**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

# Vega Remote Control Server – Reference manual

Version 1.7

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

| Date | Version | Note | Author |
|---|---|---|---|
| 07/03/2012 | 1.0 | First english version | Andrea Monella |

Notes:

Tattile Srl assumes no responsibility for any errors, omissions or inaccuracies in this document..

Tattile Srl assumes no liability for any damages, direct or indirect, caused by the use of the information contained in this document.

Tattile Srl also reserves the right to modify or update at any time without prior notice the information contained herein.

It is not permitted the disclosure of the information contained herein without the prior written consent Tattile Srl.

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

# Indice generale

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

# 0 Introduction

The purpose of this paper is to describe the basic controls on devices that support the Vega Remote Client (VRC).

Each device provides a series of commands which may differ depending on the capabilities and needs of software running.

# 1 VRC communication protocol

VRC procotol is based on TOP (Tattile Object Protocol) header which identifies construction methods of the header and on a payload which contains VRC data to transmit.

For further information about standard protocol (IP, TCP) please refer to RFC 793 and 791.

**General:**

Every single request sent by VRC client needs an acknowledge message from VRC server, except in special cases.

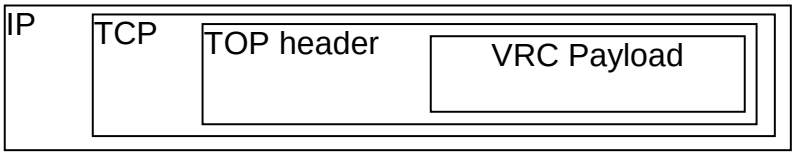
Default VRC server communication port (TCP) is 31000.

Whether TCP connection is unused (no data sent from client to server) for longer then 5 minutes, it will be automatically closed. To maintain an open connection is enough to send periodically a NOP command within connection timeout.

All data sent and received by VRC clients are structured according to the standard TOP (Tactile Object Protocol).

**TCP communication:**

In the TCP communication is sufficient to use TOP-VRC protocol to ensure packets integrity (TCP ensure integrity by itself).




## 1.1 TOP Header (CommandHeader)

TOP header contains following information:

- command to send to the camera
- channel selection
- error code

Communication protocol is based on TOP protocol (Tattile Object Protocol).

| Field | N. bytes | Description |
|---|---|---|
| a | 4 | Header Dimension |
| b | 4 | CommandCode |
| c | 4 (2 word) | Sender |
| d | 4 (2 word) | Receiver |
| e | 4 | Error |
| f | 4 | DataDimension |
| g | N | Message to send |

The fields must be filled according to the following:

| **CommandHeader: Fields description** | |
|---|---|
| Header Dimension | Header dimension(bytes). Set to 24. |
| CommandCode | Identification code of the command sent to the server |
| Sender | Command sender identification. If the sender is a PC, the value is 0xFFFFFFFF. |
| Receiver | Command receiver.<br>Some of Vega products have two cameras (BW and color). You have to use this field to distinguish to which camera this message is related (if necessary):<br>0: select BW channel<br>1: select Color channel<br>0xFFFFFFFF: no channel selected (for ex. to read file it's not necessary channel type). |
| Error | When VRC client sends a message this field has to be 0.<br>When VRC client receives a message from the server, in this field is reported error code generated by VRC server. |
| DataDimension | Data dimension (bytes), aligned to 32 bits, of data following the header to be sent regardless of the size of the header itself. |

WARNING: In order for the header TOP is accepted by the VRC server, it is necessary that 24byte header data are contained in the same ethernet packet. Otherwise the data is discarded and the socket is closed.

**VRC Client to VRC Server (PC to Vega)**

Transmission operations are:

- Adding to the message the header CommandHeader as follows:

a) HeaderDimension: "24"

b) CommandCode: code of the desired command.

c) Sender: Use 0xFFFFFFFF.

d) Receiver: channel of the device to which the message is intended. "0" for channel BW, "1" for color channel. If the message does not refer to a specific channel, use 0xFFFFFFFF.

e) Error: Enter "0" (no errors).

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

f) DATADIMENSION: Enter 0 if the command does not provide additional data to send, otherwise enter the number of data bytes for this command, aligned to the next multiple of 4byte.

- Sending message.

**Receiving data from VRC Server (Vega to PC)**

Receiving operations from VRC Server are:

- wait the incoming of the first packet of the message and extract the header CommandHeader. It will appear as follows:

a) HeaderDimension: "24"

b) CommandCode: command code received.

c) Sender: channel of the device to which the response to the command comes from. "0" channel BW "1" per color channel. If the command does not refer to a specific channel, that is 0xFFFFFFFF.

d) Receiver: Value: 0xFFFFFFFF.

e) Error: error code received. If "0" there were no errors.

f) DATADIMENSION: number of bytes, aligned to the next multiple of 4byte, the data that carries the command and is expected to receive after CommandHeader. The 24 byte command headers are excluded from this rule.

If DATADIMENSION is greater than zero, wait for the arrival of subsequent packets until they are received "DATADIMENSION" bytes.

If Error field contains an error value other than zero means that the camera has detected an error in the command.

**Example:**

For a better understanding, below an example of command transmission (25166, request image from the camera).

Other examples can be found in the Appendix.

**Command sent to the device (camera):**

| JPEG image aquisition *CLIENT-->SERVER* | | |
|---|---|---|
| **N. bytes** | **Description** | **Value** |
| 4 | Header Dimension | 24 |
| 4 | Command Code | 25166 |
| 4 | Sender | 0xFFFFFFFF |
| 4 | Receiver | Camera ID<br>0 BW camera<br>1 Color camera |
| 4 | Error | 0 |
| 4 | DataDimension | 28 |
| 4 | DataCode | 10015 |
| 4 | DataSize | 4 |
| 4 | DataBuffer | JPEG quality [1,100] |
| 4 | DataCode | 10069 |
| 4 | DataSize | 8 |
| 8 | DataBuffer | 4 byte Image Width in pixel<br>4 byte Image Height in pixel |

**Answer from device:**

| JPEG image aquisition *SERVER-->CLIENT* | | |
|---|---|---|
| **N. bytes** | **Description** | **Value** |
| 4 | Header Dimension | 24 |
| 4 | Command Code | 25166 |
| 4 | Sender | Camera ID<br>0 BW camera<br>1 Color camera |
| 4 | Receiver | 0xFFFFFFFF |
| 4 | Error | 0 |
| 4 | DataDimension | 28 + header and data image (40+IMAGE_SIZE) |
| 4 | DataCode | 2001 |
| 4 | DataSize | 12 |
| 12 | DataBuffer | Image time-stamp<br>1byte: day<br>1byte: month<br>2bytes:year<br>2bytes:hour<br>2bytes:minutes<br>2bytes:seconds<br>2bytes:milliseconds |
| 4 | DataCode | 2000 |
| 4 | DataSize | 40+IMAGE_SIZE |
| align_long(40+IMAGE_SIZE) | DataBuffer | Image Header:<br>4 byte: NOT USED<br>4 byte: NOT USED<br>4 byte: Image header size = 40<br>4 byte: IMAGE_SIZE (JPEG dimension)<br>4 byte: Image Width in pixel<br>4 byte: Image Height in pixel<br>4 byte: NOT USED<br>4 byte: NOT USED<br>4 byte: NOT USED<br>4 byte: NOT USED<br>IMAGE_SIZE byte: JPEG image data buffer |

**Request:**

```
0000   18 00 00 00 4e 62 00 00 ff ff ff ff 00 00 00 00
0010   00 00 00 00 1c 00 00 00 1f 27 00 00 04 00 00 00   Header quality (10015)
0020   50 00 00 00 55 27 00 00 08 00 00 00 40 01 00 00   Header resolution (10069)
0030   c8 00 00 00
```

**Answer:**

```
0000   18 00 00 00 4e 62 00 00 00 00 00 00 ff ff ff ff
0010   00 00 00 00 4c 1b 00 00 d1 07 00 00 0c 00 00 00   Header timestamp (2001)
       0020 14 05 db 07 09 00 07 00 2f 00 3e 00 d0 07 00 00   Header image (2000)
0030   2f 1b 00 00 00 00 00 00 00 00 00 00 28 00 00 00
0040   07 1b 00 00 40 01 00 00 c8 00 00 00 00 00 00 00
0050   08 00 00 00 80 f4 6d 9a 01 00 00 00 ff d8 ff db
0060   00 43 00 06 04 05 06 05 04 06 06 05 06 07 07 06
0070   08 0a 10 0a 0a 09 09 0a 14 0e 0f 0c 10 17 14 18
0080   18 17 14 16 16 1a 1d 25 1f 1a 1b 23 1c 16 16 20
0090   …
```

## 1.2 VRC Payload

The payload of information coming from or going to the camera is passed as the payload of

the protocol TOP, which may have zero length if the command does not need to have parameters.

The packaging format of the data is described below.

The payload contains a variable number of data blocks, called fields, each of which has the following structure:

| Field | Dimension | Notes |
|---|---|---|
| Field code | 4 bytes | "Field code" number |
| Data dimension | 4 bytes | Specifies the length of the field data, in bytes. |
| Data field | m bytes | The data field must contain any bytes of padding up to the size of the nearest multiple of 4 bytes. |

The fields are identified by a field code, the application interprets the data block must know the type of data into the field.

Moreover, according to the knowledge of the size of each field, the application can scan through all the fields from the first by simply moving a number of bytes necessary.

Each field code requires the presence of an associated payload size implied or inferred based on the following conventions:

1) All fields are always aligned 4byte.

2) The header of each field is composed of 2 fields 4byte each: the first field contains the code ('DATA_CODE' or 'EVENT_FIELD'), the second field contains the usable size of transported data, excluding the size of the header itself.

3) Navigation within a structured data block is advancing into the buffer size (aligned to the nearest higher multiple of 4 bytes) contained in the current field plus the size of the header.

4) The order of appearance of the fields within a data block is not influential. The application that performs the extraction should not be affected by the order in which fields were included in the buffer.

5). The fields of type 'LONG' fields are signed 4 bytes in size. The size indicated in the 'size' header field is therefore equal to 4.

6) The fields of type 'STRING' are byte sequences identified by at least one string terminator '\ 0'. Any character '\ 0' will be needed for the extra padding to the size nearest higher multiple of 4byte. The size inserted in the 'size' header is equal to the actual value of the length of the string plus 1, ie the output value of strlen () + 1 byte that identifies the '\ 0'). Ex: the string "test" will be encoded as follows:

* Usable size of the data: strlen ("test") +1 = 5.

* Field data: 't' 'e' 's' 't' '\ 0' '\ 0' '\ 0' '\ 0'.

7) The fields of type 'STRING FIXEDSIZE "are sequences of bytes identified by at least one string terminator' \ 0 ', where all the bytes in excess of the useful data, up to the fixed size, are set to' \ 0 ' .

8) The fields of type 'BUFFER' are fields of bytes containing generic data. The size inserted in the 'size' header is equal to the size of useful data. Any padding bytes are not counted.

9) The fields of type 'buffer' may contain structured data codes (use nested fields). The

inserted codes are described in the specific chapter.

10) Any fields within a data block (see Section 7) are generally unique, however, in some cases it is possible that a data block contains the same fields with the same code. Knowing which codes contain structured data in this way, you can extract all the necessary information. In this case it is necessary to perform a sequential search in the buffer.

10) Some fields are used for managing dynamic codes. These codes are represented by a start code numbering and code numbering of the end. The use of these fields is described in more detail when it comes to actual use.

# 2 Basics

Here below some features you need to know to optimize the operation of the device.

## 2.1 Date and Hour management.

Keep the device synchronized with a reference is necessary for the proper functioning of the machine and to be able to associate the events recorded with the real time in which they occurred.

Since the internal clock of the devices in general suffers a drift over time, the device has different methods of synchronization with respect to a reference server.

The device is designed for use GMT reference time for all its operations. In addition you can configure the use of daylight time and time zone to make it possible to convert the system time to local time. This information is configured on the device and made available to clients connected.

So when switching summer / winter time there are no discrepancies reported on the time, in addition the client is aware of the state summer / winter time and then can calculate the conversion from GMT to local time for any information recorded on the device.

The devices support the SNTP time synchronization. On some devices you can use synchronization via GPS, or via IEEE1588 protocol.

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

# 3 VRC commands

| Name | Command Code | Description | Supported devices |
|---|---|---|---|
| Get Image (obsolete) | 25008 | single generic image request | All Vega cameras |
| Output On | 25051 | To enable a digital output | All Vega cameras |
| Output Off | 25052 | To disable a digital output | All Vega cameras |
| Read Input/Output | 25134 | To read input/output status | All Vega cameras |
| No Operation | 25150 | Connection keep-alive | All Vega cameras |
| Get Single Image | 25166 | Single jpeg image request | All Vega cameras |
| Reset Device | 26000 | Device reboot | All Vega cameras |
| Read File | 26001 | To read file | All Vega cameras |
| Write File | 26002 | To write file | All Vega cameras |
| Delete File | 26003 | To delete file | All Vega cameras |
| Write Firmware (TSK) | 26004 | To write firmware on the camera (TSK file) | All Vega cameras |
| Read Firmware (TSK) | 26005 | To read firmware on the camera (TSK file) | All Vega cameras |
| Get state | 26006 | To read status od asinchronous operations (R/W TSK, FPGA, Installpack) | All Vega cameras |
| Write Date Time | 26010 | To write date and time | All Vega cameras |
| Read Date Time | 26011 | To read date and time | All Vega cameras |
| Write FPGA | 26020 | To write FPGA | All Vega cameras |
| Write Installpack | 26028 | To write install pack | All Vega cameras |
| Reload List A | 35000 | To charge in RAM list A | All Vega cameras |
| Reload List B | 35001 | To charge in RAM list B | All Vega cameras |
| Read List A | 35002 | To read list A | All Vega cameras |
| Read List B | 35003 | To read list B | All Vega cameras |
| Engine Start | 35004 | To enable engine | All Vega cameras |
| Engine Stop | 35005 | To disable engine | All Vega cameras |
| Read Error Counters | 35006 | To read error counters | All Vega cameras |
| Reset Error Counters | 35007 | To reset error counters | All Vega cameras |
| Trigger Start | 35008 | Start ethernet trigger | All Vega cameras |
| Trigger Stop | 35009 | Stop ethernet trigger | All Vega cameras |
| Vega Device Info | 35010 | Device information request | All Vega cameras |
| Format SD | 35016 | To format SD card | All Vega cameras |
| Get SD Information | 35017 | SD information request | All Vega cameras |
| Reset Transit data | 35022 | Delete transits buffer (free run mode) | All Vega cameras |
| Get Image Date Time | 36000 | To request an image with a certain timestamp | Vega Network Ctx |

# 4 Commands description

## 4.1 Image acquisition (25008) -- OBSOLETE

**Description:**

With this command you can capture an image. The output format is device-dependent and is specified in Tbanco header that is sent with the image.

Commands to use with last version of vega firmware are:

- 25166: to extract a jpeg image with quality and resolution desired.
- 36000: to extract a jpeg image acquired at a specific timestamp.

On some models Vega are two cameras, BW and colour. It's possible to select which of the two cameras to capture the image specifying the camera ID into Receiver field of CommandHeader:

- 0: Channel BW.
- 1: Colour Channel.

**Input:**

None.

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| IMAGE_TIME | 2001 | datetime | Plate acquisition timestamp (local time). |
| IMAGE | 2000 | banco | Acquired image |

**Notes:**

In the image field, the first 40 bytes correspond to the structure type header "banco" as defined in the following paragraphs.

The use of this message stops the LPR process; it will resume automatically after the image has been sent.

## 4.2 Digital input/output status reading (25134)

**Description:**

The command requests the device status of the inputs and outputs. It is returned the physical state of the inputs and outputs.

**Input :**

None

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| DIGITAL_INPUT_NUMBER | 12041 | long | Number of digital input on the camera |

| | | | |
|---|---|---|---|
| DIGITAL_INPUT_STATUS | 12043 | long | Bit field that identifies the physical state of the digital inputs. The input 0 is the least significant bit (0-31). |
| DIGITAL_OUTPUT_NUMBER | 12042 | long | Number of digital output on the camera |
| DIGITAL_OUTPUT_STATUS | 12044 | long | Bit field that identifies the physical state of the digital inputs. The output 0 is the least significant bit (0-31). |

## 4.3 Digital Output ON (25051)

**Description:**

The command activates a digital output

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| DIGITAL_OUTPUT_SEL_NUMBER | 12062 | long | Number of the output to activate |

**Output:**

None

**Notes:**

The parameter "DIGITAL_OUTPUT_SEL_NUMBER" can assume a value greater than 0 until NUM_OUTPUT-1, where NUM_OUTPUT is the number of digital outputs on the unit. This value can be retrieved with the message read status of the inputs and outputs (command 25134).

## 4.4 Digital output OFF (25052)

**Description:**

The command deactivates a digital output

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| DIGITAL_OUTPUT_SEL_NUMBER | 12062 | long | Number of the output to deactivate |

**Output:**

None

**Notes:**

The parameter "DIGITAL_OUTPUT_SEL_NUMBER" can assume a value greater than 0 until NUM_OUTPUT-1, where NUM_OUTPUT is the number of digital outputs on the unit. This value can be retrieved with the message read status of the inputs and outputs (command 25134).

## 4.5 No operation (25150)

The command allows to keep active TCP or UDP connection.

The camera is expected to receive a command within 60s of this type (TCP) or 20s (UDP) from the last command received. If the condition is not met, the socket communication is closed.

**Input:**

None.

**Output:**

None.

**Note:**

- no answer from this command.

## 4.6 Get single image (25166)

**Description:**

With this command you can capture an image in JPG format.

The image size must be specified in the data message sent to the VRC server will also be less or equal than the resolution of the CCD camera. In the message must also specify the desired JPEG quality.

On some models Vega are two cameras, a black and white and color. It's possible to select which of the two cameras to use by specifying the camera ID into Receiver field of CommandHeader:

- 0: BW channel.
- 1: Color channel.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| JPEG_QUALITY | 10015 | long | Image compression quality required (0..100) |
| POINT | 10069 | point | Resolution required (width, height) |

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| IMAGE_TIME | 2001 | datetime | Acquiring time-stamp of the plate (ora locale). |
| IMAGE | 2000 | banco | Image data buffer |

**Note:**

- Into the IMAGE field, first 40 bytes correspond to an header structure called "banco" explained in following sections.
- This command stops the reading process, that will be restart after image sending 

## 4.7 Device Reboot (26000)

**Description:**

this command reboots the camera after a right shut down, if necessary.

This is an asynchronous command: after command sending it needs to wait for the answer. It's possible to start get_state() command sequence to determine when the camera is rebooted.

The camera waits up to 110 seconds; during this time it close running services. After this time

a reboot is forced.

**Input:**
None
**Output:**
None

## 4.8 Read file (26001)

**Description:**
Command to read a setting file into device reserved memory.
**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| FILE_NAME | 12038 | String | File name |

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| FILE_DATA | 12039 | buffer | File contents |

**Note:**

- CAUTION: the permission to read file without control could compromise system safety.

Damage caused by incorrect configurations can be:
a) inaccessibility of the system via Ethernet.
b) continuous rebooting of the device.
c) Failure to register or failure to report alarms.
d) inability to perform any operation on the device.
The system recovery is usually deleting all the files from the device and start from a clean situation.

## 4.9 Write file (26002)

**Description:**
Command to write a file into device reserved memory.
**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| FILE_NAME | 12038 | string | Filename, max 15 characters. |
| FILE_DATA | 12039 | buffer | File contents |

**Output:**
None

**Note:**

- CAUTION: wrong writing of these very important files can compromise system working.

The complete filling of this reserved memory space can compromise system working.

The damage can be caused by incorrect configurations:

a) inaccessibility of the system via Ethernet.

b) continuous device rebooting.

c) Failure to register or failure to report alarms.

d) inability to perform any operation on the device.

The system recovery is usually deleting all the files from the device and start from a clean situation.

The file name length must not exceed 15 characters.

directory management is not supported.

## 4.10 Delete file (26003)

**Description:**

command to delete a file into device reserved memory.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| FILE_NAME | 12038 | string | File name |

**Output:**

None

**Note:**

- CAUTION: wrong deleting of these very important files can compromise system working.

The damage can be caused by incorrect configurations:

a) inaccessibility of the system via Ethernet.

b) continuous device rebooting.

c) Failure to register or failure to report alarms.

d) inability to perform any operation on the device.

The system recovery is usually deleting all the files from the device and start from a clean situation.

If the file is not present, no error occurs.

## 4.11 Write Firmware-TSK(26004)

**Description:**

command to upgrade the device with a .TSK file provided as input parameter.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| TSK | 12034 | buffer | Buffer of "TSK" file to install on the device. |

**Output:**

None.

**Note:**

- CAUTION: upgrading with an incorrect or not compatible TSK file can produce these damages:

a) inaccessibility of the system via Ethernet.

b) continuous device rebooting.

c) inability to perform any operation on the device.

System recovery must be made by switch on and at the same time putting the camera in "Linker" mode. Sometimes could be necessary to give back the device.

## 4.12 Read Firmware-TSK(26005)

**Description:**

Command to read TSK from device.

**Input:**

None.

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| TSK | 12034 | buffer | Buffer of "TSK" file read from device |

## 4.13 Get device status (26006)

**Description:**

Command to read the completion status of some VRC command. VRC commands you can check status are:

- Write tsk (26004)
- Write fpga (26020)
- Write installpack (26028)

**Input:**

None.

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| DEVICE_STATUS | 12033 | long | Completion command status:<br>0: ready |

| | | | |
|---|---|---|---|
| | | | 1: busy<br>2: error |
| COMPLETION_STATUS | 10082 | long | Completion level (0...1000) |

## 4.14 Write date and time (26010)

**Description:**

command to configure date, time, GMT offset and daylight saving (DST) time management.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| TIME | 12026 | datetime | Current date and time (GMT) |
| TIME_ZONE | 10037 | long | GMT offset in minutes |
| VEGA_DAYLIGHT_AUTO | 13210 | long | Enable (1) or Disable (0) automatic daylight saving time management. |

**Output:**

None.

## 4.15 Read date and time (26011)

**Description:**

command to read current date and time, GMT offset and DST status on the device.

**Input:**

None.

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| TIME | 12026 | datetime | Current device date and time: GMT if a time server synchronisation is active; in this case following parameter allow to calculate device local time. |
| TIME_ZONE | 10037 | long | GMT offset in minutes |
| TIME_DST_VALUE | 10041 | long | DST status (0: disabled, 1: enabled) |

**Note:**

- It's advised to use a time server synchronisation in order to always maintain the correct time..
- The camera internally works with GMT time to avoid issue due to DST changes.
- To determine actual time you need to add GMT time and GMT offset; if DST value is 1 it's necessary to add 60 minutes too.

## 4.16 Write FPGA (26020)

**Description:**

command to upload FPGA firmware.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| FILE_NAME | 12038 | string | FPGA file name.<br>"fpga" and "rbf" formats are supported. |
| BOARD_CODE | 12013 | long | Board code, depending on the hardware mounted on the device:<br>243: Vega<br>518: Vega 2H<br>635: Vega HD / Vega 2HD<br>640: Vega III |
| FPGA | 12406 | buffer | FPGA data buffer. |

**Output:**

None.

**Note:**

- CAUTION: upgrading with an incorrect or not compatible FPGA file can produce these damages:

a) continuous device rebooting.

b) inability to perform any operation on the device.

c) in extreme case, it could be necessary to give back the device.

## 4.17 Write Installpack (26028)

**Description:**

Command to upload an install pack on the device.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| SUB_OPTIONS | 12067 | long | Upload type:<br>0: upload with keeping current settings.<br>1: upload with default settings<br>2: upload with keeping current settings and FPGA upgrade (with a correct install pack with FPGA). |
| INSTALLPACK | 10087 | buffer | Install pack data buffer to upload. |

**Output:**

None.

**Note:**

- CAUTION: upgrading with an incorrect or not compatible install pack can produce these damages:

a) continuous device rebooting.

b) inability to perform any operation on the device.

c) in extreme case, it could be necessary to give back the device.

## 4.18 Reload list A (35000)

**Description:**

Command to reload authorization file list "A" into internal camera.

In order to permit this command, list management must be enabled (see User Guide of the camera). If list management is not enabled, VRC server answers with an error.

**Input:**

None.

**Output:**

None.

## 4.19 Reload list B (35001)

**Description:**

Command to reload authorization file list "A" into internal camera.

In order to permit this command, list management must be enabled (see User Guide of the camera). If list management is not enabled, VRC server answers with an error.

**Input:**

None.

**Output:**

None.

## 4.20 Read file list A (35002)

**Description:**

Command to read authorization file list "A" into internal camera.

In order to permit this command, list management must be enabled (see User Guide of the camera). If list management is not enabled, VRC server answers with an error.

**Input:**

None.

**Output:**

buffer

| Name | Value | Type | Description |
|---|---|---|---|
| NUM_PLATE_IN_LIST | 13208 | long | Number of plates into the list |
| PLATE_LIST_STRING | 13209 | string | License plates list string |

**Note:**

- License plates list string is formatted as follows:

*License Plate;Country;Notes*

Example:

ZF333CX;ITA;Auto rossa test1
GF345LL;ITA;Auto test2
DD876BU;ITA;Auto test3

## 4.21 Read file list B (35003)

**Description:**

Command to read authorization file list "B" into internal camera.

In order to permit this command, list management must be enabled (see User Guide of the camera). If list management is not enabled, VRC server answers with an error.

**Input:**

None.

**Output:**

buffer

| Name | Value | Type | Description |
|---|---|---|---|
| NUM_PLATE_IN_LIST | 13208 | long | Number of plates into the list |
| PLATE_LIST_STRING | 13209 | string | License plates list string |

**Note:**

- See command 35002.

## 4.22 Engine Start (35004)

**Description:**

Command to enable the engine. This command is equivalent to *Enable Engine* into General Settings web page of the camera. If the engine is already enabled, this command has no effect and VRC server doesn't send any error message (error code = 0).

**Input:**

None.

**Output:**

None.

## 4.23 Engine Stop (35005)

**Description:**

Command to disable the engine. This command is equivalent to *Enable Engine* into General Settings web page of the camera. If the engine is already disabled, this command has no effect and VRC server doesn't send any error message (error code = 0).

**Input:**
None.
**Output:**
None.

## 4.24 Lettura dei contatori di errori (35006)

**Description:**
The camera, during normal working, traces some kinds of error, in order to check if some problem occurs.
**Input:**
None.

**Output:**

| Name | Value | Type | Description |
|---|---|---|---|
| ERR_FTP_IMAGE | 13200 | long | Number of errors of FTP image transfer from camera to FTP server. |
| ERR_FTP_DB | 13201 | long | Number of errors of FTP csv transfer from camera to FTP server. |
| ERR_PNS_ALARM | 13202 | long | Number of errors of message sending to Gate Master server. |
| ERR_FTP_LOG | 13203 | long | Number of errors of log sending. |
| ERR_IMAGE_ACQUISITION | 13204 | long | Number of errors of image acquisition. |
| ERR_SNTP_SYNC | 13205 | long | Number of errors of time server synchronisation |
| ERR_TCP_MSG | 13212 | long | Number of errors of TCP message sending |

## 4.25 Error counters reset (35007)

**Description:**
Command to reset counters error.
**Input:**
None.
**Output:**
None.

## 4.26 Trigger Start (35008)

**Description:**
This message is usable only if the device is set to operate in "TRIGGER_ETHERNET" (see programming guide section on web servers in general settings). If this message is sent with the camera in the other mode, VRC server responds with an error. In the message sent must always be the string TriggerIdString. This string does not require a particular format, can be used as the unique identifier of the trigger if, for example, is constructed using current time

and date of the system VRC client. The string can be used as TriggerIdString TAG programming of actions performed by the firmware (see manual web programming).

When the camera receives this command, it begins to process images until the end of Gate time. When Gate time is expired, the camera generate the transit. If the camera receive "Trigger Stop" message before Gate time ending, Gate time will be ignored and the camera generates the transit.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| NET_TRIG_ID_STRING | 13206 | string | Trigger identification string |
| NET_TRIG_GATE_TIME | 13207 | long | Gate time |

**Output:**

None.

**Note:**

- If the camera is in WEB live mode (Camera OCR page), Trigger start message will be ignored and VRC server will answer with error.

## 4.27 Trigger Stop (35009)

**Description:**

This message is usable only if the device is set to operate in "TRIGGER_ETHERNET" (see programming guide section on web servers in general settings). If this message is sent with the camera in the other mode, VRC server responds with an error.

When the camera receives this command, it stops to process images and generates the transit.

This message needs the same TriggerIdString used on related Trigger Start message.

**Input:**

| Name | Value | Type | Description |
|---|---|---|---|
| NET_TRIG_ID_STRING | 13206 | string | Trigger identification string |

**Output:**

None.

**Note:**

- If the camera is in WEB live mode (Camera OCR page), Trigger start message will be ignored and VRC server will answer with error.

## 4.28 Vega Device Info (35010)

This command is to obtain basic device information.

**Input:**

None

**Output:**

| Name | Value | Type | Note |
|---|---|---|---|

| | | | |
|---|---|---|---|
| SERIAL_NUMBER | 12000 | long | Device main board serial number |
| BOARD_CODE | 12013 | long | Hardware main board code |
| BOARD_REVISION | 12014 | long | Main board revision |
| TEMPERATURE | 12019 | long | Temperature tenths of Celsius degrees) |
| FPGA_VERSION | 12020 | long | FPGA version |
| DESCRIPTION | 12010 | string | Firmware description |
| NUM_HW_CHANNEL | 12040 | long | Number of video channel (1=single head camera, 2=double head camera) |

**Note:**

- Different firmware version could provide more fields, depending on technical features of different devices.

## 4.29 Secure Digital format (35016)

**Description:**

This command, if supported, permits to format of all secure digital partitions, independently on the number of SD cards mounted on the device.

**Input:**

None

**Output:**

None

## 4.30 Secure Digital information request (35017)

**Description:**

This command is to obtain basic information of secure digital card buffering partition.

Buffering partition (see user manual) is useful to store network events not managed because the network connection is off.

**Input:**

None

**Output:**

| Name | Value | Type | Note |
|---|---|---|---|
| RCB_TOTAL_SIZE_KB | 13213 | long | SD buffering memory total size (KB) |
| RCB_USED_SIZE_KB | 13214 | long | SD buffering memory used size (KB) |
| RCB_NUM_STORED_EVENTS | 13215 | long | Number of stored events |

**Note:**

- In case of server error, it's possible to have no answer fields.

## 4.31 Last transits buffer reset (35022)

**Description:**

With this command, lists off interior license plate actually read are empty. In this way any license plate list of transits already generated will be deleted, making the device ready to generate any new transit even if the license plate read is the same as previous readings.

This command can be used in free-run mode to "trigger" asynchronously the camera.

**Input:**

None.

**Output:**

None

**Note:**

- This command is useful especially if "Enable Multi Out Same Plate" is 0.

## 4.32 Image request an exact timestamp (36000)

**Description:**

Command to get an image from the device at a specified timestamp.

**Input:**

| Codice | Value | Type | Note |
|---|---|---|---|
| VEGA_CTX_IMG_TIME_PARAM | 13501 | ImgParam | Acquisition image parameters |

**Output:**

| Codice | Value | Type | Note |
|---|---|---|---|
| TIME | 12026 | time | Real image timestamp |
| VEGA_CTX_IMG_RAW | 13500 | bancoTx | Request image |

**Note:**

- In case of server error, it's possible to have no answer fields.
- **This command is for Vega Network Context only**

# 5 Field codes

List of field codes used into VRC commands.

| Name | Value | Type | Description |
|---|---|---|---|
| CODE_NULL | 0 | - | - |
| IMAGE | 2000 | buffer | |
| IMAGE_TIME | 2001 | datetime | |
| JPEG_QUALITY | 10015 | long | |

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

| Name | Value | Type | Description |
|---|---|---|---|
| GMT_OFFSET_MINUTES | 10037 | long | |
| DAYLIGHT_STATUS | 10041 | long | |
| POINT | 10069 | point | |
| CRC_TSK | 10088 | long | |
| CRC_PLATE_READER | 10089 | long | |
| PLATE_READER | 10090 | buffer | |
| SERIAL_NUMBER | 12000 | long | |
| DESCRIPTION | 12010 | string | |
| BOARD_CODE | 12013 | long | |
| BOARD_REVISION | 12014 | long | |
| TEMPERATURE | 12019 | long | |
| FPGA_VERSION | 12020 | long | |
| TIME | 12026 | datetime | |
| TSK | 12034 | buffer | |
| FILE_NAME | 12038 | string | |
| FILE_DATA | 12039 | buffer | |
| NUM_HW_CHANNEL | 12040 | long | |
| DIGITAL_INPUT_NUMBER | 12041 | long | |
| DIGITAL_OUTPUT_NUMBER | 12042 | long | |
| DIGITAL_INPUT_STATUS | 12043 | long | |
| DIGITAL_OUTPUT_STATUS | 12044 | long | |
| DIGITAL_OUTPUT_SEL_NUMBER | 12062 | long | |
| ERR_FTP_IMAGE | 13200 | long | |
| ERR_FTP_DB | 13201 | long | |
| ERR_PNS_ALARM | 13202 | long | |
| ERR_FTP_LOG | 13203 | long | |
| ERR_IMAGE_ACQUISITION | 13204 | long | |
| ERR_SNTP_SYNC | 13205 | long | |
| NET_TRIG_ID_STRING | 13206 | string | Max 32 characters |
| NET_TRIG_GATE_TIME | 13207 | long | |
| NUM_PLATE_IN_LIST | 13208 | long | |
| PLATE_LIST_STRING | 13209 | string | |
| DAYLIGHT_SAVING_TIME_AUTO | 13210 | long | |

| Name | Value | Type | Description |
|---|---|---|---|
| VEGA_CTX_IMG_RAW | 13500 | bancoTx | |
| VEGA_CTX_IMG_TIME_PARAM | 13501 | imgParam | |

# 6Appendix A: data types

Following will be described all data types used into VRC protocol.

## *6.1.1 Data types*

Data types:

- **long**: 32bit signed integer, little endian.
- **string**: 8bit ASCII string (sequence of characters), with terminator '\0'.
- **datetime**: 12byte data structure containing date and time as follows (single multi-byte fields are little endian):

| Field | Type | Value |
|---|---|---|
| Day | byte | 1..31 |
| Month | byte | 1..12 |
| Year | short (2byte) | |
| Hour | short (2byte) | 0..23 |
| Minute | short (2byte) | 0..59 |
| Second | short (2byte) | 0..59 |
| Millisecond | short (2byte) | 0..999 |

- **point:** 8 byte data structure containing 2 integer values:

| Field | Type | Value |
|---|---|---|
| X Coordinate | long (4byte) | >0 |
| Y Coordinate | long (4byte) | >0 |

- **buffer**: byte sequence with specific decoding based on data contained

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

- **banco**: data buffer with first 40 bytes which contain following header and next bytes contain desired image buffer.

| Field | Type | Value |
|---|---|---|
| Reserved | long (4byte) | - |
| Reserved | long (4byte) | - |
| Header size | long (4byte) | 40 |
| Useful data dimension | long (4byte) | Image data dimension |
| X Resolution | long (4byte) | X image resolution |
| Y Resolution | long (4byte) | Y image resolution |
| Image type | long (4byte) | |
| Bit per pixel | long (4byte) | |
| Reserved | long (4byte) | |
| Image format | long (4byte) | 0: Raw image<br>1: Jpeg image |

- **imgParam**: 16 byte data buffer made as follows:

| Field | Type | Value |
|---|---|---|
| TimeStamp | datetime | Requested Image Timestamp (GMT) |
| DeltaMs | long (4byte) | Maximum accepted image timestamp deviation (ms) |

- **bancoTx**: 52 byte data buffer made as follows:

| Field | Type | Value |
|---|---|---|
| Gain | long (4 byte) | Gain |
| Shutter | long (4 byte) | Shutter time (µs) |
| MosaicPhase | long | Phase (for Mosaic raw images only) |

| | | |
|---|---|---|
| | (4 byte) | |
| banco | Banco (40byte) | Banco structure as described before |

# 7 Examples

### Engine start (35004)

Following wire-shark grabbing after sending "Engine Start".

18:00:00:00:bc:88:00:00:ff:ff:ff:ff:ff:ff:ff:ff:00:00:00:00:00:00:00:00

| Nr of bytes | Description | Value | Value to send |
|---|---|---|---|
| 4 | Header Dimension | 24 | 18:00:00:00 |
| 4 | Command Code | 35004 | BC:88:00:00 |
| 4 | Sender | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Receiver | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Error | 0 | 00:00:00:00 |
| 4 | Data Dimension | 0 | 00:00:00:00 |

### Read file

To change some configuration parameter, it's necessary to modify files with extensione .ini.
To make this you have to use the specific command to read file (26001) and to write file (26002).

For ex, we launch "device.ini" file (760 byte) reading.

| Nr of bytes | Description | Value | Value to send |
|---|---|---|---|
| 4 | Header Dimension | 24 | 18:00:00:00 |
| 4 | Command Code | 26001 | 91:65:00:00 |
| 4 | Sender | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Receiver | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Error | 0 | 00:00:00:00 |
| 4 | Data Dimension | 20 | 14:00:00:00 |
| 4 | Data Code (filename to read) | 12038 | 06:2f:00:00 |
| 4 | Data Size (file name length) | 11 | 0B:00:00:00 |

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

| | | | |
|---|---|---|---|
| 12 (real size is 11 becomes 12 with zero-padding 4 bytes alignment) | Data Buffer | device.ini | 64:65:76:69:63:65:2e:69:6e:69:00:00 |

The answer from the device will be:

| Nr of bytes | Description | Value | Value to send |
|---|---|---|---|
| 4 | Header Dimension | 24 | 18:00:00:00 |
| 4 | Command Code | 26001 | 91:65:00:00 |
| 4 | Sender | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Receiver | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Error | 0 | 00:00:00:00 |
| 4 | Data Dimension | 760 | F8:02:00:00 |
| 4 | Data Code (data of file code) | 12039 | 07:2f:00:00 |
| 4 | Data Size (file length) | 752 | F0:02:00:00 |
| 752 | Data Buffer | device.ini data | ... |

### Write File

Writing of a file (device.ini -- 780 byte), no answers (see 26002 command).

| Nr of bytes | Description | Value | Value to send |
|---|---|---|---|
| 4 | Header Dimension | 24 | 18:00:00:00 |
| 4 | Command Code | 26002 | 92:65:00:00 |
| 4 | Sender | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Receiver | 0xFFFFFFFF | FF:FF:FF:FF |
| 4 | Error | 0 | 00:00:00:00 |
| 4 | Data Dimension | 760 | 0C:02:00:00 |
| 4 | Data Code (file name to write) | 12038 | 06:2f:00:00 |
| 4 | Data Size (file name length) | 11 | 0B:00:00:00 |
| 12 (real size is 11 becomes 12 with zero-padding 4 bytes alignment) | Data Buffer | device.ini | 64:65:76:69:63:65:2e:69:6e:69:00:00 |
| 4 | Data Code (data | 12039 | 07:2F:00:00 |

| | | | |
|---|---|---|---|
| | contained in the file) | | |
| 4 | Data Size | 752 | F0:02:00:00 |
| 752 | Data buffer | | ... |

# 8 Appendix B: configuration files

CAUTION: wrong change or incorrect key value into configuration file can produce unpredictable damages and compromise normal working of device.

Configuration files contain the configuration of the machine that is read at startup of the device. Any changes to these files usually require a reset of the device.

The main configuration file is Device.ini, that contains basic information regarding your network configuration.

The content of files depends on the firmware running on the device, so some sections may be missing or contain a number of different keys.

The client performing a read / write ini files must leave unchanged all the sections and keys that can not handle, whether it has knowledge of the Description of the keys themselves, and who has not.

## 8.1 Device.ini file

This file contains network parameters. The correct configuration of these parameters is a prerequisite for the accessibility of the device.

Depending on the type and number of network interfaces the file will contain one or more sections.

**[Device Parameters]**
**NetBiosName=POINTER**
**KeepAliveTime=7200000**
**KeepAliveInterval=1000**
**KeepAliveCounterClose=3600**

**[Network XXXXXXXXXXX]**
**IP=172.25.6.112**
**Netmask=255.255.0.0**
**Gateway=172.25.0.254**
**DNS=0.0.0.0**
**WINS=0.0.0.0**
**TimeServer=172.25.0.241**
**DHCP=0**
**DHCP wait save=5**
**DHCP IPAutoConfiguration=0**
**MTU = 1500**
**Speed = 0**

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

**Mode = full**

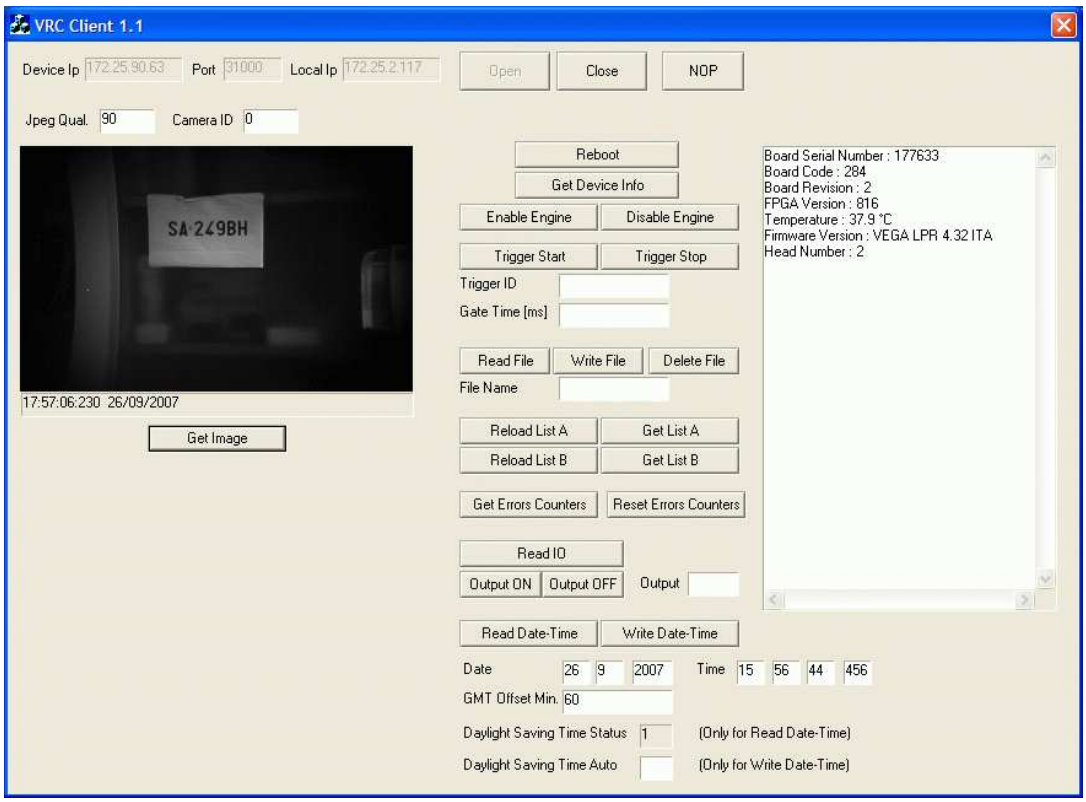
Useful keys:

- Netbiosname: device net bios name. If not present Vega sw will set "SCL".
- IP: device IP address
- Netmask: subnet mask.
- Time server: time server IP address.
- Gateway: network gateway IP address.

# 9 Appendix C: VRC client example

## 9.1 VRC Client developed in Visual C++ 6.0

To simplify and help the testing and development of a client application, it was developed VRC on Windows VC + + 6.0. The VRC Client implements all features described in this document. Tattile provides source code of this project.




CommandCode and DataCode described into this document can be found in VRC-Codes.h of the example project provided.

**Tattile s.r.l.**
Via G. Donizetti 1
25030 Mairano (BS)-Italia
Tel. +39.030.9700.0
Fax. +39.030.9700.1
Web: www.tattile.com

### 9.2 VRC client using Tattile DLL

Tattile provides some DLLs to simplify the development of VRC client on Windows. For detailed examples of using the functions refer to the source code of the Client Project VRC sample.

The functionality features enable you to:

- Manage the TCP connection to server VRC
- Send and receive data
- Build the package of data to send respecting the structure DataHeder
- Look in the data packet received the requested data based on DataCode

| TCP connection management |
|---|
| itf_interface.h |
| HANDLE itf_Open(ITF_OpenParameter *op, long timeOut);<br>long itf_Close(HANDLE *chHandle);<br>long itf_SetParameter(HANDLE chHandle, long code, const void *value, long size);<br>long itf_ChannelStatus(HANDLE chHandle, long timeOut); |

| Send and receive data |
|---|
| itf_SecurCam.h |
| long SC_send_command(HANDLE chHandle, long command, long wait_answer);<br>long SC_send_long(HANDLE chHandle, long command, long value, long wait_answer);<br>long SC_send_buffer(HANDLE chHandle, long command, void *pUserData, long UserDataSize, long wait_answer);<br>long SC_receive_long(HANDLE chHandle, long command, long *value);<br>long SC_receive_buffer(HANDLE chHandle,long command,void **pUserData, long *UserDataSize);<br>long SC_send_receive_buffer(HANDLE chHandle,long command,void *buffer_tx,long buffer_tx_size,void **buffer_rx, long *buffer_rx_size); |

| Build data to send |
|---|
| itf_SecurCam.h |
| long SC_add_string(void **pBuffer, long *BufferSize, long *BufferSizeUsed, const TCHAR *str, long code, BOOL AllowRealloc);<br>long SC_add_long(void **pBuffer, long *BufferSize, long *BufferSizeUsed, long value, long code, BOOL AllowRealloc);<br>long SC_add_buffer(void **pBuffer, long *BufferSize, long *BufferSizeUsed, void *pdata, long datasize, long code, BOOL AllowRealloc); |

| Extract data received |
| --- |
| itf_SecurCam.h |
| long SC_extract_long(long *value, ULONG code, void *pData, long DataSize);<br>long SC_extract_buffer(void *pBuffer, long BufferSize, long *DataWritten, long code, void *pData, long DataSize);<br>long SC_extract_string(TCHAR *pStr, long StrSize, long *DataWritten, ULONG code, void *pData, long DataSize);<br>long SC_extract_image(Tbanco **pbanco, ULONG code, void *pData, long DataSize); |

All functions, except itf_Open(), returns 0 if success and non-zero value in case of error.

# 10 Appendix D: reference documentation

- "RFC 791 – Internet protocol"
- "RFC 793 – Transmission control protocol"