

## User Guide

Project designer and developer

[Dr. Domenico Suriano](mailto:domenico.suriano@enea.it) , [domenico.suriano@enea.it](mailto:domenico.suriano@enea.it)

### Description

*SentinAir* is a device designed and developed for data acquisition from any type of instruments, sensors or devices (see figure 1). The core of the system is the *Raspberry 3 B+* board, therefore the software presented in this guide can be installed either on a *SentinAir* device, or on a *Raspberry 3 B+* board. Devices have to be plugged into *SentinAir* by USB, Ethernet, serial, SPI, or I2C port. Currently the system has been tested only with devices connected by the USB, Ethernet, and serial port available on the *Raspberry* board ,which is the "brain" of *SentinAir*. The system is based on command line interfaces, except for the web pages served by the web server installed inside.



Figure 1: on the left, *SentinAir* device. On the right some devices currently tested by *SentinAir*.

### SentinAir software installation

This software runs on *Raspbian Stretch Lite* Operative System (OS), therefore the first step is downloading and installing it on a *SentinAir* device or on a *Raspberry 3 B+* board. For a correct *SentinAir* software installation, please, follow in sequence the below procedures. The user can log in the *SentinAir* system by plugging a keyboard into the USB port and a monitor into the on purpose

HDMI port. An alternative way is connecting to the system via SSH connections by using programs such as *putty.exe* or *WinSCP.exe* (for *Windows* operative systems).

#### a) Installing Raspbian Stretch Lite operative system

1. Download the *Raspbian Stretch Lite* OS image into a computer from the link: <https://www.raspberrypi.org/downloads/raspbian/> and flash it in an SD card memory featured by at least 2 GB of room by using, for example, *Win32diskimage.exe* program (for *Windows* operative systems).
2. Create a blank file called *ssh* (no extension). Save this file in the root of the MicroSD card with the flashed image. This is for enabling SSH connections at start up.
3. In order to allow *SentinAir* to connect to a Wi-Fi network when it turns on, create a *wpa\_supplicant.conf* file and write inside the following lines:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<your country code>
# home network; allow all valid ciphers
network={
    ssid=<your network name>
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk=<your network password>
}
```

Save and close the file. If you are using *Windows* operative system to write the file, you have to convert it in the *Linux/Unix* standard. To do this, it is possible to use the program *dos2unix.exe*. Save *wpa\_supplicant.conf* in the root of the MicroSD card with the flashed image.

4. Plug the Sd card in the *Raspberry* board and power it.
5. Find the *SentinAir* IP address by using programs to scan the network set in the *wpa\_supplicant.conf*, for example, for this purpose, there is an “app” for Android system called *FindPi*. Once found what is the *SentinAir* device IP address, it will be possible continue the installation process by connecting to *SentinAir* from another computer via SSH connections. To do this, for example, it is useful the program *putty.exe*, if the computer has a *Windows* operative system running on.

#### b) Preparing Raspbian for SentinAir installation

1. Power the *SentinAir* device or the *Raspberry 3 B+* board, log in it, and type the command: *sudo raspi-config*.
2. To set the device name, select *Network Options/Hostname* and insert the device name. It is mandatory to insert a name with a "hyphen" inside, for example: *sentinair-S1*. This is for the correct functioning of *SentinAir* software. The part of the name following the hyphen is the identity number you choose for the device. Do not exit from *raspi-config* yet.
3. To use the serial port for connecting devices, select *Interfacing options/Serial* and then press in sequence "NO" and "YES". If everything is fine, you should see the message "*The serial login shell is disabled, The serial interface is enabled*". Then press "OK".

4. To expand the use of the memory on your SD card, select *Advanced Options/ expand Filesystem*. Select then *FINISH* and reboot the system.
5. Run the command `sudo nano /etc/hosts` and update the file by modifying the line: “127.0.0.1 *raspberrypi*” in the line: “127.0.0.1 <device-name>”, where <device-name> has to be the same name selected in the step 2 (which is, as example *sentinair-S1*). Reboot the system.
6. To complete the configuration for the serial port called `/dev/ttyAMA0`, we have to disable the Bluetooth service that uses this port and set it as a general purpose serial port. To do this, type `sudo nano /boot/config.txt` and then add at the end of file the row: `dtoverlay=pi3-disable-bt`. Save and close the file. After done this, prevent Bluetooth service to use serial port by typing the command: `sudo systemctl disable hciuart`. Then reboot the system by running `sudo reboot`.
7. Update the *Raspbian* repository by typing the command: `sudo apt-get update`.
8. From now on, to perform the next steps of installation, it is needed to log in the system by the help of a keyboard and a monitor. This is necessary because the built-in Wi-Fi modem will turn in a Wi-Fi access point, and therefore it is impossible to continue while *SentinAir* is connected to a pre-existing Wi-Fi LAN. Moreover, to connect the device to internet, it is going to be necessary an USB stick modem with a valid SIM card enabled by an Internet Service Provider to connect to internet (for example, the model *Huawey E303* does not require installation procedures). This is why the Ethernet port will be reserved to connect instruments to form a local “stand-alone” LAN. Therefore run `sudo halt` to stop the system, plug the USB stick modem, the keyboard and a monitor, then log in again.
9. Download and install *dnsmasq* and *hostapd* by typing the command `sudo apt-get install -y dnsmasq hostapd`.
10. Stop the execution of the software just installed by running the commands: `sudo systemctl stop dnsmasq` and `sudo systemctl stop hostapd`
11. Set a static IP for the system and configure *dhcpcd* by opening the file *dhcpcd.conf* with the command `sudo nano /etc/dhcpcd.conf` and add at the end of the file the lines:  
interface wlan0  
static ip\_address=192.168.4.1/24 (or an alternative IP address you choose)  
nohook wpa\_supplicant  
dhcp-range=192.168.4.2,192.168.4.100,255.255.255.0,24h  
interface eth0  
static ip\_address=192.168.20.1/24  
static routers=192.168.20.1  
nogateway  
Save and close the file.
12. Configure the *hostapd* software by creating the file *hostapd.conf*, to do this, run the command `sudo nano /etc/hostapd/hostapd.conf`. Then insert in the file the following lines:  
interface=wlan0  
driver=nl80211  
ssid=device-name (for example: *sentinair-S1*)  
hw\_mode=g  
channel=7  
wmm\_enabled=0  
macaddr\_acl=0

```
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=sentinair1
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Then save and close the file.

13. Complete *hostapd* configuration by typing `sudo nano /etc/default/hostapd`. Then modify the line:

```
#DAEMON_CONF=""
```

in the new line

```
DAEMON_CONF="/etc/hostapd/hostapd.conf "
```

Save and close the file.

14. Run the command `sudo nano /etc/init.d/hostapd` and replace the line

```
DAEMON_CONF=
```

with the line

```
DAEMON_CONF=/etc/hostapd/hostapd.conf.
```

Save and close the file.

15. Run the command `sudo nano /etc/dnsmasq.conf` and add at the end of the file the following lines:

```
interface=wlan0
```

```
dhcp-range=192.168.4.2,192.168.4.100,255.255.255.0,24h.
```

Save and close the file.

16. Finally run the commands:

```
sudo systemctl restart dhcpcd.
```

```
sudo systemctl unmask hostapd
```

```
sudo systemctl enable hostapd
```

```
sudo systemctl start hostapd
```

```
sudo service dnsmasq start
```

```
sudo reboot
```

### c) Installing lighttpd web server

1. Install *lighttpd* by the command

```
sudo apt-get install -y lighttpd.
```

2. Enable the *cgi-bin* module of the web server by the command

```
sudo lighty-enable-mod cgi
```

3. Open `/etc/lighttpd/conf-available/10-cgi.conf` and modify the lines:

```
$HTTP["url"] =~ "^/cgi-bin/" {
    cgi.assign = ( "" => "" )
}
```

in

```
$HTTP["url"] =~ "^/cgi-bin/" {
    alias.url += ("/cgi-bin/" => "/var/www/cgi-bin/")
    cgi.assign = ( "" => "" )
}
```

4. Run the command *sudo service lighttpd force-reload*
5. Create the folders: */var/www/html/data*, */var/www/html/log* and */var/www/html/img*. Give them the read, write, and execute rights for all the users.
6. Rename the file */var/www/html/index.lighttpd.html* in */var/www/html/index.lighttpd.html\_*. Copy the files *index.html*, *sentinair.jpg*, and *btnplt.png* in the folder */var/www/html/*.
7. Copy the files *fileinspector.py* and *surianocliweb.py* in the folder */var/www/cgi-bin/* and give them read, write, and execution rights for all the users. Then reboot the system.

#### d) Installing SentinAir software modules

1. Install *python3-serial* by running the command *sudo apt-get install python3-serial*.
2. Install *pip3* by running the command *sudo apt-get install python3-pip*
3. Install the library *RPi.GPIO* by running the command *sudo apt-get install python3-RPi.GPIO*.
4. Install the library *libscrc* by running the command *sudo pip3 install libscrc*.
5. Install the library *python3-matplotlib* by running the command *sudo apt-get install -y python3-matplotlib*
6. Reboot the system.
7. Create the folder */home/pi/sentinair*. SentinAir software files have to be mandatorily copied in this folder
8. Copy in */home/pi/sentinair/* the following files and folders:
  - *sentinair.py*
  - *sentinair\_system\_manager.py*
  - *sentinair\_system\_installer.py*
  - *imap-smtp-interface.py*
  - *imap-smtp-monitor.sh*
  - *devices*

The folder *devices* is the folder where devices drivers have mandatorily to be placed.

9. In order to get started SentinAir software when system boots, insert the following lines in the file */etc/rc.local* before the line "exit 0":

```
python3 /home/pi/sentinair_system_manager.py&
sh /home/pi/sentinair/imap-smtp-monitor.sh&
```

Open the *imap-smtp-interface.py* file and put your e-mail account data in the lines following the line *"## imap/smtp connection settings and e-mail account"* like this:

```
FROM_EMAIL = "my-email address"
FROM_PWD   = "my email account password"
SMTP_SERVER = "my smtp server url"
IMAP_SERVER = "my imap server url"
```

10. Reboot the system

## Setting up SentinAir internet connection

The channel to reach a *SentinAir* system from Internet is given by the USB stick modem plugged into one of the USB ports available. If a *Huawey E303* is used for the purpose, no installation procedures are required. Others USB stick modem models has not been tested yet. Internet Service Providers (ISP) usually does not give public IP addresses to the user, therefore *SentinAir* could not be reachable from

anywhere. To overcome this issue, it is possible to use the “IP tunneling” services, that give the possibility to get around this hurdle. Prices of these services can vary from free to few Euros per month. Examples of web companies offering this services are [www.dataplicity.com](http://www.dataplicity.com), or [www.pitunnel.com](http://www.pitunnel.com) or. <https://remote.it>.

## SentinAir start-up

SentinAir get started when the electronic hardware is powered, or alternatively, but it is not a normal procedure, by typing the command:

```
sudo python3 /home/pi/sentinair/sentinair_system_manager.py.
```

When SentinAir get started, before getting ready, it performs some preliminary operations to properly operate. Firstly, it turns on the red check light to indicate that the system is starting, then it performs the scanning of the system ports to find what devices, among the one whose driver is already installed in the system, are plugged into. When a device is found, it got connected to the system. This operation might last few seconds, and during this time you should see, if you have a monitor plugged into the HDMI port, system messages on the screen. From now on all the relevant operations and events are logged in the on purpose log file of the system manager (the default name is *sentinair-log.txt*). During this period the system cannot accept commands from the user, sometimes it can last few minutes. As soon as this operation is completed, *SentinAir* reads the file *status.sentinair*, where there are information about the device status: “stand-by”, or “active monitoring”. If the system is in “active monitoring” status, a new measurement session get started, therefore the green led check light turns on and a new file containing the measurement data is created. The last operation of the start-up phase is the sending of an e-mail to the e-mail address indicated in the *imap-smtp-interface.py* module, if *imap-smtp-interface.py* is configured and if the USB stick modem is plugged into the system (see figure 2). In the current version of the software, *SentinAir* configuration is featured by a static IP on the cable Ethernet port. Devices connected to this interface will form a “stand-alone” net, that is not connected to internet. If the user needs to connect more devices, for example, it is possible to use an Ethernet switch.

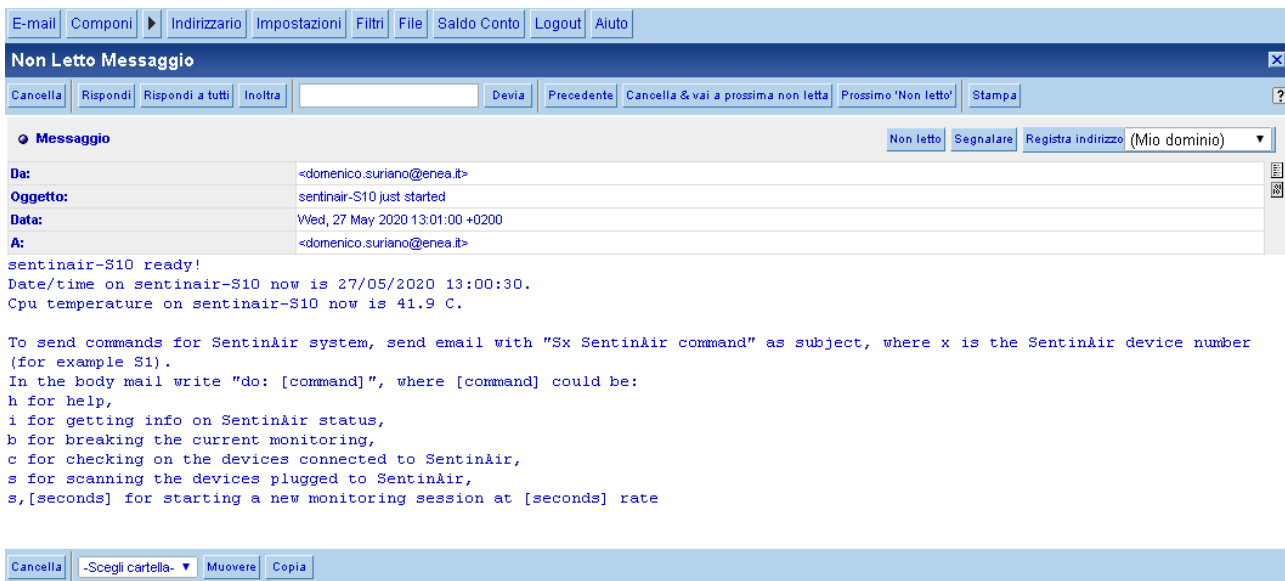


Figure 2: E-mail sent by SentinAir at start-up time.

## Installing devices drivers in SentinAir

SentinAir can read data from any devices as long as the specific driver of the device is installed in the system. It is mandatory that the devices drivers files have to be in the folder */home/pi/sentinair/device*. In order to easily install drivers, the user can log in the system by SSH connection via Wi-Fi link through the LAN set up by *SentinAir*. Then the user has to use the module *sentinair\_system\_installer.py* by opening it through the command:

*sudo python3 /home/pi/sentinair/sentinair\_system\_installer.py*.

Once opened the installer, the commands available are listed in table 1.

*Table 1: commands available for the SentinAir system installer*

Command function	Command syntax	Example	Effect
Device driver installation	i,device_name.py	i,pms3003.py	Installs the driver for the device pms3003
Device driver uninstalling	u,device_name.py	u,pms3003.py	Uninstalls the driver for the device pms3003
Checking the current drivers installed	c	c	Displays a list of the drivers currently installed
Modifying the current path of the system manager module	m	m	Writes in the file "manager_dir.sentinair" the new path of the manager system module
Asking for the commands list available	h	h	Displays a brief manual where are shown the commands
Quit the program	q	q	Quits the programs

An example of *sentinair\_system\_installer.py* use is shown in figure 3. After updating the system with new installations, it has to be rebooted or restarted.

```
pi@sentinair-S10: ~
login as: pi
pi@192.168.4.1's password:
Linux sentinair-S10 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 27 13:05:33 2020 from 192.168.4.14

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@sentinair-S10:~$ sudo python3 /home/pi/sentinair2.0/sentinair_system_installer.py

SENTINAIR SYSTEM DEVICES DRIVERS INSTALLER 1.0
This software has been written by Dr. Domenico Suriano for SentinAir system.

These are the commands available in SentinAir system installer:
1) to modify the path where "sentinair_system_manager.py" is located,
press 'm' and the "Enter" key
2) to check what devices are installed, press 'c' and then the "Enter" key
3) to uninstall devices, type "u,your_device_file_name.py" and then the "Enter" key
4) to install devices, type "i,your_device_file_name.py" and then the "Enter" key
5) to get help, press 'h' and the "Enter" key
6) to quit, press 'q' and the "Enter" key

>> c
ircal is installed in SentinAir
ac32 is installed in SentinAir
af22 is installed in SentinAir
co12m is installed in SentinAir
lcss_adapter is installed in SentinAir
multisensor_board is installed in SentinAir
nox405 is installed in SentinAir
o342 is installed in SentinAir
pms3003 is installed in SentinAir
v72m is installed in SentinAir
go3 is installed in SentinAir
>> u,ircal
ircal successfully uninstalled from SentinAir system.
>> i,ircal
Impossible to install ircal:
ircal does not exist in /home/pi/sentinair2.0/devices
>> i,ircal.py
ircal successfully installed in SentinAir!
>> q
pi@sentinair-S10:~$
```

Figure 3: the installer user interface

## Using the SentinAir user interface

*SentinAir* system main interface is provided by the *sentinair.py* module. The user has to log in the system by SSH connection via Wi-Fi link through the LAN set up by *SentinAir*. To open the *SentinAir* user interface, type the command:

*python3 /home/pi/sentinair/sentinair.py.*

Once opened the user interface, the commands available are listed in table 2.



Table 2: commands available for the *SentinAir* user interface

Command function	Command syntax	Example	Effect
Starting a new monitoring session	s,seconds	s,60	A new monitoring session get started. Every 60 seconds devices measurements data are read and stored in a file
Stopping a measurement session	b	b	Stops the current measurement session
Retrieving information about the system status	i	i	Displays information such as: the devices currently connected to the system, the sampling rate of the measurement session, or if the system is in stand-by.
Retrieving information about the devices connected to the system	c	c	The system returns the information about the connected devices: identity, units of measurements, current measurements
Asking for the commands list available	h	h	Displays a brief manual where are shown the commands
Scanning the ports and connecting the devices plugged into <i>SentinAir</i> ports	s	s	The system scans all the ports (USB, Ethernet, etc.) and recognize if one the installed device is plugged into
Quit the program	q	q	Quits the programs

An example of the use of the *SentinAir* user interface is shown in figure 4.

```
pi@sentinair-S10: ~
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@sentinair-S10:~ $ python3 /home/pi/sentinair2.0/sentinair.py

INSTRUMENTATION MANAGER by Domenico Suriano opening...

press i[ENTER] to get info on the current status
press q[ENTER] to quit the command console
press h[ENTER] for command viewing
press s[ENTER] for searching devices
press c[ENTER] for checking devices
press b[ENTER] for stopping sampling sessions
press s,<sampling rate in seconds>[ENTER] to start and log a sampling session

INSTRUMENTATION MANAGER by Domenico Suriano ready!

No measurement session ongoing

Devices connected:

IRCA1-ttyUSB0
co2[ppm]
1799.7

PMS3003-ttyAMA0
pm1[ug/m3];pm2.5[ug/m3];pm10[ug/m3]
5.0;7.0;7.0

>>> s,60

Session started at 60 sec. rate
>>>
27/05/2020_13:22:11
IRCA1-ttyUSB0
co2[ppm]: 1518.7
PMS3003-ttyAMA0
pm1[ug/m3]: 4.0
pm2.5[ug/m3]: 5.0
pm10[ug/m3]: 6.0

>>> b
Measurement session stopped

>>>
File /var/www/html/data/sentinair-S10_2020-05-27_13-22-11.txt closed

>>>
File /var/www/html/data/sentinair-S10_2020-05-27_13-22-11_hourlymeans.txt closed

>>>
File /var/www/html/data/sentinair-S10_2020-05-27_13-22-11_dailymeans.txt closed

>>> █
```

Figure 4: the *SentinAir* user interface.

## Controlling SentinAir by e-mail

*SentinAir* has been designed for the use in all that cases where it is necessary to perform experiments or measurements in remote areas, where possibly the internet link is weak or unstable, and therefore, where it is difficult to remotely control and download data from the device. For this purpose, commands and data can be exchanged by e-mail sending and receiving. *SentinAir* checks periodically on the IMAP server configured in the *imap-smtp-interface.py* for the presence of e-mails containing commands, and when the radio signal allows it, sends back the response. The *imap-smtp-interface.py* module is optional, and it is not essential for *SentinAir* operation. If it is present, it is launched by *sentinair\_system\_manager.py* at start-up. The user can send commands through e-mail for both the *SentinAir* system or for the operative system programs and services. A

typical *SentinAir* command is, for example, “s,60”, which causes the start of a new measurement session at 60 seconds sampling rate. A command directed for the operative system could be, for example, “ls -l”, which returns the list of files and directories contained in the current system directory. E-mails containing commands for the operative system must have as subject the text: “*device\_ID* system command”,

where the “*device\_ID*” is the part of the device name which follows the hyphen. For example: if the device name was set as “*sentinair-S1*” during the software installation process, then the *device\_ID* will be “S1” (see figure 5 and 6). Moreover, e-mails containing commands directed to *SentinAir* software must have as subject the text:

“*device\_ID SentinAir* command” (please, mind the correct case of letters).

The body of the e-mail must have a line where there is “do: command”. For example: “do: s,60” (please, mind the “space” character between “do:” and the command). It is possible to download every kind of file present in the device by sending an e-mail having the subject as: “*device\_ID* system command”, and the body having the line as: “do: fget /path\_of\_the\_file/file\_to\_download” (mind the space characters between “do:” and “fget” and between “fget” and “/path\_of\_the\_file/file\_to\_download”). For example: “do: fget /var/www/html/log/sentinair-log.txt”.



Figure 5: e-mail sent to *SentinAir* containing a command for the operative system

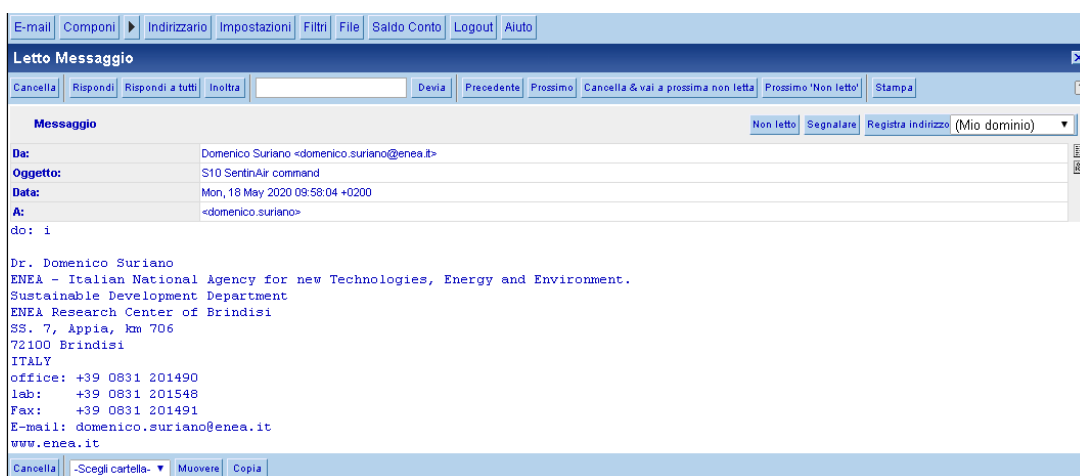


Figure 6: - mail sent to *SentinAir* containing a command for SentinAir system

## SentinAir operations featuring the measurement session

When a new measurement session get started (for example by the command “s,60”), the green led check light turns on. Three new files are created: the file containing the records of the measurements along with the time stamps (see figure 7) and named: “*device\_name\_date\_time.txt*” (for example: *sentinair-S1\_2020-02-03\_09-30-15.txt*), the file containing hourly averages of measurement and named: “*device\_name\_date\_time\_hourlymeans.txt*” (for example: “*sentinair-S1\_2020-02-03\_09-30-15\_hourlymeans.txt*”), and the file containing the daily averages of the measurements, named: “*device\_name\_date\_time\_dailymeans.txt*” (for example: “*sentinair-S1\_2020-02-03\_09-30-15\_dailymeans.txt*”). Hourly and daily averages are computed in real time by the module *sentinair\_system\_manager.py* and they are stored in the path indicated in that module (which is for default: “*/var/www/html/data*”). Another operation performed by the *sentinair\_system\_manager.py* module during the measurement session is the plotting of each magnitude measured by each of the devices connected to *SentinAir*. The plots are stored in jpg files placed in the path “*/var/www/html/img*” (which is their default path). Those files are encapsulated in web pages and served to the user when he connects by a web browser to *SentinAir*.

```
Date/time;IRCA1-ttyUSB0_co2[ppm];PMS3003-ttyAMA0_pm1[ug/m3];PMS3003-ttyAMA0_pm2.5[ug/m3];PMS3003-ttyAMA0_pm10[ug/m3]
27/05/2020_15:28:22;1228.9;5.0;6.0;7.0
27/05/2020_15:28:52;2156.7;5.0;7.0;7.0
27/05/2020_15:29:22;4035.6;4.0;6.0;6.0
27/05/2020_15:29:52;2188.8;5.0;7.0;8.0
27/05/2020_15:30:22;1676.5;5.0;8.0;8.0
27/05/2020_15:30:52;1955.0;5.0;7.0;7.0
27/05/2020_15:31:22;2026.3;5.0;6.0;7.0
27/05/2020_15:31:52;2185.1;5.0;6.0;6.0
27/05/2020_15:32:22;3281.6;4.0;6.0;7.0
27/05/2020_15:32:52;2805.8;4.0;5.0;5.0
27/05/2020_15:33:22;1940.0;5.0;5.0;5.0
27/05/2020_15:33:52;2222.1;5.0;6.0;7.0
27/05/2020_15:34:22;2477.7;4.0;5.0;5.0
27/05/2020_15:34:52;1438.0;4.0;6.0;6.0
27/05/2020_15:35:22;2571.1;5.0;7.0;7.0
27/05/2020_15:35:52;2544.4;4.0;6.0;6.0
27/05/2020_15:36:22;1882.4;4.0;6.0;6.0
27/05/2020_15:36:52;3862.6;5.0;7.0;7.0
```

Figure 7: file containing records of a *SentinAir* measurement session.

## SentinAir web pages

The user can monitor *SentinAir* activity by connecting through a web browser to *SentinAir* (see figure 8). As for the other user interfaces given by the *sentinair.py* module and the *sentinair\_system\_installer.py* module, the web server of *SentinAir* can be reached by connecting through Wi-Fi link at the address set in the */etc/dhcpd.conf* file (which default is 192.168.4.1), or through internet connections via the selected “IP tunneling” service. Web pages of *SentinAir* allow the user to download all data and log files that are in system.



Figure 8: web pages from *SentinAir*

## How to write drivers devices for *SentinAir*

The creation of new devices drivers for *SentinAir* is highly encouraged. It is desirable that researchers, stakeholders or users write drivers for their own devices to use with *SentinAir* system. It is mandatory that drivers files have to be placed in the subfolder “*device*” of the folder where the *SentinAir* software is copied. When a new driver is going to be written, the user must follow some rules in order to get it fully compatible and operating with the pre-existing software:

- The driver must contain a Python class which name must be the same of the file, but capitalized. For example: the new driver file is called “*irca1.py*”, so the class inside must be named *Ircal*. Therefore the line will be: “*class Ircal:*”;
- The class must have the function called “*getConnectionType(self)*” that must return one of these string: “usb”, “serial”, “eth”, “spi”, or “i2c”;
- The class must have the function named “*getConnectionParams (self)*” that must return a Python tuple with the connection parameters. For example: the device *Ircal* connects to *SentinAir* through USB port, therefore the tuple returned is [*port name, baud rate*];
- The class must have the function called “*getDeviceType(self)*” that must return a string describing the device type;
- The class must have the function called “*getIdentity(self)*” that must return a string describing the device name;
- The class must have the function called “*getSensors(self)*” that must return a string describing the magnitudes measured separated by a semicolon mark. For example: “*NO2[ppb];NO[ppb];NOx[ppb]*” ;
- The class must have the function called “*sample(self)*” that must return a string containing the last measurements separated by a semicolon mark. For example: “*27,76;0.1234;456,9*”. It is mandatory that the fields contained in the string must be in the same number of the ones in the string returned by the function “*getSensors(self)*”. For example: if the

`getSensors(self)` function returns a string like `"NO2[ppb];NO[ppb];NOx[ppb]"`, the `sample(self)` function must return a string like `"27,76;0,1234;456,9"`, and it cannot be like `"27,76;0,1234"` or `"27,76"`.

- The class must have the function called `connect(self)`, if the device has to be plugged to the Ethernet port, or `connect(self,port)`, if the device has to be plugged to the USB or the "serial" port. This function must establish a connection to SentinelAir system. If the connection succeeds, it must return the integer value 1. As example of this function, please see the files `irca1.py` and `col2m.py`, where is written the code of devices drivers that are plugged into a USB port and into the Ethernet port respectively.
- The class must have the function called `terminate(self)`. It returns nothing, but it must close the USB port, if the device is connected to the USB port, or it must close the socket, if the device is connected to the Ethernet port. As example, see the files `irca1.py` and `col2m.py`.