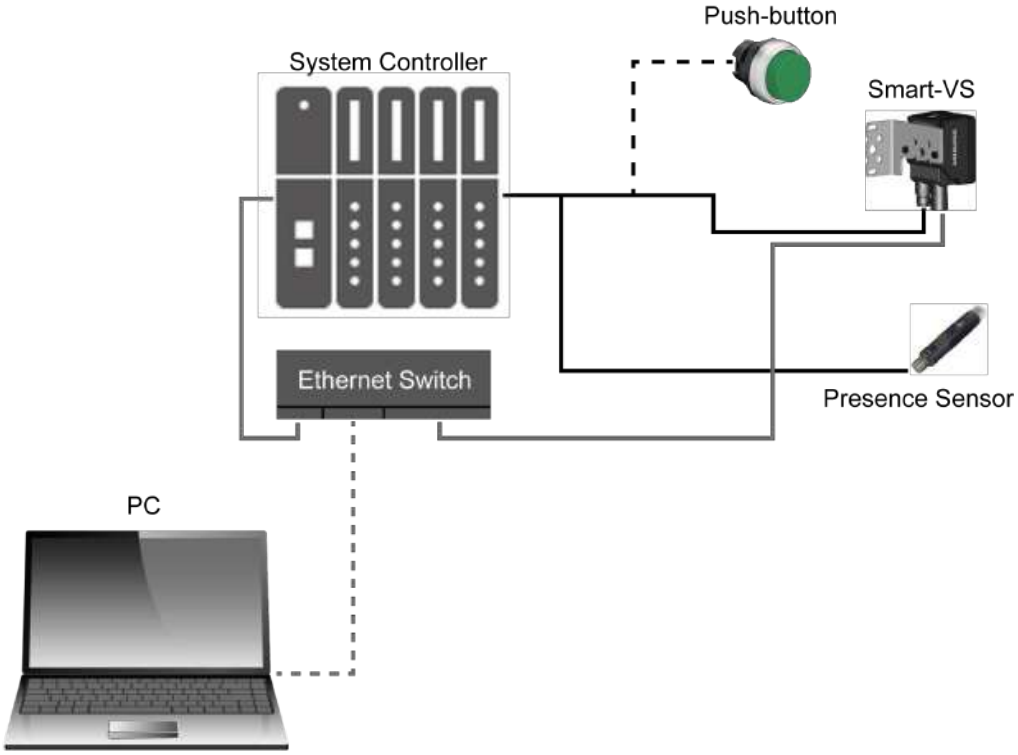


Figure 18 - Telnet Setup

For more information on Job switching via Telnet, refer to "Job Switching via Telnet Communication" on page 39.

CHAPTER 6
READING FEATURES

FOV CALCULATION

Use the data in the following table to calculate the FOV for your application, referring to Figure 19 and the formula below.

d_0	View Angle Horizontal	View Angle Vertical	View Angle Diagonal	Min Reading Distance
11 mm	19°	14.5°	24°	50 mm

The viewing angle has a tolerance of ± 1° depending on the reading distance.

$$FOVx = 2 [(d + d0) * \tan (\alpha_x/2)]$$

where:

FOV_x = horizontal, vertical or diagonal FOV

α_x = horizontal, vertical or diagonal viewing angles.

d = reading distance (in mm) from window surface to code surface

d_0 = offset (in mm) from center of lens to external window surface

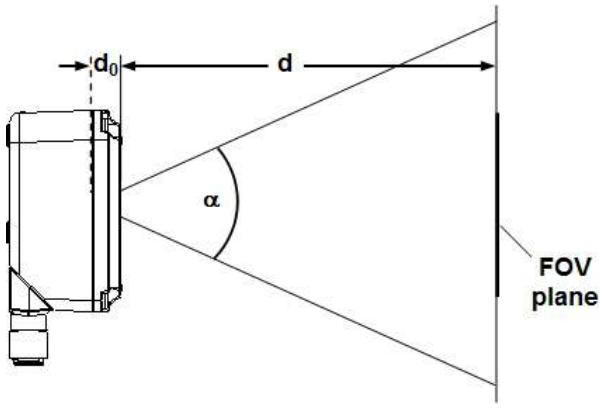


Figure 19 - Reading Distance References

Example

The FOV at a **reading distance of 100 mm** is:

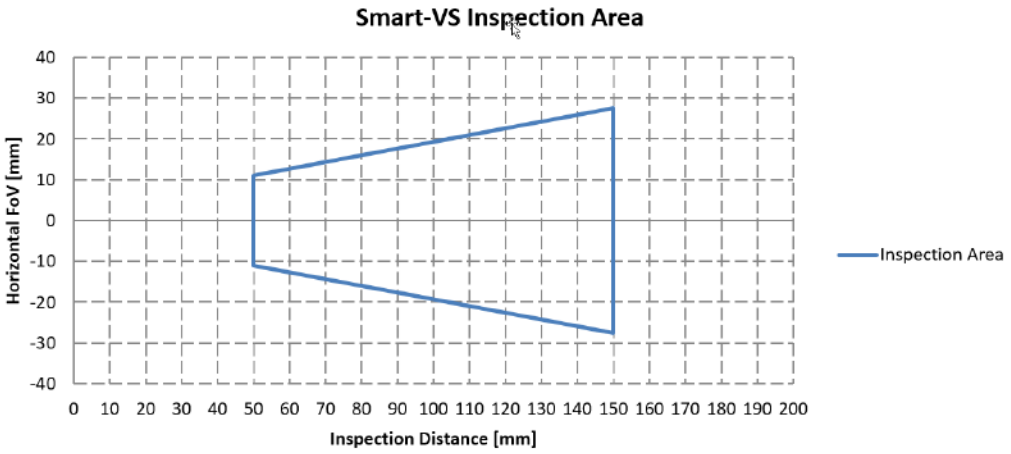
FOV_H = 2 [(100 mm + 11 mm) * tan (19°/2)] ≅ **37 mm**

FOV_V = 2 [(100 mm + 11 mm) * tan (14.5°/2)] ≅ **28 mm**

DETECTION DIAGRAM

The following diagram shows the maximum obtainable Field of View for object inspection.

NOTE: All features of interest related to the object inspection should be included in the Field of View.



CHAPTER 7
MULTI JOB CONFIGURATION

The Smart-VS can manage Multi Job operations. This feature is useful when different object batches must be inspected within a single machine.

Up to 32 Jobs can be stored in the device internal memory.

Job switching can be performed either via Smart-VS WebApp or Telnet communication.

JOB SWITCHING VIA SMART-VS WEBAPP

To recall a Job in the Smart-VS WebApp, go to the Monitoring page and select the Job ID from the *Current Job* list.



Confirm the running Job by clicking on the *Save and Play* button before leaving the page. This will also be the Job loaded at the next system startup.

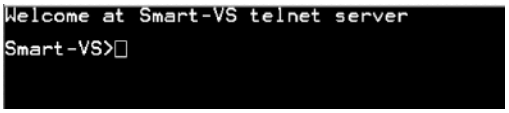


JOB SWITCHING VIA TELNET COMMUNICATION

Jobs can also be recalled via Ethernet connection using a Telnet communication.

Open a TCP/IP Telnet session using a Telnet client with the following parameters:

- IP: Device IP Address (default 192.168.3.100)
- Port: 23



The following commands are available to switch Jobs and retrieve information on the running Job:

Command [a]	`job load <Bank#>#CR`
Description	Load a bank (0 to 31). Changes will be applied to the next system startup.
Example	`job load 6#CR`

Output message	`job load <Bank#>#CR#LFDONE;#LF#CR#CR#LFSmart-VS>`
	
	or
	
	`job load <Bank#>#CR#LFFAIL;Failed to get status, already in configuration#LF#CR#CR#LFSmart-VS>`

Command [a]	`job status#CR`
Description	Retrieve information on the current bank.
Example	`job status#CR`

Output message	`current loaded bank:#CR#LF<Bank#>;#LF#CR#CR#LFSmart-VS>`
	
	`current startup bank:#CR#LF<Bank#>#LF#CR#CR#LFSmart-VS>`

a. All commands must end with #CR (carriage return, byte dec value = 13)

NOTE: If the command string is sent through a pre-allocated array larger than the string itself, the unused portion of the array must be initialized to 00 (bytes) i.e. the [NULL] character.

CHAPTER 8
MAINTENANCE

CLEANING

Clean the lens cover periodically for continued correct operation of the device. See "General View" on page xii.

Dust, dirt, etc. on the lens cover may alter the device performance.

Repeat the operation frequently in particularly dirty environments.

Use soft material and alcohol to clean the lens cover and avoid any abrasive substances.

CHAPTER 9
TROUBLESHOOTING

TROUBLESHOOTING GUIDE	
PROBLEM	**SUGGESTION**
Overrun	Overrun occurrences are mostly due to a noisy trigger signal and/or a too fast sequence of trigger events (e.g. more than two events every 50 ms). Choose a proper Debounce filter time and/or adjust the line speed to have 20 pieces per second.
The Trigger LED does not light up when the object passes in front of the sensor	• In case of PNP signal, check that one of the two trigger inputs (I1A or I1B) is connected to the signal coming from the sensor (or PLC) and the other input is connected to 0Vdc. • In case of NPN signal, check that one of the two trigger inputs (I1A or I1B) is connected to the signal coming from the sensor (or PLC) and the other input is connected to +Vdc. • Check that the trigger sensor switches regularly when the object passes through. • Check that the Trigger Debounce is not set too high.
The Remote input signal does not start the Teaching procedure	• In case of PNP signal, check that one of the two Remote inputs (I2A or I2B) is connected to the signal coming from the button (or PLC) and the other input is connected to 0Vdc. • In case of NPN signal, check that one of the two Remote inputs (I2A or I2B) is connected to the signal coming from the button (or PLC) and the other input is connected to +Vdc. • Check the correct operation of the button using a multimeter. • Check that the Button Debounce is not set too high.
The PLC does not receive GOOD or NO GOOD signals from the Smart-VS sensor	• Check that the GOOD object and NO GOOD object LEDs light up correctly. • If the PLC requires a PNP signal, check that the Smart-VS outputs are set as Push-Pull or PNP. • If the PLC requires a NPN signal, check that the Smart-VS outputs are set as Push-Pull or NPN.
The Power On LED is off	• Check that Pin 1 (brown wire) is connected to +Vdc and Pin 2 (blue wire) is connected to 0Vdc. • Use a multimeter to check that the power supply voltage is 10 to 30 Vdc (see "Technical Features" on page 43).

TROUBLESHOOTING GUIDE	
PROBLEM	**SUGGESTION**
All HMI LEDs blink on power-up	This may occur after a software update that broke compatibility with the previous version. • Keep using latest software update (delete stored jobs): access the WebApp and clear all configurations using the *Clear Configurations* button on the *Utilities* page. Discard latest software update (do not delete stored jobs): perform a rollback procedure via HMI (see "Firmware Rollback and Reset Configurations (optional)" on page 6).
All HMI LEDs are off	This may occur after a software downgrade that compromise compatibility of the saved data. • Reset the configurations. If the issue occurs: • Perform a new rollback.

APPENDIX A
TECHNICAL FEATURES

ELECTRICAL FEATURES	
Power	
Supply Voltage	10 to 30 Vdc
Consumption	0.40 - 0.14 A (4.2 W) max
Communication Interfaces	
Ethernet (embedded) [a]	10/100 Mbit/s
Inputs	Opto-coupled and polarity insensitive
Max. Voltage	30 Vdc
Max. Input Current	10 mA
Outputs	PP, NPN or PNP short circuit protected;
Max. Output Voltage Drop	3V
V_{OUT} HIGH (Max. / Min.)	V_{supply} / V_{supply} - 3V
V_{OUT} LOW (Max. / Min.)	3V / 0V
I_{LOAD} Max.	100 mA
GOOD and NO GOOD output response time	50 ms from input trigger

a. The embedded Ethernet interface is intended for configuration only through connection to the device IP address 192.168.3.100. Point-to-Point connection is recommended.

OPTICAL FEATURES	
Operating Distance	50 - 150 mm
View Angle	19°

ENVIRONMENTAL FEATURES	
Operating Temperature [a]	0 to 50 °C (32 to 122 °F)
Storage Temperature	-20 to 70 °C (-4 to 158 °F)
Max. Humidity	90% non condensing
Vibration Resistance **EN 60068-2-6**	14 mm @ 2 to 10 Hz; 1.5 mm @ 13 to 55 Hz; 2 g @ 70 to 500 Hz; 2 hours on each axis
Shock Resistance **EN 60068-2-27**	30g; 11 ms; 3 shocks on each axis
Protection Class [b] **EN 60529**	IP65 and IP67

a. high ambient temperature applications should use metal mounting bracket for heat dissipation.
b. when correctly connected (fully tightened) to IP67 cables with seals.

PHYSICAL FEATURES	
Dimensions Smart-VS with connector at 0° Smart-VS with connector at 90°	H x W x L 78 x 47 x 38 mm (3.1 x 1.9 x 1.5 in) 58 x 47 x 58 mm (2.3 x 1.9 x 2.3 in)
Weight	173 g (6.1 oz)
Material	Aluminum with PMMA protective window cover

APPENDIX B
TCP/IP COMMUNICATION

CONNECTION

The TCP/IP communication requires that a connection be initiated between the two devices. The Smart-VS (server) is listening on port 1023 and the PLC or PC (clients), must send a connection request to establish communication. When the communication is established, commands can be sent to the device.

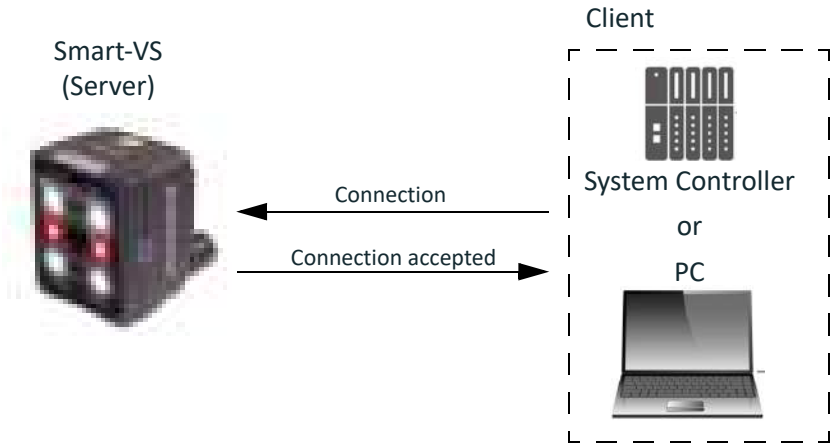


Figure 20: Connection to device

PROTOCOL OVERVIEW

The client sends a command to the Smart-VS and receives a reply with the result. The Smart-VS (server) is able to process only one command at a time. There are two types of commands:

1. Device control commands
2. Commands for Upload and Download of binary files

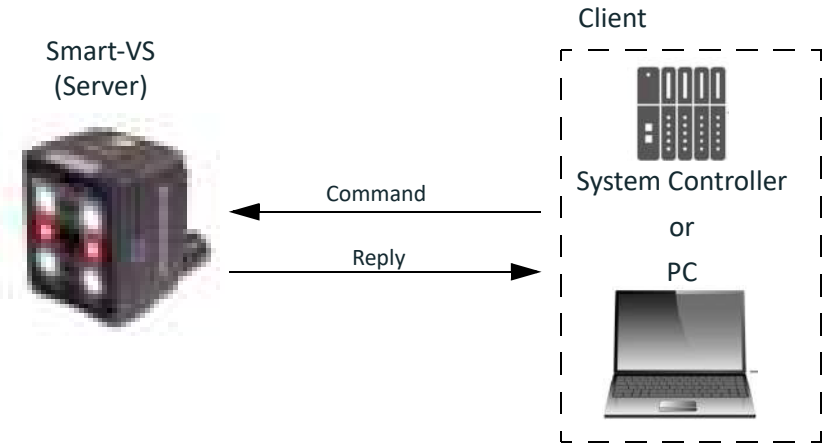


Figure 21: Example of communication

Description of available features

There are two macro-groups of features:

1. Synchronous features: begins features that require short execution times. The result will be available directly in the reply.
2. Asynchronous features: begins features that take long execution times. The result must be requested once the completion of the task has been verified. In the event that the task is immediately finalized without previously verifying that the status has ended, the call will be blocked and the reply will be sent at the end of the task execution.

Use of synchronous features

When a synchronous feature is performed, the reply is sent at the end of the execution. See example in the figure.

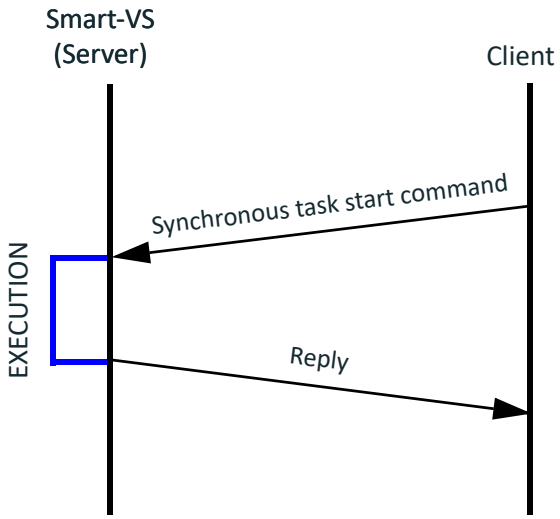


Figure 22: Example of using a synchronous feature

If the feature requires results to be sent, they will be contained in the reply.

Use of asynchronous features

An asynchronous feature must be started with the appropriate command. It will not be possible to run multiple asynchronous features at the same time. During the execution of the asynchronous feature, the device can be queried about the status of the activity. Once the activity is terminated, it will be possible for the client to request the finalization of the task, if provided, the finalization command will also contain the reply. It will not be possible to start another asynchronous task if the previously executed task is not terminated first.

Scenario 1

The client queries the device on the status of the task and, once the completion is verified, finalizes the task.

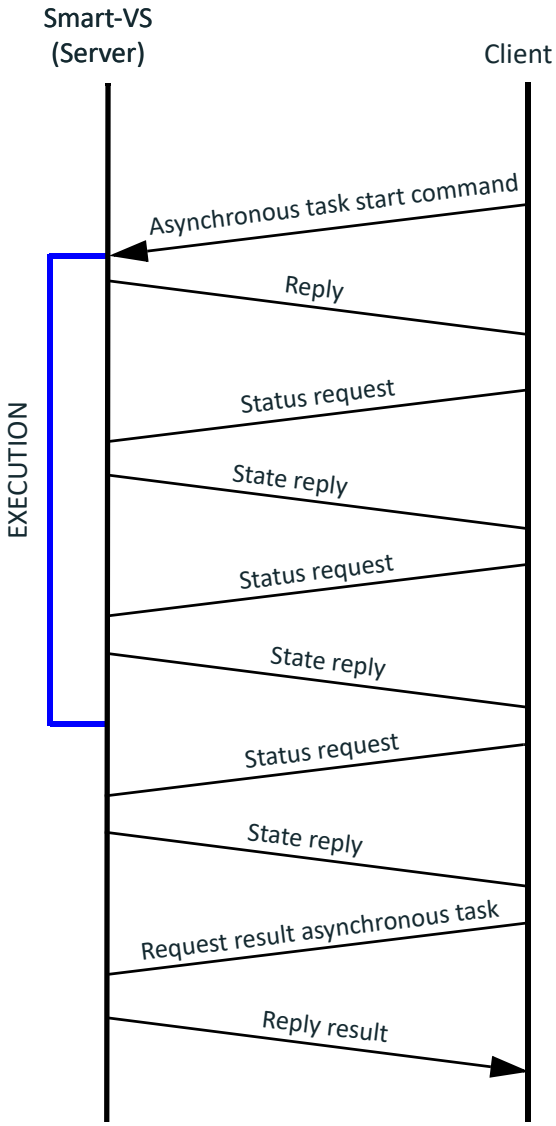


Figure 23: Example of using an asynchronous feature

Scenario 2

The client starts the task and immediately asks for its finalization. The reply will be sent at the end of the task.

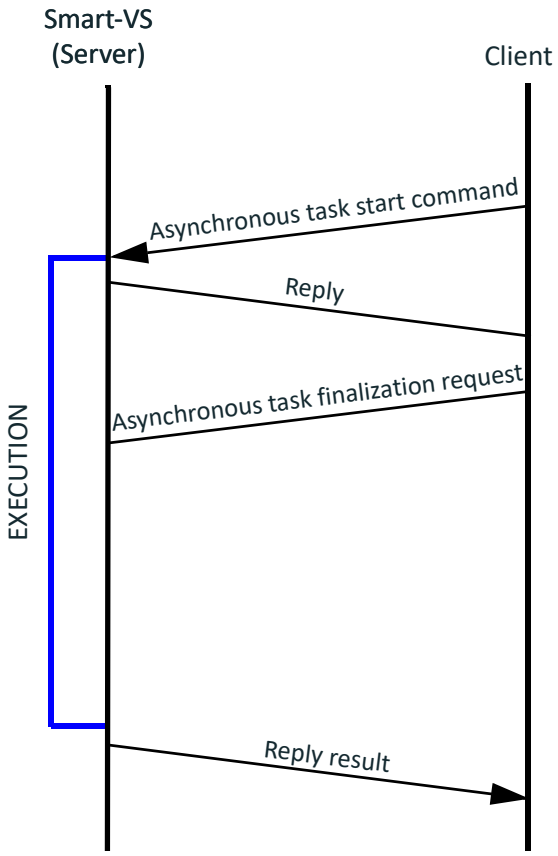


Figure 24: Example of synchronous use of an asynchronous feature

Delimiters and separator

The grammar of the protocol provides a character used as a separator (S) of the fields within the message and a two-character sequence used as a delimiter (D) of the message.

	CHARACTER\S	DECIMAL VALUE	HEX VALUE
Separator	;	59	0x3B
Delimiter	<CR><LF>	13 10	0x0D 0x0A

DEVICE CONTROL COMMANDS

These commands allow to remotely control some features of the device. They are encoded in ASCII with a defined grammar.

Command format

Command	S	Parameter 1	S		S	Parameter N	D

Figure 25: Control command frame structure

- Command → encoding containing the command.
- S → separator.
- Parameter 1-N→ command parameter list. The number of parameters is variable.
- D → Delimiter.

Example of command with one parameter:

CNGJB	;	1	CR	LF

Figure 26: Example using command with one parameter

Example of command without parameter:

GTRJB	CR	LF

Figure 27: Example of using a command without parameters

Reply format

Command	S	Return Code	S	Value 1		S	Value N	D

Figure 28: Control command reply frame structure

- Command → command to which the reply is related.
- Return Code → contains the result of the command, a code that identifies whether the command was accepted or not, and identifies the type of error (See Tabella 2 on page 51).
- S → separator.
- Value 1-N → return values of the command. It depends on the type of command.
- D → Delimiter

Positive reply example with return values:

GTRJB	;	0	;	15	;	Job Name	CR	LF

Figure 29: Example of reply to a successful command with return values

Positive reply example with no return values:

CNGJB	;	0	CR	LF

Figure 30: Positive reply example with no return values

Example of a negative reply:

CNGJB	;	1	CR	LF

Figure 31: Example of a reply to a failed command

List of available device control functions

COMMAND	FUNCTION	PARAMETER	TYPE
CRTJB	Create a new job	Bank number to be configured, name to be assigned to the job	asynchronous
MDFJB	Modify a job	Bank number to be changed	synchronous
EXTJB	Exit the configuration without saving		synchronous
TRNJB	Train job		asynchronous
CNGJB	Change job	Bank number to load	synchronous
BNKST	Bank Status		synchronous
GTRJB	Current Bank	See here	synchronous
GTDVCS	Device Status		synchronous
ACQIMG	Reference image acquisition	Class of the image	synchronous
CLRBNK	Clear Bank	Bank number	synchronous
CLRJBS	Clear Jobs		synchronous
GTATS	Get Async Task Status		synchronous
FNZJB	Finalize Job creation		synchronous
FNZTRN	Finalize Train		synchronous

Table 1: Device control command list

Return codes

The return code is contained in all replies as a parameter following the name of the command to which the reply is related. The return code allows to understand if the command was successful or rejected.

CODE	NAME	DESCRIPTION
0	Success	The command was successful
1	NotInSession	The device is running and the command cannot be executed
2	Failed	Generic command failure
3	-	(Reserved)
4	NotInJobEditing	The creation or modification of a job has not been started
5	-	(Reserved)
6	OtherInProgress	Another asynchronous task is in progress or an asynchronous task has not been finalized, therefore it is not possible to start a new one
7	-	(Reserved)
8	InvalidInput	Invalid input
9	-	(Reserved)
10	AlreadyInConfigura-tion	A configuration has already been started, it is not possible to start a new one before closing the previous one
11	MaxNumberOfImage	Maximum number of images reached
12	NotInProgress	No asynchronous tasks running
13	ProtocolError	Syntax error in the protocol
14	UnknownMethod	Call unavailable (call name error)
99	NotRelevant	Parameter not relevant

Table 2: List of possible return codes in reply

BankStatus

The BankStatus is contained in many replies and allows you to understand the status of a bank.

VALUE	NAME	DESCRIPTION
0	Empty	Empty
1	Available	Full
2	HasWarning	It is necessary to connect to the Web Page to check possible alarms relating to the bank
3	NotAvailable	The bank requires to be retrained
4	Emergency	The device memory is corrupted. It is necessary to connect to the Web Page to restore full operation of the device
128	NotRelevant	Parameter not relevant

Table 3: List of possible Bank Status values in reply

CRTJB (CReaTeJoB)

Allows you to start a configuration session. The device switches from the “Running” state to the “Configuration” state. By sending the command, the auto-setup of the photometric parameters and the focus distance is automatically done based on the object positioned in front of the device. The auto-setup procedure requires the object to be statically positioned in front of the sensor.

Syntax:

CRTJB;BankId;JobName<CR><LF>

Possible return codes:

- Success → Command successful.
- InvalidInput → The specified bank is out of the range (0-31)
- OtherInProgress → Another asynchronous task is in progress. This task cannot be started. Finalize the previously started task.
- AlreadyInConfiguration → The device is already in configuration, it is not possible to start a new one. Finalize the previous configuration by completing it or exiting without saving.
- Failed → The device is already in the configuration state (HMI, WebApp).



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Complete the job creation procedure**
- **Interrupt the procedure by exiting with the appropriate command**
- **Close the client connection with the device**

MDFJB (MoDiFyJoB)

It allows to add images to a previously configured job thus using the device's incremental training functionality. The device switches from the “Running” state to the “Configuration” state.

Syntax:

MDFJB;BankId<CR><LF>

BankId: Bank number on which to create the new job (may not be the current bank).

Reply:

MDFJB;ReturnCode <CR><LF>

Possible return codes:

- Success → Command successful.
- InvalidInput → The specified bank is out of the range (0-31) or no editable jobs are saved in the specified bank.
- AlreadyInConfiguration → no editable jobs are saved in the specified bank. Finalize the previous configuration by completing it or exiting without saving.
- Failed → The device is already in the configuration state (HMI, WebApp).



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Complete the job creation procedure**
- **Interrupt the procedure by exiting with the appropriate command**
- **Close the client connection with the device**

EXTJB (EXiTJoB)

It allows to exit without saving an editing session\job creation.

Syntax:

EXTJB<CR><LF>

Reply:

EXTJB;ReturnCode<CR><LF>

Possible return codes:

- Success → Command successful.
- NotInSession→ The command was called without a configuration session being started (the device is in running state).
- NotInJobEditing → The device is in configuration but no job is being edited / created.

TRNJB (TRaiNJoB)

Allows to start training the job being edited / created. The command is successful only if all the necessary reference image acquisitions have been made. The job is automatically trained.

Syntax:

TRNJB<CR><LF>

Reply:

TRNJB;ReturnCode<CR><LF>

Possible return codes:

- Success → Command successful.
- NotInSession → The command was called without a configuration session being started (the device is in running state).
- NotInJobEditing → The device is in configuration but no job is being edited / created.
- OtherInProgress → Another asynchronous task is in progress and therefore it is not possible to launch the training. Finalize the previous task and start the training.

CNGJB (ChaNGeJoB)

Allows to change the job in progress.

Syntax:

CNGJB;BankId<CR><LF>

BankId: Bank number to be used.

Reply:

CNGJB;ReturnCode;BankStatus;JobName <CR><LF>

JobName: Name of the job contained (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → Command successful.
- InvalidInput → The specified bank is out of the range (0-31) or no editable jobs are saved in the specified bank.
- AlreadyInConfiguration → The device is in configuration, it is not possible to change the bank. Finalize the previous configuration by completing it or exiting without saving.
- Failed → The device is already in the configuration state by an external interface (HMI, WebApp).

BNKST (BaNKSTatus)

It allows to check if a bank contains a job or not. The reply also contains the job name.

Syntax:

BNKST;BankId<CR><LF>

BankId: Number of the bank to be checked.

Reply:

BNKST;ReturnCode;BankStatus;JobName<CR><LF>

BankStatus: bank status.

JobName: Name of the job contained (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → Command successful.
- Failed → The device is in a job editing session. It is not possible to check the bank status. Finalize the session tasks (if any) by completing it or exiting without saving.
- InvalidInput → The specified bank is out of the range (0-31).

GTRJB (GeTRunningJoB)

Allows to check which job is loaded as the currently running job.

Syntax:

GTRJB<CR><LF>

Reply:

GTRJB;ReturnCode;BankId;BankStatus;JobName<CR><LF>

BankId: Bank number currently loaded.

JobName: name of the job present in the bank (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → Command successful.

GTDVCS (GeTDeViCeStatus)

It allows to check if the device is in the job execution or paused status.

Syntax:

GTDVCS<CR><LF>

Reply:

GTDVCS;ReturnCode;DeviceStatus<CR><LF>

DeviceStatus:

CODE	DEVICE STATUS
0	Running
1	Pause (the pause status is controlled by the device that carried out the command)
2	Pause (the pause status is controlled by another device)

Possible return codes:

Success → Command successful.

ACQIMG (ACQuireIMaGe)

It allows acquiring an image that will be saved among the reference images of the specified class within the job being edited/created.

Syntax:

ACQIMG;LabelID<CR><LF>

LabelID: index of the label to be assigned to the acquired image.

INDEX	LABEL
0	GOOD
1	NO GOOD

Reply:

ACQIMG;ReturnCode<CR><LF>

Possible return codes:

- Success → Command successful.
- NotInSession → The command was called without a configuration session being started (the device is in running state).
- NotInJobEditing →The device is in configuration but no job is being edited/created.
- InvalidInput → The index of the indicated label cannot be used (index> 1).
- MaxNumberOfImage → The image cannot be acquired because the maximum number of images that can be captured has been reached (per class or for the entire job).

CLRBNK (CLeaRBaNK)

Cancel the job on the bank indicated by the parameter.

Syntax:

CLRBNK;BankId<CR><LF>

BankId: Bank number on which to cancel the job (may not be the current bank).

Reply:

CLRBNK;ReturnCode<CR><LF>

Possible return codes:

- Success → Command successful.
- AlreadyInConfiguration→ The command was called while creating\editing a job (through any interface).
- InvalidInput → The bank number to be configured is invalid (bankId > 31).
- Failed → The specified bank does not contain any jobs.

CLRJBS (CLeaRJoBS)

Delete all jobs on the device.

Syntax:

CLRJBS<CR><LF>

Reply:

CLRJBS;ReturnCode<CR><LF>

Possible return codes:

- Success → Command successful.
- AlreadyInConfiguration→ The command was called while creating\editing a job (through any interface).

GTATS (GeTAsyncTaskStatus)

Query the device about the status and type of asynchronous task in progress.

Syntax:

GTATS<CR><LF>

Reply:

GTATS;ReturnCode;AsyncTaskType;AsyncTaskStatus<CR><LF>

AsyncTaskType: identification code of the asynchronous task.

CODE	ASYNCHRONOUS TASK TYPE
0	Creating a new job
1	Job training
2	Creation of binary file containing a job
3	Creation of binary file containing device backup
4	Saving a file on the internal memory
99	Not relevant

AsyncTaskStatus: status of the asynchronous task.

CODE	ASYNCHRONOUS TASK STATUS
0	Task in progress
1	Task finished
99	Not relevant

Possible return codes:

- Success → Command successful.
- NotInProgress → No asynchronous tasks have been started.

FNZJB (FiNaliZeJoB)

Finalize the asynchronous command to create a new job (“CRTJB (CReaTeJoB)" on page 52).

Syntax:

FNZJB<CR><LF>

Reply:

FNZJB;ReturnCode <CR><LF>

Possible return codes:

- Success → Command successful.
- NotInProgress → No asynchronous tasks in progress.
- OtherInProgress → Another asynchronous task is in progress.
- Failed → The asynchronous task did not finish successfully.

FNZTRN (FiNaliZeTRaiNing)

Finalize the asynchronous job training command and actually save the job in flash (“TRNJB (TRaiNJoB)" on page 53).

Syntax:

FNZTRN<CR><LF>

Reply:

FNZTRN;ReturnCode;BankStatus;JobName <CR><LF>

JobName: Name of the job contained (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → Command successful.
- NotInProgress → No asynchronous tasks in progress.
- OtherInProgress → Another asynchronous task is in progress.
- Failed → The asynchronous task did not finish successfully.

Example of creating a new job

Below is the list of operations to be performed to create a new job via a remote client:

1. Connect to the device.
2. Place the **GOOD** object in front of the Smart-VS.
3. Send the **CRTJB** command specifying the bank on which the job is to be saved and the name to be assigned to the job. The device will start the focusing procedure and the adjustment of the photometric parameters.
4. Wait for the completion of the image auto-setup operation.
5. Finalize the image auto-setup with the **FNZJB** command.
6. Send the **ACQIMG** command specifying the label to be assigned to the object.
7. Place the **NO-GOOD** object in front of the Smart-VS.
8. Send the **ACQIMG** command specifying the label to be assigned to the object.
9. Send the **TRNJB** command. The device will train.
10. Wait for the training to finish.
11. Finalize the training through the **FNZTRN** command, thus saving the job on the device.

Example of incremental training

Below is the list of operations to be carried out to modify a job by adding additional images via a remote client:

1. Connect to the device.
2. Place the **GOOD** or **NO-GOOD** object in front of the Smart-VS.
3. Send the **MDFJB** command specifying the bank containing the job to be modified.
4. Send the **ACQIMG** command specifying the label to be assigned to the object.
5. Send the **TRNJB** command. The device performs a training.
6. Wait for the end of the training.
7. Finalize the training through the **FNZTRN** command, thus saving the job on the device.

BINARY FILE UPLOAD AND DOWNLOAD COMMANDS

These commands allow to upload and download binary files from the device and check the status of the procedures. They are divided into two types of commands:

1. Upload and download.
2. Procedure control.

More generally, the upload and download of the file are carried out through an exchange area, of variable size, which can contain only one file. Each time a command to create a binary file or upload a binary file is executed, the exchange area is overwritten.



NOTE: The binary files are encoded by the protocol in Base64 with padding.

Upload command format

The upload command remains encoded in ASCII.

Command	;	Binary File	CR	LF

- Command → encoding containing the command.
- Separator → standard protocol separator (see “Delimiters and separator" on page 48).
- Binary File → Base64 encoded binary file.
- Delimitatore → standard protocol delimiter (see “Delimiters and separator" on page 48).

Upload reply format

The reply is in ASCII format.

Command	;	Return Code	CR	LF

- Command → command to which the reply is related.
- Separator → standard protocol separator (see “Delimiters and separator" on page 48).
- Return Code → it contains the result of the command, a code that identifies whether the command has been accepted or not, and possibly the type of error.
- Delimiter → standard protocol delimiter (see “Delimiters and separator" on page 48).

Download command format

The download command remains encoded in ASCII.

Command	CR	LF

- Command → encoding containing the command.
- Delimiter → standard protocol delimiter (see “Delimiters and separator" on page 48).

Download reply format

The reply remains encoded in ASCII.

Command	;	Return Code	;	Binary File	CR	LF

- Command → command to which the reply is related.
- Separator → standard separator of protocol (see “Delimiters and separator" on page 48).
- Return Code → it contains the result of the command, a code that identifies whether the command has been accepted or not, and possibly the type of error.
- Separator → standard separator of protocol (see “Delimiters and separator" on page 48).
- Binary File → Base64 encoded binary file.
- Delimiter → standard delimiter of protocol (see “Delimiters and separator" on page 48).

DEVICE PROCEDURE CONTROL COMMANDS

Procedure control command format and return codes

The Procedure control command format is the same as the Device control command format, see “Device control commands" on page 49.

For the Return Codes see “Return codes" on page 51.

List of available procedure control functions

COMMAND	FUNCTION	PARAMETERS	TYPE
CRTJBF	Create Job File	Index of the bank on which the job is contained	asynchronous
CRTBCK	Create BackUp File		asynchronous
ULBF	Upload Binary File	Data to upload on the device	synchronous
DLBF	Download Binary File		synchronous
STJBF	Store job Binary file	Force overwrite bank and bank index on which to save the job	asynchronous
STBCK	Store Backup Binary file	Force overwrite in case one of the banks is full	asynchronous
FNZJBF	Finalize Job File Creation		synchronous
FNZBCK	Finalize BackUp File Creation		synchronous
FNZJST	Finalize Job File Storing		synchronous
FNZBST	Finalize BackUp File Storing		synchronous

CRTJBF (CReaTeJoBFile)

Allows the device to start the creation of the binary file containing the information relating to a single job. The device switches from the “Running” state to the “Configuration” state.

Syntax:

CRTJBF;BankId<CR><LF>

BankId: index of the bank containing the job whose binary file is to be created.

Reply:

CRTJBF;Returncode<CR><LF>

Possible return codes:

- Success → command successful.
- InvalidInput → Incorrect bank number or the bank does not contain any saved jobs.
- OtherInProgress → Another asynchronous task is in progress or has not been finalized.



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Finalize the creation of the package containing the job using the FNZ-JBF command**
- **Close the client connection with the device**

CRTBCK (CReaTeBaCKup)

Allows the device to start the creation of the binary file containing the device backup. The device switches from the “Running” state to the “Configuration” state.

Syntax:

CRTBCK <CR><LF>

Reply:

CRTBCK;Returncode<CR><LF>

Possible return codes:

- Success → command successful.
- OtherInProgress → another asynchronous task is in progress or has not been finalized.



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Finalize the creation of the package containing the job using the FNZ-BCK command**
- **Close the client connection with the device**

ULBF (UpLoadBinaryFile)

Allows you to upload a binary file to the exchange area on the device.

Syntax:

ULBF;BinaryFile <CR><LF>

BinaryFile: File that must be uploaded to the exchange area.

Reply:

ULBF;Returncode<CR><LF>

Possible return codes:

- Success → command successful.
- AlreadyInConfiguration → A configuration has already started and the file cannot be loaded. Exit the configuration.

DLBF (DownLoadBinaryFile)

Allows you to download a binary file from the exchange area on the device.

Syntax:

DLBF <CR><LF>

Reply:

DLBF;Returncode;Binaryfile<CR><LF>

BinaryFile: binary file downloaded from the exchange area.

Possible return codes:

- Success → command successful.
- AlreadyInConfiguration → a configuration has already started and the file cannot be loaded. Exit the configuration.
- Failed → the exchange area is empty.

STJBF (SToreJoBFile)

It allows to save the binary file containing the data of a single job, on the device's flash memory. The file must be previously uploaded to the exchange area. The device switches from the “Running” state to the “Configuration” state.

Syntax:

STJBF;Force;BankId<CR><LF>

Force: forces the bank to be overwritten if it already contains a job.

VALUE	FORCE
0	False
1	True

BankId: index of the bank on which the file is to be saved.

Reply:

STJBF;Returncode<CR><LF>

Possible return codes:

- Success → command successful.
- AlreadyInConfiguration → a configuration has already started and the file cannot be loaded.
- InvalidInput → corrupted file or not loaded (exchange area empty).
- Failed → The file loaded in the exchange area is not a file containing a single job.



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Finalize the creation of the package containing the job using the FNZ-JST command**
- **Close the client connection with the device**

STBCK (SToreBaCKup)

It allows to save the binary file containing the device backup on the device's flash memory. The file must be previously uploaded to the exchange area. The device switches from the “Running” state to the “Configuration” state.

Syntax:

STBCK;Force <CR><LF>

Force: forces the flash to be overwritten.

VALUE	FORCE
0	False
1	True

Reply:

STBCK;Returncode<CR><LF>

Possible return codes:

- Success → command successful.
- AlreadyInConfiguration → a configuration has already started and the file cannot be loaded.
- InvalidInput → corrupted file or not loaded (exchange area empty).
- Failed → the file loaded in the exchange area is not a file containing a BackUp.



CAUTION: To return to the “Running” state, you need to perform one of the following operations:

- **Finalize the creation of the package containing the job using the FNZBST command**
- **Close the client connection with the device**

FNZJBF (FiNalaZeJoBFile)

Finalizes the asynchronous command to create a binary file containing a single job (“CRTJBF (CReaTeJoBFile)" on page 61).

Syntax:

FNZJBF<CR><LF>

Reply:

FNZJBF;ReturnCode;FileSizeByte <CR><LF>

FileSizeByte: Returns the size of the created file in bytes. The size of the file to be downloaded from the device, due to the base64 encoding, can be obtained with the following formula:

$$4\left\lceil\frac{FileSizeByte}{3}\right\rceil$$

or *divided by 3, rounded up to the nearest integer, then multiplied by 4*.

Possible return codes:

- Success → command successful.
- NotInProgress → no asynchronous tasks in progress.
- OtherInProgress → another asynchronous task is in progress.
- Failed → package creation failed.

FNZBCK (FiNalaZeBaCKup)

Finalize the asynchronous command to create a binary file containing the device backup (“CRTBCK (CReaTeBaCKup)" on page 61).

Syntax:

FNZBCK<CR><LF>

Reply:

FNZBCK;ReturnCode;FileSizeByte <CR><LF>

FileSizeByte: Returns the size of the created file in bytes. The size of the file to be downloaded from the device, due to the base64 encoding, can be obtained with the following formula:

$$4\left\lceil\frac{FileSizeByte}{3}\right\rceil$$

or *divided by 3, rounded up to the nearest integer, then multiplied by 4*.

Possible return codes:

- Success → command successful.
- NotInProgress → no asynchronous tasks in progress.
- OtherInProgress → another asynchronous task is in progress.
- Failed → package creation failed

FNZJST (FiNaliZeJobSTorage)

Finalizes the asynchronous command to save a binary file containing a single job (“STJBF (SToreJoBFile)" on page 62).

Syntax:

FNZJST<CR><LF>

Reply:

FNZJST;ReturnCode;BankStatus;JobName <CR><LF>

JobName: Name of the job contained (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → command successful.
- NotInProgress → no asynchronous tasks in progress.
- OtherInProgress → another asynchronous task is in progress.
- Failed → saving the package did not finish successful.

FNZBST (FiNaliZeBackupSTorage)

Finalizes the asynchronous command to save a binary file containing the device backup (“STBCK (SToreBaCKup)" on page 63).

Syntax:

FNZBST<CR><LF>

Reply:

FNZBST;ReturnCode;BankStatus;JobName <CR><LF>

JobName: Name of the job contained (Empty Bank in the case of an empty bank).

Possible return codes:

- Success → command successful.
- NotInProgress → no asynchronous tasks in progress.
- OtherInProgress → another asynchronous task is in progress.
- Failed → saving the package did not finish successfully.

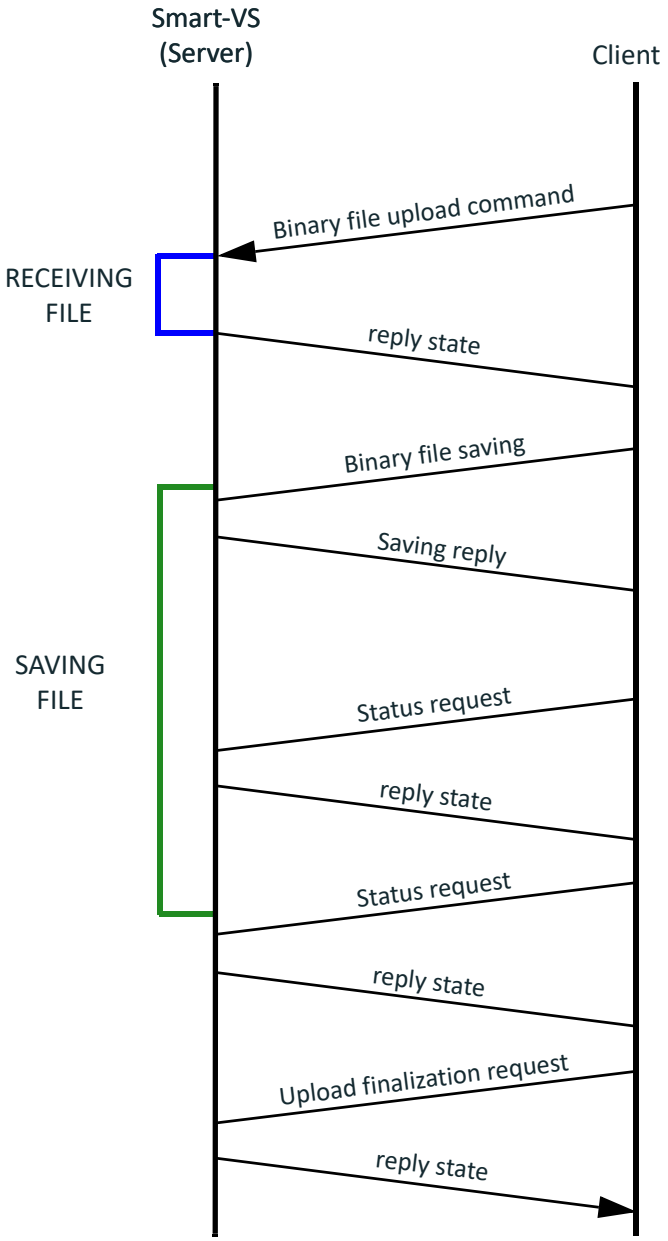
Example Upload of a binary file

In order to upload a binary file to a device it is necessary to perform two steps:

1. Upload the package to your device.
2. Save the package in Flash.

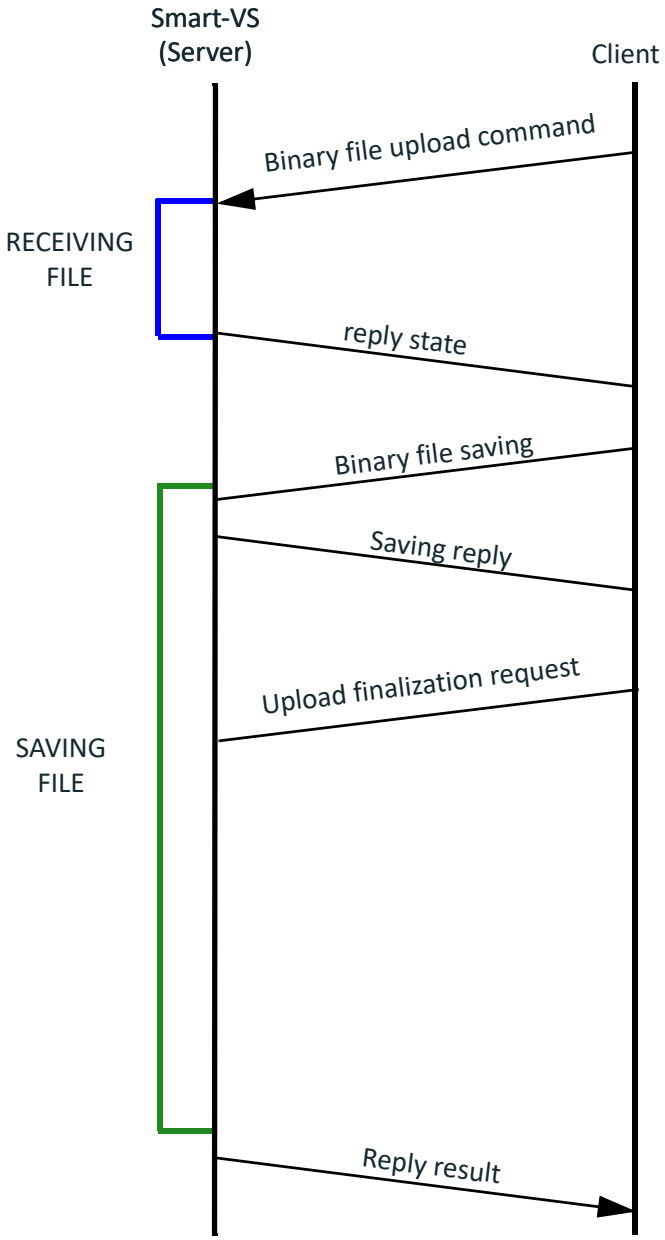
Scenario 1

The finalization of the saving is requested once the actual end of the task has been verified.



Scenario 2

The finalization of the saving is requested without verifying the actual end of the task.



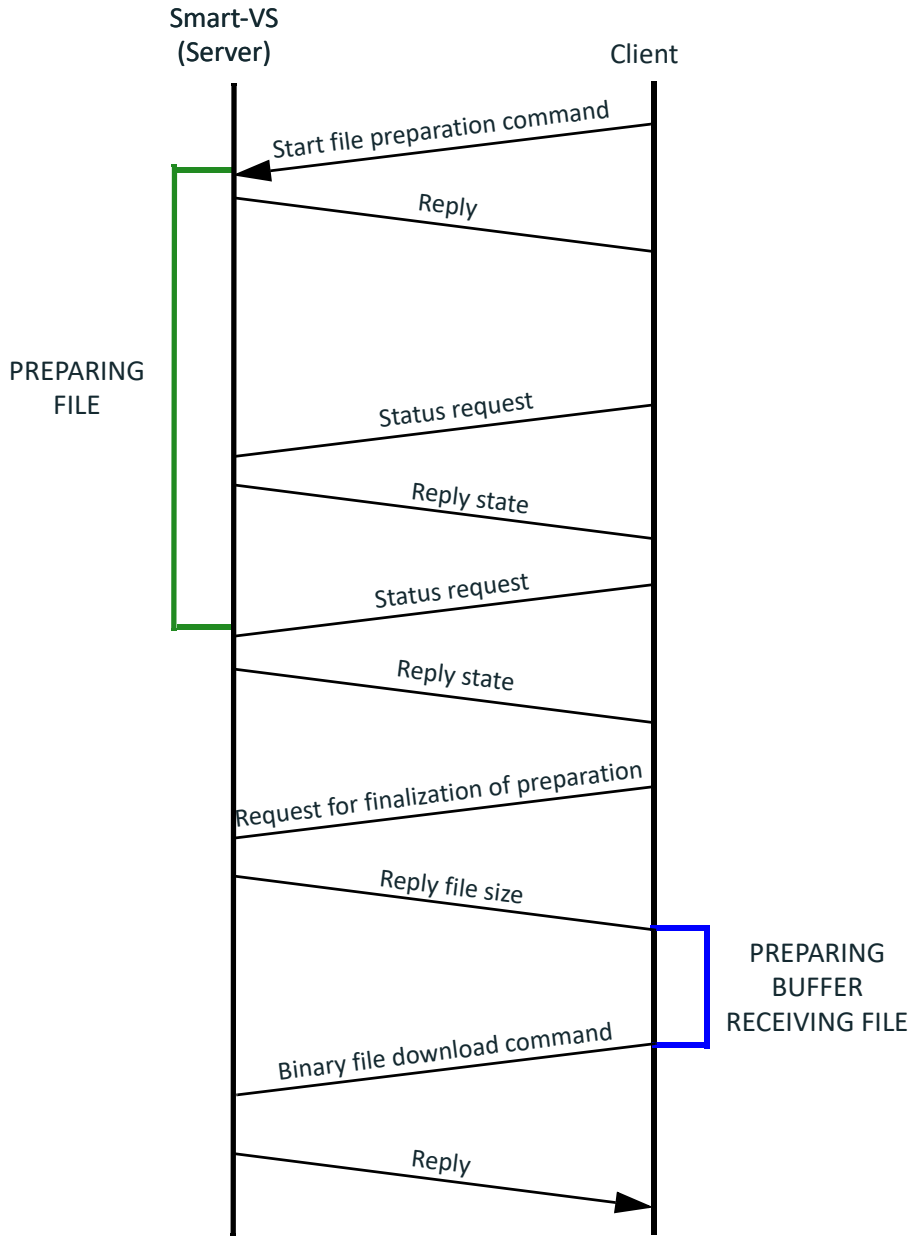
Example Downloading a binary file

In order to download a binary file from a device it is necessary to perform two steps:

1. Request package preparation from the device.
2. Download the package.

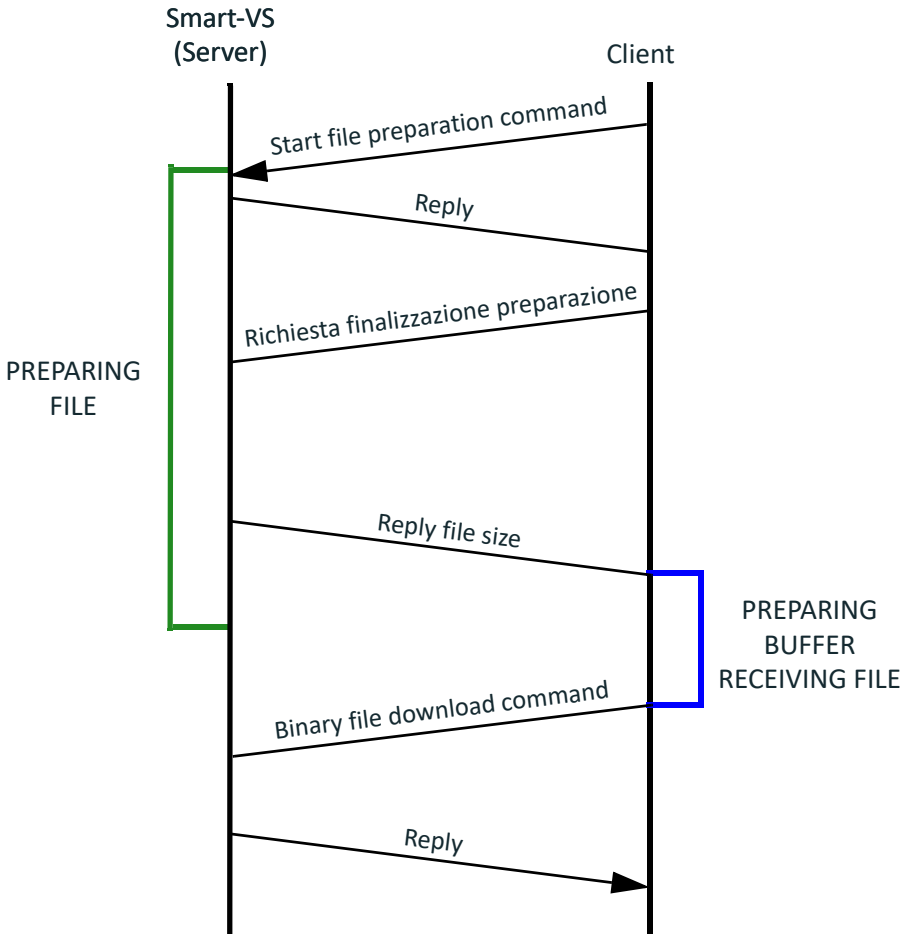
Scenario 1

The file size is requested once the package creation has been successfully completed.



Scenario 2

The finalization of the task is requested without verifying the actual end of the task.



Protocol exceptions

In the case of syntax errors, two exceptions can be generated by the protocol.

The format of the exceptions is the same as the format of the replies to the *upload**download* control commands (see “Reply format" on page 49).

There are two return codes dedicated to the case of exceptions in the protocol:

- Code 13 (ProtocolError) → indicates a syntax error on the command, such as the presence of an unexpected separator or an unnecessary parameter
- Code 14 (UnknownMethod) → the protocol is syntactically correct but the required functionality does not exist.

© 2020-2023 Datasensing S.r.l. • All rights reserved • Without limiting the rights under copyright, no part of this documentation may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means, or for any purpose, without the express written permission of Datasensing S.r.l. • Datasensing and the Datasensing logo are trademarks of Datasensing S.r.l. • Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S. and the E.U.



Datasensing S.r.l.
Strada S. Caterina 235 | 41122 Modena | Italy
Tel. +39 059 420411 | Fax +39 059 253973

806000500 (Rev A) March 2023