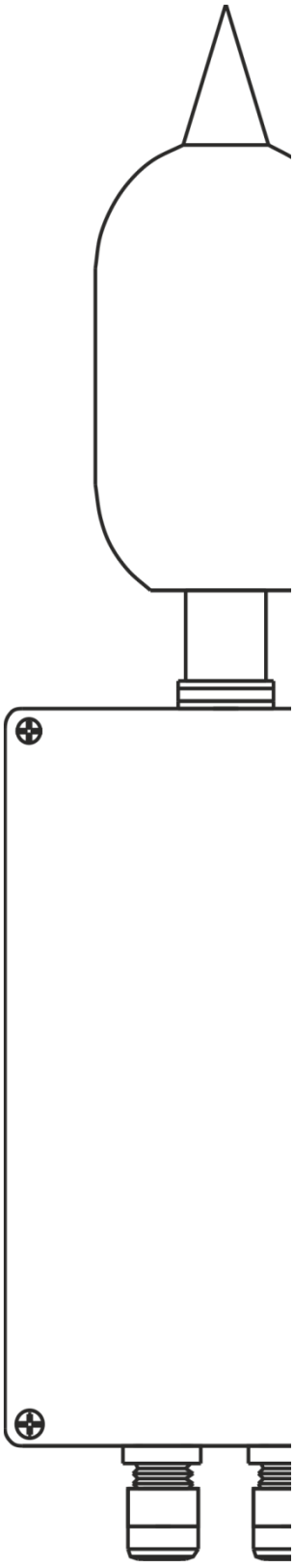
CESVA

TA120

PROTOCOLS

PR_TA120_v0041_20171220_EN



CONTENTS

HTTP / HTTPS PROTOCOLS 1

1.1 Sensor setting: CESVA Sensor Manager

With the **CESVA Sensor Manager** software, you can configure the follow parameters:

▪ Protocol	Select the protocol (UL2.0, Sentilo or CESVA)
▪ Security	Select your server security (HTTP or HTTPS)
▪ Host	Name of your API domain
▪ Port	Number of server port
▪ Linkdata	Name of your API path
▪ Token	Value of your API Key
▪ TLeq	Select the integration time T
▪ Overload/Underrange	Activate or deactivate to send Overload and Underrange
▪ LAeq1s	Activate or deactivate to send LAeq1s registers

For more information, please, see the manual of the **CESVA Sensor Manager** software.

It is very important to check, with **CESVA Sensor Manager** software, that the sensor is programmed correctly.

If you have selected the “UltraLight 2.0” protocol, see **chapter 1.2**, but if you have selected “Sentilo”protocol, see **chapter 1.3**.

NOTE: By default, the Overload/Underrange and LAeq1s are activated.

1.2 UltraLight 2.0 (UL2.0)

1.2.1 Sending data

Protocol description:

Format	Protocol	Method
UL2.0 DEVICE API	HTTP or HTTPS	POST

Header description:

```
POST/<link_data>?k=<tkn_value>&i=<id_value>&t=<time_value>&getCmd=<gc_value>
HTTP/1.1
Host: <host_name>
Content-Type: text/plain; charset=UTF-8
```

URL	Name		Description
	link_data		Server API path (*)
	host_name		Server API domain name (*)
Parameters	**Name**	**Value**	**Description**
	k	tkn_value	server API Key (*)
	i	id_value	Sensor identifier: TA120-Txxxxxx (where *Txxxxxx* is the serial number)
	t	time_value	Timestamp (Zulu format: *yyyy-mm-dd*T*hh:ii:ss*Z)
	getCmd	gc_value	request for receive commands (boolean: 1 = yes -default-, 0 = no)

*: Programmable by “CESVA Sensor Manager” software

Body description:

```
<par1_name>|<par1_value>|<par2_name>|<par2_value>|...
```

Parameters	Name	Value	Description
	n	n_val	Sound pressure level (L_{Aeq})
	o	o_val	Overload (boolean: 1 = true, 0 = false) (**)
	u	u_val	Underrange (boolean: 1 = true, 0 = false) (**)
	b	b_val	Optional module BA120 - Battery level (%)
	p	p_val	Optional module BA120 - Power supply status (boolean: 1 = true, 0 = false)
	w	w_val	Optional module WF120 - Wi-fi strength (%)
	m	m_val	Optional module MR120 - Modem strength (%)
	s	s_val	LAeq1s registers (***)

**: If “Send Overload/Underrange” is activated (see the manual of the “CESVA Sensor Manager” software)
***: If “Send LAeq1s” is activated (see the manual of the “CESVA Sensor Manager” software)

1.2.2 Sensor configurations

Ethernet interface with battery (BA120)

```
n|<n_val>|o|<o_val>|u|<u_val>|b|<b_val>|p|<p_val>|s|<s_val>
```

Ethernet interface without battery

```
n|<n_val>|o|<o_val>|u|<u_val>|s|<s_val>
```

Wi-Fi interface (WF120) with battery (BA120)

```
n|<n_val>|o|<o_val>|u|<u_val>|b|<b_val>|p|<p_val>|w|<w_val>|s|<s_val>
```

Wi-Fi interface (WF120) without battery

```
n|<n_val>|o|<o_val>|u|<u_val>|w|<w_val>|s|<s_val>
```

Modem 3G interface (MR120) with battery (BA120)

```
n|<n_val>|o|<o_val>|u|<u_val>|b|<b_val>|p|<p_val>|m|<m_val>|s|<s_val>
```

Modem 3G interface (MR120) without battery

```
n|<n_val>|o|<o_val>|u|<u_val>|m|<m_val>|s|<s_val>
```

1.2.3 Response data

The server will respond with an HTTP/HTTPS status code “200” if the request was accepted and processed correctly. Other code different to “200”, the sensor will consider a server’s error.

If **there is a command**, the server also will respond in the server body:

```
TA120-<id_value>@setConfig|<cmd_name>=<cmd_value>
```

Parameter	Description
id_value	Sensor identifier: Txxxxxx (where *Txxxxxx* is the serial number)
cmd_name	Command name
cmd_value	Command value

The TA120 accepts the follow commands:

Name	Values	Format	Description
t	0010 to 3600	Number (4 digits)	Averaging time parameter in seconds (NOTE: default is 0060)
onlylevel	1 or 0	Boolean	Deactivate (1) or activate (0) the “Overload/Underrange” option
seconds	1 or 0	Boolean	Activate (1) or deactivate (0) the “LAeq1s” option

1.2.4 Example

- Optional modules: BA120 (battery) and MR120 (modem)
- Host name: yourserver.com
- Link data: sensor/file
- Token: abcdefgh
- ID sensor: T123456

The sensor data:

```
POST /sensor/file?k=abcdefgh&i=TA120-T123456&t=2015-06-10T14:12:14Z&getCmd=1
HTTP/1.1
Host: yourserver.com
Content-Type: None
Connection: close

n|041.5|o|1|u|0|b|56|p|1|m|45|s|
046.6,0,0;048.4,0,0;047.4,0,0;043.3,0,0;039.9,0,0;039.8,0,0;039.4,0,0;040.5,0,0;040
.4,0,0;040.4,0,0;040.8,0,0;040.1,0,0;040.2,0,0;040.1,0,0;039.7,0,0;040.3,0,0;039.9,
0,0;040.1,0,0;040.1,0,0;039.8,0,0;040.0,0,0;040.1,0,0;041.0,0,0;045.3,0,0;044.4,0,0
;040.1,0,0;040.0,0,0;040.0,0,0;040.0,0,0;040.0,0,0;039.5,0,0;039.9,0,0;040.1,0,0;03
9.9,0,0;040.4,0,0;040.5,0,0;040.5,0,0;040.4,0,0;041.6,0,0;041.5,0,0;044.2,0,0;040.3
,0,0;039.9,0,0;039.9,0,0;040.1,0,0;039.9,0,0;039.9,0,0;039.5,0,0;040.1,0,0;040.0,0,
0;039.8,0,0;040.1,0,0;039.9,0,0;042.5,0,0;043.4,0,0;041.3,0,0;040.5,0,0;040.0,0,0;0
40.2,0,0;041.6,0,0
```

The sensor values are:

t = 2015-06-10T14:12:14Z -> Timestamp: 2015-06-10 14:12:14 (zulu)
n =110.3 -> Sound pressure level: 110.3 dB
o = 1 -> Overload: true
u = 0 -> Underrange: false
b = 56 -> Battery level: 56 %
p= 1 -> Power supply connected
m = 45 -> Modem strength: 45 %
s -> 60 registers of LAeq1s with overload and underrange

When there isn't a command to change, the server will respond the following data:

```
HTTP/1.1 200 OK
Date: Thu, 10 Jun 2015 14:13:38 GMT
Content-Type: none
Content-Length: 0
Connection: close
```

And if there is a command (f.e. change the averaging time to **30 seconds**), the server will respond the following data:

```
HTTP/1.1 200 OK
Date: Thu, 10 Jun 2015 14:13:38 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 32
Connection: close

TA120-T123456@setConfig|t=0030
```

1.3 Sentilo (JSON Format)

1.3.1 Sending data

An API must be created on the server with domain *host_name* and located in the path *link_data*. The TA120 automatically makes a PUT to this API. And the API needs to be able to read the data and return a status code 200 to the TA120 (to report that it was processed correctly).

Protocol description:

Format	Protocol	Method
JSON	HTTP or HTTPS	PUT

Header description:

```
PUT /<link_data> HTTP/1.1
Host: <host_name>
IDENTITY_KEY: <tkn_value>
Content-Type: application/json; charset=UTF-8
```

Parameters	Name	Description
	link_data	Server API path (*)
	host_name	Server API domain name (*)
	tkn_value	server API Key (*)

*: Programmable by "CESVA Sensor Manager" software

Body description:

```
{"sensors":[
      {
            "sensor":"TA120-<id_val>-<par1_name>",
            "observations":[{"value":"<par1_value>", "timestamp":"<time_val>"}]
      },{
            "sensor":"TA120-<id_val>-<par2_name>",
            "observations":[{"value":"<par2_value>", "timestamp":"<time_val>"}]
      },{
            ...
      }
]}
```

Parameters	Name	Value	Description
	Txxxxxx	id_val	Sensor identifier (*Txxxxxx* is the serial number)
	*dd/mm/yyyy*T*hh:ii:ss*UTC	time_val	Timestamp (UTC format)
	N	n_val	Sound pressure level (L_{Aeq})
	O	o_val	Overload (boolean) (**)
	U	u_val	Underrange (boolean) (**)
	B	b_val	Optional module BA120 - Battery level (%)
	P	p_val	Optional module BA120 - Power supply status (boolean)
	W	w_val	Optional module WF120 - Wi-fi strength (%)
	M	m_val	Optional module MR120 - Modem strength (%)
	S	s_val	LAeq1s registers (***)

**: If "Send Overload/Underrange" is activated (see the manual of the "CESVA Sensor Manager" software)
***: If "Send LAeq1s" is activated (see the manual of the "CESVA Sensor Manager" software)

1.3.2 Sensor configurations

Ethernet interface with battery (BA120)

```
    {
        "sensor":"<id_val>-N",
        "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-O",
        "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-U",
        "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-B",
        "observations":[{"value":"<b_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-P",
        "observations":[{"value":"<p_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-S",
        "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
    }
```

Ethernet interface without battery

```
    {
        "sensor":"<id_val>-N",
        "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-O",
        "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-U",
        "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-S",
        "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
    }
```

Wi-Fi interface (WF120) with battery (BA120)

```
    {
        "sensor":"<id_val>-N",
        "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-O",
        "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-U",
        "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-B",
        "observations":[{"value":"<b_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-P",
        "observations":[{"value":"<p_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-W",
        "observations":[{"value":"<w_val>", "timestamp":"<time_val>"}]
    },{
        "sensor":"<id_val>-S",
        "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
    }
```

Wi-Fi interface (WF120) without battery

```
{
      "sensor":"<id_val>-N",
      "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-O",
      "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-U",
      "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-W",
      "observations":[{"value":"<w_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-S",
      "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
}
```

Modem 3G interface (MR120) with battery (BA120)

```
{
      "sensor":"<id_val>-N",
      "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-O",
      "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-U",
      "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-B",
      "observations":[{"value":"<b_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-P",
      "observations":[{"value":"<p_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-M",
      "observations":[{"value":"<m_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-S",
      "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
}
```

Modem 3G interface (MR120) without battery

```
{
      "sensor":"<id_val>-N",
      "observations":[{"value":"<n_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-O",
      "observations":[{"value":"<o_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-U",
      "observations":[{"value":"<u_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-M",
      "observations":[{"value":"<m_val>", "timestamp":"<time_val>"}]
},{
      "sensor":"<id_val>-S",
      "observations":[{"value":"<s_val>", "timestamp":"<time_val>"}]
}
```

1.3.3 Response data

The server will respond with an HTTP/HTTPS status code "**200**" if the request was accepted and processed correctly. Other code different to "200", the sensor will consider a server's error.

1.3.4 Retrieving orders

An API must be created on the server with domain *host_name* and located in the path *link_order*. The API makes a GET to the TA120. And then the TA120 reads the Command name and value.

Protocol description:

Format	Protocol	Method
JSON	HTTP or HTTPS	GET

Header description:

```
GET /<link_order>/TA120-<id_value> HTTP/1.1
Host: <host_name>
IDENTITY_KEY: <tkn_value>
Content-Type: application/json; charset=UTF-8
```

Parameters	Name	Description
	link_order	Server API path (*)
	id_val	Sensor identifier (*Txxxxxx* is the serial number)
	host_name	Server API domain name (*)
	tkn_value	server API Key (*)

*: Programmable by "CESVA Sensor Manager" software

If **there is an order**, the server response body will be:

```
{"orders":[{"order":"<cmd_name> <cmd_value>"}]}
```

Parameter	Description
cmd_name	Command name
cmd_value	Command value

The TA120 accepts the follow orders:

Name	Values	Format	Description
t	0010 to 3600	Number (4 digits)	Averaging time parameter in seconds (NOTE: default is 0060)
onlylevel	1 or 0	Boolean	Deactivate (1) or activate (0) the "Overload/Underrange" option
seconds	1 or 0	Boolean	Activate (1) or deactivate (0) the "LAeq1s" option

1.3.5 Examples

To send data:

- Optional modules: BA120 (battery) and MR120 (3G modem)
- api_host = yourserver.com
- api_linkdata = data/path/
- api_token = abcdefgh12345678

The data sensor:

```
PUT /data/path/ HTTP/1.1
Host: yourserver.com
IDENTITY_KEY: abcdefgh12345678
Content-Type: application/json; charset=UTF-8

{"sensors":[
     {
          "sensor":"TA120-T123456-N",
          "observations":[{"value":"041.5", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-O",
          "observations":[{"value":" false", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-U",
          "observations":[{"value":" true", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-B",
          "observations":[{"value":"100", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-P",
          "observations":[{"value":"true", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-M",
          "observations":[{"value":"70", "timestamp":"10/06/2015T14:12:38UTC"}]
     },{
          "sensor":"TA120-T123456-S",
           "observations":[{"value":"046.6,0,0;048.4,0,0;047.4,0,0;043.3,0,0;039.9,
0,0;039.8,0,0;039.4,0,0;040.5,0,0;040.4,0,0;040.4,0,0;040.8,0,0;040.1,0,0;040.2,0,0
;040.1,0,0;039.7,0,0;040.3,0,0;039.9,0,0;040.1,0,0;040.1,0,0;039.8,0,0;040.0,0,0;04
0.1,0,0;041.0,0,0;045.3,0,0;044.4,0,0;040.1,0,0;040.0,0,0;040.0,0,0;040.0,0,0;040.0
,0,0;039.5,0,0;039.9,0,0;040.1,0,0;039.9,0,0;040.4,0,0;040.5,0,0;040.5,0,0;040.4,0,
0;041.6,0,0;041.5,0,0;044.2,0,0;040.3,0,0;039.9,0,0;039.9,0,0;040.1,0,0;039.9,0,0;0
39.9,0,0;039.5,0,0;040.1,0,0;040.0,0,0;039.8,0,0;040.1,0,0;039.9,0,0;042.5,0,0;043.
4,0,0;041.3,0,0;040.5,0,0;040.0,0,0;040.2,0,0;041.6,0,0",
"timestamp":"10/06/2015T14:12:38UTC"}]
     }
]}
```

The server will respond the following data:

```
HTTP/1.1 200 OK
Date: Thu, 10 Jun 2015 14:13:38 GMT
Content-Type: application/json; charset=UTF-8
```

To retrieve orders:

- api_host = yourserver.com
- api_linkorder = order/path/
- api_token = abcdefgh12345678
- id_value = T123456

The data sensor:

```
GET /order/path/TA120-T123456 HTTP/1.1
Host: yourserver.com
IDENTITY_KEY: abcdefgh12345678
Content-Type: application/json; charset=UTF-8
```

When there isn't an order to change, the server will respond the following data:

```
HTTP/1.1 200 OK
Date: Thu, 26 Mar 2015 09:50:41 GMT
Content-Type: application/json; charset=UTF-8
```

And when there is an order (f.e. change the averaging time to **30 seconds**), the server will respond the following data:

```
HTTP/1.1 200 OK
Date: Thu, 26 Mar 2015 09:50:41 GMT
Content-Type: application/json; charset=UTF-8

{"orders":[{"order":"t 0030"}]}
```

2.1 Sensor setting: CESVA Sensor Manager

With the **CESVA Sensor Manager** software, you can configure the follow parameters:

▪ Average time	Select the Average time T

For more information, please, see the manual of the **CESVA Sensor Manager** software.

2.2 Technical specifications

Power supply	8 to 36 VDC
Output	4 to 20 mA / 25.0 to 125.0 dB $L_{AeqT} = \left((I_{out} - 4) \cdot \frac{100}{16} \right) + 25\ dB$ Note 1: I_{out} is the output current, in mA Note 2: The measurement range is 35.0 to 120.0 dB Note 3: T is the programmed averaging time. The output current is updated every T. Note 4: If overload occurs, L_{AeqT} > 124.5 dB
Maximum error	±0.2 dB, respect to the measured level
Connections	Without polarity

2.3 Brief description

A typical 4-20mA current-loop circuit is made up by next elements: a sensor (TA120); a power supply for the loop; and a receiver (to read the information). These elements are connected in a closed, series circuit, loop.

The sensor output transmits the proportional 4-20mA dc-current that circulates within the closed series-loop. The 4 mA represents the sensor's lowest level output, and 20 mA representing the sensor's highest level output.

The receiver, normally a data acquisition system, and converts the 4-20mA current back into a voltage which can be further processed and/or displayed.

Current loops are ideal for data transmission because of their inherent insensitivity to electrical noise. In a 4-20 mA current loop, all the current flows through all components.

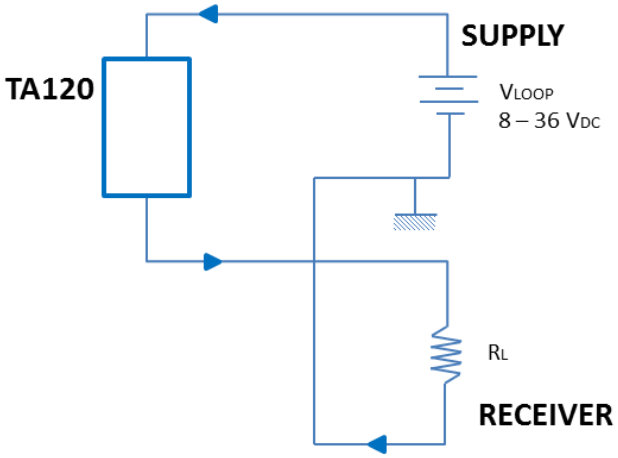


Fig.1 Current Loop (4 – 20 mA)

Reserves the right to amend the characteristics and accessories in this manual without prior notice.

CESVA instruments, s.l.u

Maracaibo, 6 • 08030 BARCELONA, SPAIN

Tel. (+34) 934 335 240 • FAX (+34) 933 479 310

info@cesva.com • www.cesva.com