SoftwareX 12 (2020) 100589

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

SentinAir system software: A flexible tool for data acquisition from heterogeneous sensors and devices

Domenico Suriano

ENEA - Italian National Agency for new technologies, Energy and Environment. Sustainable Development Department, Research Center of Brindisi, SS. 7, Appia, km 706, 72100 Brindisi, Italy

ARTICLE INFO

Article history:
Received 30 May 2020
Received in revised form 14 September 2020
Accepted 14 September 2020

Keywords:
Air quality monitoring
Wireless sensors
Measurement software
Portable monitoring unit
Expandable software

ABSTRACT

SentinAir system has been designed and developed to perform data acquisition from heterogeneous devices, sensors, or instruments. Although designed for facilitating the on-field evaluation and calibration of air quality sensors, it can also be used to acquire data from various kinds of devices deployed far from the laboratory where the Internet connection is weak or unstable. SentinAir, written in Python, can be expanded and used with a wide range of sensors or devices for a variety of field experiments.

© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Code metadata

Current code version	*V1.0*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_234
Legal Code License	*Apache License 2.0*
Code versioning system used	*Git*
Software code languages, tools, and services used	*Python 3.5.3, python3-serial (v.3.2.1), python3-RPi.GPIO (v. 0.7.0), libscrc (v.1.3), python3-matplotlib (v.3.1.2)*
Compilation requirements, operating environments & dependencies	*Raspbian Stretch Lite OS (kernel version: 4.14.50-v7+), hostapd (v.2.6), dnsmasq (v.2.76), lighttpd (v.1.4.45), pip3 (v.9.0.1)*
If available Link to developer documentation/manual	https://github.com/domenico-suriano/SentinAir/blob/master/guide/sentinair-system-user-guide.pdf
Support email for questions	Domenico Suriano: domenico.suriano@enea.it

1. Background and motivations

In the last decades, atmospheric pollution has become one of the most concerning issues for public health [1–3]. Air quality monitoring for assessing pollution levels is, in general, performed by fixed monitoring stations based on Reference Instruments (RIs), such as chemical analyzers. Although the reference instruments give accurate measurements, they are quite expensive, cumbersome, and require frequent maintenance [4–6]. As a result, users cannot afford to acquire a sufficient number of these monitoring stations to achieve good spatio-temporal resolutions of pollutant maps. In order to supplement data given by the RIs, the use of Low-Cost Small commercial gas Sensors (LCSSs) has become more and more popular [5,7]. However, several studies have found that LCSSs have limitations [5,6,8]; for example, they provide different data quality depending on their calibration methods [4,6,9,10], and on the environmental conditions of the location [4,9]. These studies demonstrated that it is necessary to perform a field calibration for each sensor individually to achieve good results in real-world applications.

The commercial availability of LCSSs has hugely increased in the last years [6]. Their costs and performances can significantly vary depending on their technology. In general, to properly evaluate their performance, they need to be calibrated on-field by comparing their data with measurements provided by RIs, co-located with the sensors under test [9]. A vast number of

E-mail address: domenico.suriano@enea.it.

https://doi.org/10.1016/j.softx.2020.100589
2352-7110/© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Table 1
List of the device drivers currently developed for the SentinAir system. This list reports also some technical data related to the devices. Concerning the Environnement instruments, the manufacturer reports the amount of noise affecting the measurement, instead of the accuracy parameter. FS stands for Full Scale. RV stands for Reading Value. N/a means not available.

Sensor or device	Connection interface	Supplier or manufacturer	Range	Sensitivity	Accuracy
IRC-A1 (CO_2 sensor)	USB	Alphasense [11]	0–5000 ppm	N/a	1% FS
PMS3003 (PM sensor)	TTL serial port	Plantower [12]	0–500 $\mu g/m^3$	N/a	10% RV
Multisensor board (to use sensors having analog outputs)	USB	Tecnosens [13]	Depends on the attached sensors (see supplementary material)	Depends on the attached sensors (see supplementary material)	Depends on the attached sensors (see supplementary material)
106L GO3 PRO package (CO_2 and O_3 monitor)	USB	2B technologies [14]	0–2000 ppm for CO_2; 0–100 ppm for O_3	N/a	$\pm$(2% FS + 2%RV) for CO_2; greater of 1.5 ppb or 2% of RV for O_3
405 nm (NOx monitor)	USB	2B technologies [14]	0–10 ppm for NO_2; 0–2 ppm for NO	N/a	Greater of 2 ppb or 2% RV
LCSS USB adapter (to use sensors having analog outputs)	USB	Designed and built in our lab	Depends on the attached sensors	Depends on the attached sensors	Depends on the attached sensors
CO12M (CO chemical analyzer)	Ethernet port	Environnement [15]	0–200 ppm	N/a	0,025 ppm
AF22M (SO_2 chemical analyzer)	Ethernet port	Environnement [15]	0–10 ppm	N/a	0,5 ppb
AC32M (NO_x chemical analyzer)	Ethernet port	Environnement [15]	0–50 ppm	N/a	0,2 ppb
O342M (O_3 chemical analyzer)	Ethernet port	Environnement [15]	0–10 ppm	N/a	0,5 ppb
VOC72M (VOC chemical analyzer)	Ethernet port	Environnement [15]	0–1000 $\mu g/m^3$	N/a	0,025 $\mu g/m^3$

researchers have focused their studies on addressing these issues [6,10,16], leading to a growing scientific community involved in this research area. These devices provide data outputs in a wide variety of formats and types: analog signals, digital data on USB, SPI, I2C, and serial connections. For these reasons, evaluating the performance of LCSSs on the field in co-location with RIs could result in a challenging task for researchers or stakeholders.

As a consequence, the availability of a tool able to acquire, store and treat data provided by diverse devices, RIs and LCSSs in order to calibrate them on the field could be beneficial for the scientific community involved in this research area. To the best of our knowledge, few studies have addressed this issue: in [17], it is proposed a system based on expandable sensor modules with plug-and-play features. In [18], it is presented the remake of a Portable Monitoring Unit (PMU) for monitoring NH_3 and CO_2 emitted by poultry manure.

The lack of a device that enables avoiding the use of multiple and dedicated hardware, and software tools for using various sensors or instruments in harsh environments has led us to develop the *SentinAir* system. Although the system proposed in this paper has been devised to facilitate experiments and research activities in the area of air quality monitoring, it can also be used in different cases. In particular, SentinAir can be used for data acquisition from sensors, devices, or instruments involved in experiments far from the laboratory facilities. In fact, *SentinAir* can be deployed in both outdoor and indoor environments; the only requirement is the presence of a common electrical power source providing alternated current at 220 V. The cost for a SentinAir system (including the CPU board) is less than US $200 (the cost details are provided in [19]).

The software in this paper is written in the Python language for *SentinAir* devices (see Fig. 1). It must be installed on a "Raspberry 3 B+" board, which is the "brain" of *SentinAir*. A "Raspbian Stretch Lite OS" (for downloading it, see [20]) runs on it with all the required dependencies and libraries (see Code metadata).

Python was chosen as the software language of *SentinAir* because it has become a popular scientific programming language [21,22]. In fact, device drivers not yet written for *SentinAir* software are expected to be written by the users on their own. This way, *SentinAir* can be expanded, and therefore it can include the management of devices for any laboratory or research group. The devices or instruments usable by the SentinAir system are all the ones having the following output interface: USB, TTL serial, Ethernet, I2C, SPI, or analog. The devices currently tested by *SentinAir* are listed in Table 1.

Many of the sensors or sensing devices have analog outputs. All these devices have to be attached to an electronic board for converting their analog signals in digital ones using Analog-to-Digital Converters (ADCs). Unfortunately, Raspberry boards do not have built-in ADCs; therefore, devices able to adapt the analog outputs to USB input ports are required. Although several electronic boards available on the market can be fitted for the purpose (see, for example: "Arduino Uno" [23], or the "Multisensor board" by Tecnosens [13]), we preferred to design and develop the "LCSS USB adapter". The reason for creating an in-house device is given by the necessity to optimize the analog-to-digital conversion process. The files necessary to build and set up the "LCSS USB adapter" can be found in [24], while the instructions for its assembly are illustrated in the *SentinAir* user guide released along with the software in open access (see the link in Code metadata).

To use this software, it is necessary to install the specific device drivers in the system by the software module developed on purpose. Therefore, the system will proceed automatically to recognize the connected devices. Subsequently, measurements will be done by reading device and instrument outputs at the selected sampling rate. The output of the system is composed of three datasets in CSV format files. They contain the measurements,

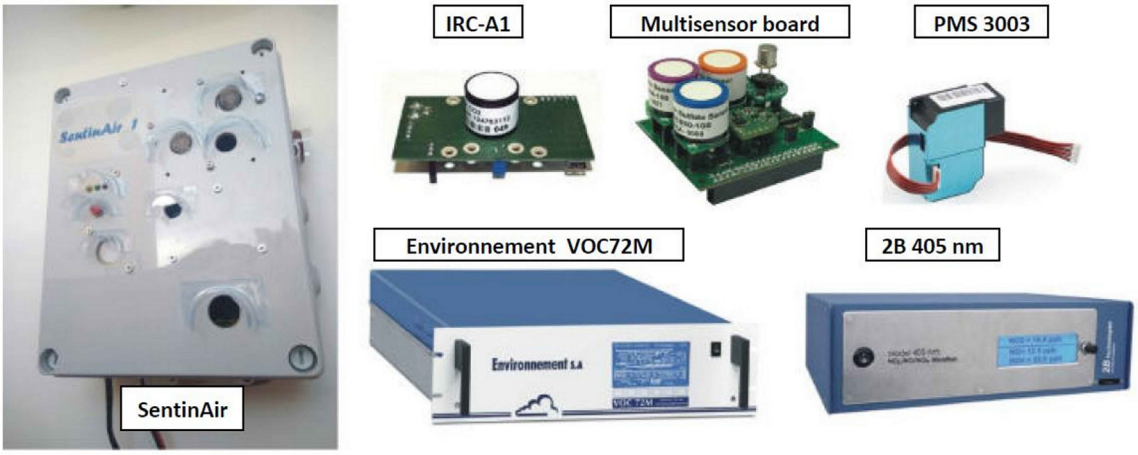

Fig. 1. *SentinAir* prototype and some of the devices currently used by its software. *SentinAir* dimensions are 24.5 cm × 20 cm × 9 cm. Its weight is typically about 1.5 kg. It depends on how many sensors or devices are mounted inside its enclosure. Environnement VOC72M and 2B405 nm are reference instruments featured by respectively a weight of 12.5 kg and 8.5 kg.

their hourly, and daily averages. Dataset records are composed of timestamps along with the device data (an example of them can be found at [25]). These files are stored in the *SentinAir* SD card, and they can be used as inputs for calibration or evaluation processes through other on purpose software tools (for example the Scikit-learn libraries [26]). For more details, it is possible to refer to the *SentinAir* user guide.

2. Software framework

2.1. Software architecture

The architecture of *SentinAir* software includes several modules interacting with each other. Some of them are optional, such as the "Imap/Smtp communication module", while the others form the core of the system. "Imap" stands for "Internet Message Access Protocol"; while "Smtp" stands for "Simple Mail Transfer Protocol". They are protocols to communicate with e-mail servers. Fig. 2 shows the general scheme of the modules and their main interactions. Each module is presented in the next sections.

Users can interact with the *SentinAir* system through three channels: keyboard and monitor plugged into the *SentinAir* hardware, Wi-Fi LAN automatically set up by *SentinAir* at startup time, or through internet connections via USB stick modem plugged into the USB port. The system can be optionally connected to the internet through the "IP tunneling" technology [27,28], which makes reachable the *SentinAir* IP private address. There are several "IP tunneling" service companies available on the web, each of them offering various pricing options: from free pricing plans to a few Euros per month [29–31]. When the internet link is not available, users can reach out to *SentinAir* through its Wi-Fi LAN channel.

2.2. User interface module

This module consists of a command-line interface (see Fig. 3A) that receives user commands and displays system answers in text strings. Commands are passed to the *SentinAir system manager* that executes them and returns the responses. Through this module, users can: start a measurement session at a selected sampling rate, so as to put the system in its active monitoring state, stop a measurement session, so as to put *SentinAir* in the standby state. Moreover, this module gives information about the devices currently connected to *SentinAir* or the current system status, including the last measurements and their sampling rate.

2.3. Web server interface modules

These modules constitute the second user interface. An "HTTP" server (currently, Lighttpd [32] is used) is installed in the system. It acts as a user interface from which it is possible to get all the information about the device. It enables the download of all the stored data files and logs (see Fig. 3B, C) as well. The web pages served by these modules show the plots of the measurements. The "Http" requests activate the *Web server interface modules* through the Common Gateway Interfaces technology [33, 34] supported by the "Http" server.

2.4. Imap/Smtp communication module

The *Imap/Smtp communication module* is the optional user interface. It is useful when the *SentinAir* device is located in areas where wireless internet link is weak or unstable. It periodically connects to Imap/Smtp servers to search for e-mails sent by the user containing commands for *SentinAir*. If something is found, it reads the e-mail, extracts the commands, and transmits them to the *SentinAir system manager*. After the command execution, the response is returned, and subsequently, it will be written in an e-mail body and send back to the user. Obviously, an e-mail account is needed to use this module.

2.5. Driver installer module

The *driver installer* is in charge of the correct device driver installing and uninstalling. When activated by the user, this module automatically modifies the code of the *SentinAir system manager* to install or uninstall drivers in the system. Through this module, it is also possible to check what are the device drivers currently installed in *SentinAir*. After the installation process, or after uninstalling device drivers, the system has to be rebooted or restarted to make the changes effective.

2.6. SentinAir system manager

This module is the engine of the system. All the commands coming from the previously presented modules are executed by it. Data exchange with other system modules is performed through UDP sockets. When the system is in its active monitoring state, it is in charge of reading measures from the connected devices, calculating hourly and daily measures averages, producing data plots, and storing data in the SD card memory of the system.

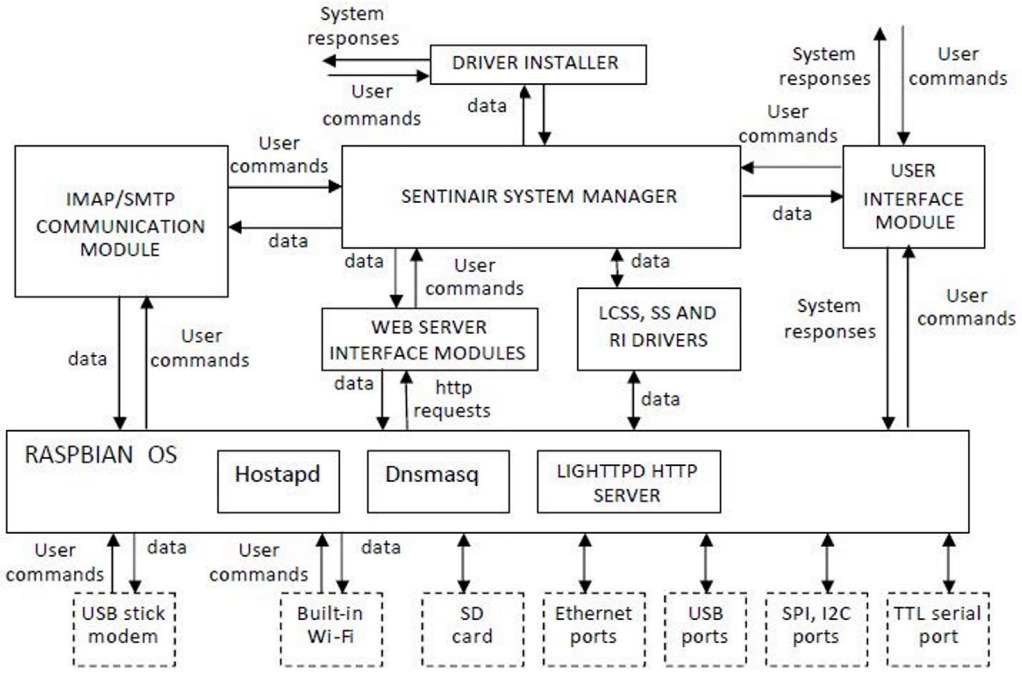

Fig. 2. SentinAir software architecture scheme.

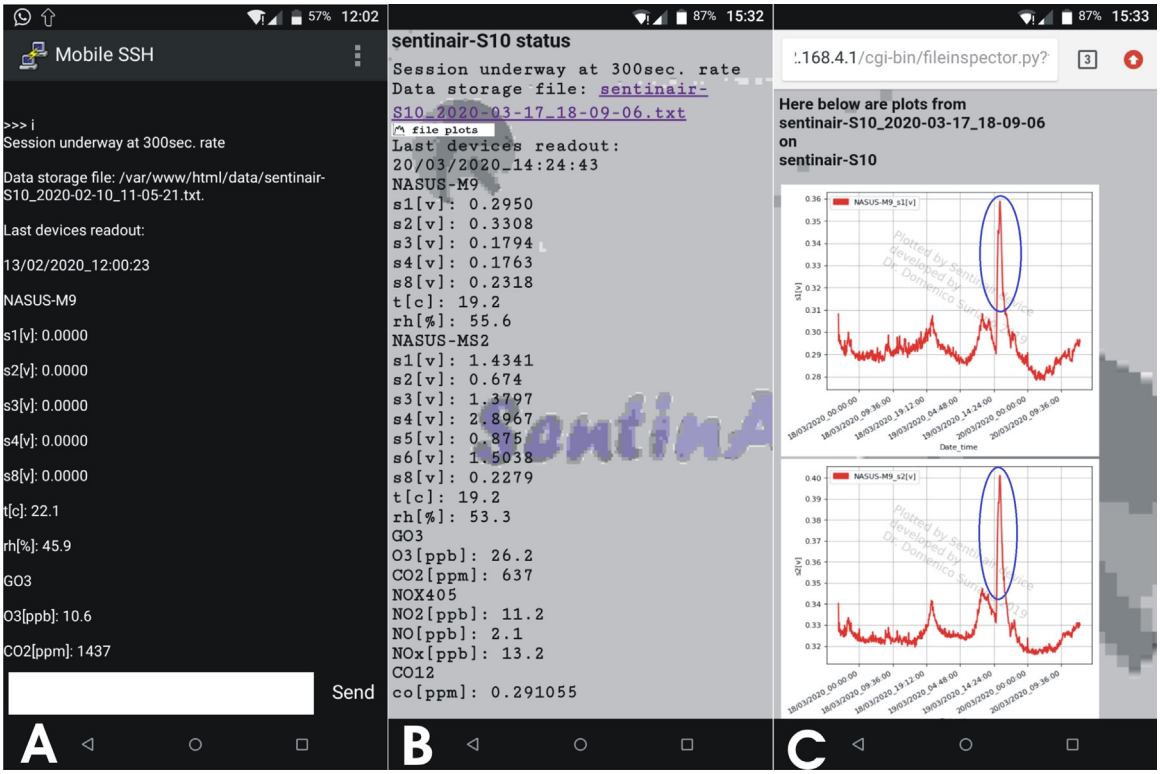

Fig. 3. User interfaces of the *SentinAir* system. (A) command-line user interface module. (B) the web page served by SentinAir, where there is a complete set of information about the system status. (C) the web page containing data coming from sensors or instruments plotted in real-time. The spikes circled in blue are the responses of the two copies of the sensors "SP61" by Nissha-Fis [35] to ozone concentration changes.

It is also in charge of automatically detecting which are the devices connected to the system, and establishing a connection with them, thanks to the device drivers previously installed. This module is a multitasking process composed of several threads simultaneously executed. Most of the computational load is concentrated in the thread called *"plotter"*, and it depends on the number of devices currently in input to the system (see Table 2). It plots the graphs of all the measurements featuring the current

Table 2
Statistics on the execution time of the "plotter" thread. Values are expressed in seconds.

Number of sensors	Average execution time	Standard deviation	Minimum execution time	Maximum execution time
11	17.29	1.20	11.33	20.94
21	34.69	3.18	19.76	42.51
32	50.39	4.37	30.42	62.54
43	64.21	4.96	42.31	79.85

Table 3
SD card memory footprint. Values are expressed in Kbytes. The maximum measurement session duration is calculated, supposing a sampling rate of 120 seconds and a total amount of available memory of 2 GBytes.

Number of sensors	Maximum initial memory allocation	Memory usage per records	Maximum measurement session duration (days)
11	881	0.09	31 276
21	1681	0.15	18 765
32	2881	0.22	12 794
43	3441	0.29	9705

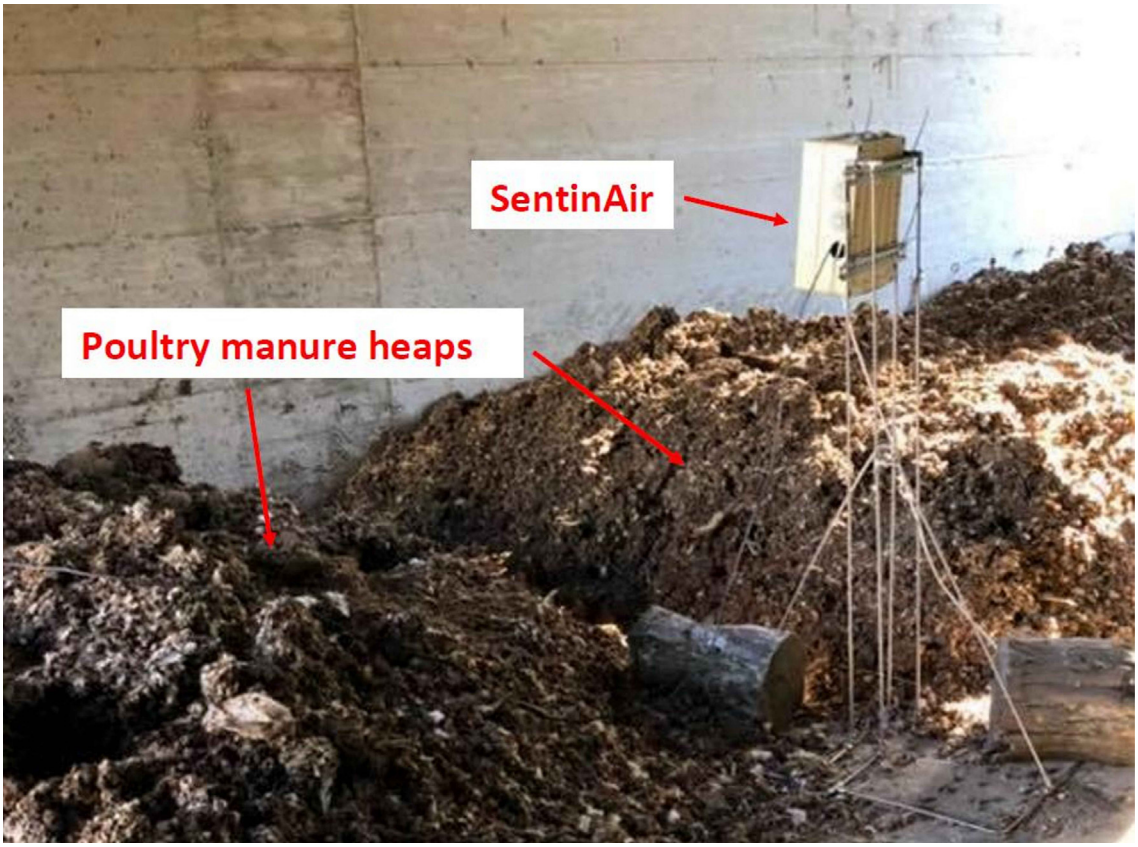

Fig. 4. *SentinAir* system used in gas monitoring activity for the POREM project. The device was controlled from the remote by using the Dataplicity "IP tunneling" service [29]. The sensors inside the device were the TGS825 (H_2S sensor), the TGS826 (NH_3 sensor), and the TGS2611 (CH_4 sensor) by Figaro, the IRC-A1 (CO_2 sensor), the HIH5013 (relative humidity sensor) by Honeywell, and two TC1047A by Microchip (one used for measuring the temperature outside the poultry manure heaps, the other used to measure the internal temperature of the poultry manure).

session. They are stored in "jpg" files, which are viewable through web pages (see Fig. 3C).

Concerning the SD card memory usage, the system uses a fixed and initial amount of memory at the measurement session launch; while, during the "active monitoring status", memory usage increases as new records are stored. Clearly, the final memory footprint strongly depends on the sampling rate of the measurement session. In Table 3, some indicative values of memory usage are shown.

2.7. LCSS, or RI drivers

These modules allow the *SentinAir system manager* to use any LCSS, device, or RI connected to the system. The drivers have to manage the connection ports through which the devices are connected. They also are in charge of reading correctly the data coming from the devices and pass them to the *SentinAir system manager* in the correct format. In order to have full compatibility with the *SentinAir* system, the code structure of drivers has to follow some specific rules: for example, some functions are mandatory, and also a device class must be inside the driver

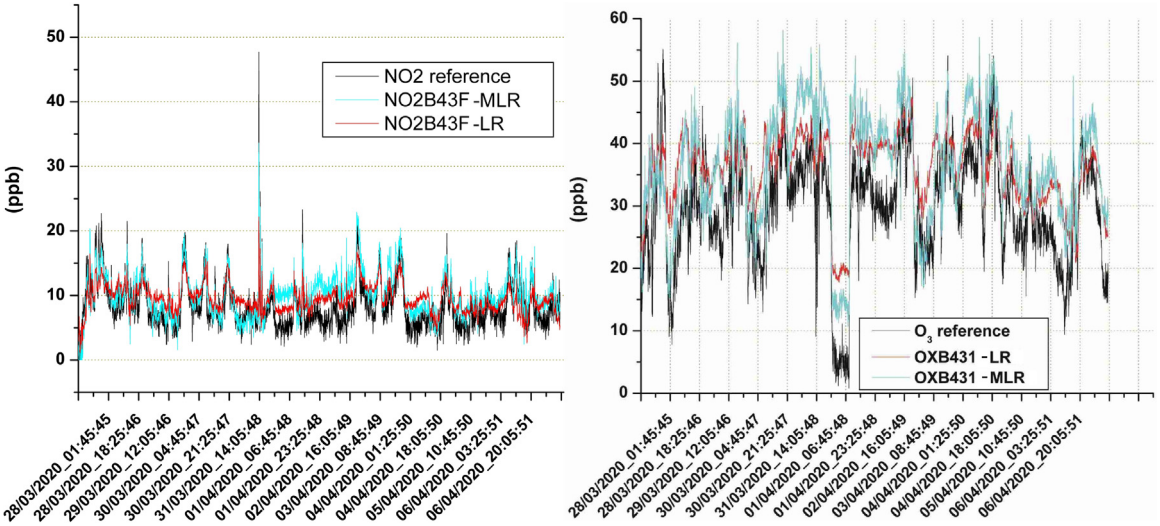

Fig. 5. Calibration and evaluation of NO2B43F and OXB431 sensors by Alphasense [11]. The sensors were mounted on the Individual Sensor Boards (ISBs) provided by the manufacturer [36]. They were wired to the LCSS adapter board. The datasets produced by *SentinAir* were the inputs to scripts based on Scikit-learn libraries [26] for calibrating the sensors using the Linear Regression (red line), and the Multivariate Linear Regression algorithm (cyan line). The black line represents the RI outputs.

code. For more details about how writing drivers compatible with the *SentinAir* system, users can refer to the *SentinAir user guide.* Currently, eleven device drivers have been developed for the *SentinAir* system (listed in Table 1).

3. Illustrative examples

In order to validate the effectiveness of the system, a prototype of the *SentinAir* device was tested for evaluating CO_2, NH_3, H_2S, and CH_4 gas sensors. They were selected to monitor gases emanated by poultry manure heaps in the context of the POREM–LIFE17/ENV/IT/333 project [37]. After the evaluation stage, *SentinAir* was used to measure poultry manure gas emission in a rural area featured by weak and unstable radio signal (see Fig. 4).

SentinAir was also tested as a tool for acquiring data from some LCSSs devoted to monitoring NO_2 and O_3. The system was deployed outdoor in a semi-rural area along with RIs (see the video in the supplementary material section). The datasets built by *SentinAir* were used to calibrate and evaluate the sensors. Some plots representing a comparison of calibrated LCSSs with RIs are shown in Fig. 5.

The main *SentinAir* software functions are shown in the *SentinAir* user guide released along with this paper, while the supplementary material section provides an explanatory video about the *SentinAir* system use.

4. Impact

The interest in investigating the potential of LCSS for supplementing pollutant data provided by RIs is relevant [5–7,10,16]. *SentinAir* system enables data acquisition from various devices or instruments in harsh environments without using multiple dedicated hardware or software tools. Moreover, it allows the real-time control of the experiments from the remote. We anticipate that the ease of use of the Python language will enable researcher groups to write more drivers for widening the number of devices usable by *SentinAir*. However, the impact of the software presented in this paper goes beyond the research area of new technologies for air quality monitoring. In fact, *SentinAir* can be fundamental in all those situations where it is necessary to perform measurements or data acquisition from devices or instruments outside the laboratory environment. Anyway, by being an easy-to-use tool, stakeholders and researchers can easily set up their experiments with the required sensors or instruments also in their laboratory.

5. Conclusions

SentinAir system software has been designed to acquire data from various sensors, devices, or instruments during experiments or activities performed in harsh environments, far from laboratory facilities. The effectiveness of the system proposed in this paper was tested and proved during research activities related to the POREM– LIFE17/ENV/IT/333 project. The most important features of this software are:

- Easy sensor or instrument data acquisition, storing, and remote transmission.
- High flexibility: it is adaptable to a wide range of experiments and research activities
- High personalization level: new device driver developers can quickly expand the software capability to use a wide range of instruments or devices.

Therefore, thanks to its features, this software is a candidate to be a useful multipurpose tool for a wide range of researchers and stakeholders. The creation of new drivers for devices with SPI or I2C output connections, their tests, and the re-writing of *SentinAir* code for alternative operative systems are considered as future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by funds of POREM (“Poultry manure based bioactivator for better soil management through bioremediation”) project co-funded by EU, within the LIFE program (LIFE17/ENV/IT/333).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2020.100589.

References

[1] WHO. Household air pollution and health. World Health Organization; 2018, https://www.who.int/en/news-room/fact-sheets/detail/household-air-pollution-and-health (accessed on 25 May 2020).

[2] Kampa M, Castanas E. Human health effects of air pollution. Environ Pollut 2008;151(2):362–7. http://dx.doi.org/10.1016/j.envpol.2007.06.012.

[3] McConnell R, Berhane K, Gillil F, London SJ, Islam T, Gauderman WJ, et al. Asthma in exercising children exposed to ozone: a cohort study. Lancet 2002;359(9304):386–91. http://dx.doi.org/10.1016/S0140-6736(02)07597-9.

[4] Castell N, Dauge FR, Schneider P, Vogt M, Lerner U, Fishbain B, et al. Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?. Environ Int 2017;99:293–302. http://dx.doi.org/10.1016/j.envint.2016.12.007.

[5] Snyder EG, Watkins TH, Solomon PA, Thoma ED, Williams RW, Hagler GSW, et al. The changing paradigm of air pollution monitoring. Environ Sci Technol 2013;47(20):11369–77. http://dx.doi.org/10.1021/es4022602.

[6] Karagulian F, Barbiere M, Kotsev A, Spinelle L, Gerboles M, Lagler F, et al. Review of the performance of low-cost sensors for air quality monitoring. Atmosphere 2019;10(9):506–47. http://dx.doi.org/10.3390/atmos10090506.

[7] Kumar P, Morawska L, Martani C, Biskos G, Neophytou M, Di Sabatino S, et al. The rise of low-cost sensing for managing air pollution in cities. Environ Int 2015;75:199–205. http://dx.doi.org/10.1016/j.envint.2014.11.019.

[8] Lewis A, Edwards P. Validate personal air-pollution sensors. Nature 2016;535:29–31. http://dx.doi.org/10.1038/535029a.

[9] Munir S, Mayfield M, Coca D, Jubb SA, Osammor O. Analysing the performance of low-cost air quality sensors, their drivers, relative benefits and calibration in cities—a case study in sheffield. Environ Monit Assess 2019;191(94). http://dx.doi.org/10.1007/s10661-019-7231-8.

[10] Spinelle L, Gerboles M, Villani MG, Aleixandre M, Bonavitacola F. Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide. Sensors Actuators B 2015;215:249–57. http://dx.doi.org/10.1016/j.snb.2015.03.031.

[11] Alphasense website, https://www.alphasense.com (last access on 25 May 2020).

[12] Plantower website, https://www.plantower.com (last access on 25 May 2020).

[13] Tecnosens website, https://www.tecnosens.it (last access on 25 May 2020).

[14] 2B Technologies website, https://www.twobtech.com (last access on 25 May 2020).

[15] Environnement website, https://www.environnement-sa.com (last access on 25 May 2020).

[16] Borrego C, Costa AM, Ginja J, Amorim M, Coutinho M, Karatzas K, et al. Assessment of air quality microsensors versus reference methods: The eunetair joint exercise. Atmos Environ 2016;147:246–63. http://dx.doi.org/10.1016/j.atmosenv.2016.09.050.

[17] Yi WY, Leung KS, Leung Y. A modular plug-and-play sensor system for urban air pollution monitoring: design, implementation and evaluation. Sensors 2018;18(1). http://dx.doi.org/10.3390/s18010007.

[18] Ji B, Zheng W, Gates RS, Green AL. Design and performance evaluation of the upgraded portable monitoring unit for air quality in animal housing. Comput Electron Agric 2016;124:132–40. http://dx.doi.org/10.1016/j.compag.2016.03.030.

[19] Webpage for SentinAir costs, https://github.com/domenico-suriano/SentinAir/tree/master/hardware/Bill%20of%20Materials (last access on 22 July 2020).

[20] Raspbian Stretch Lite OS download webpage, http://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2018-06-29/ (last access on 9 July 2020).

[21] Perkel JM. Programming: Pick up python. Nature 2015;518(7537):125, http://www.nature.com/doifinder/10.1038/518125a.

[22] IEEE-SPECTRUM. The 2020 top programming languages. 2020, https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2020 (accessed on 4 August 2020).

[23] Arduino Uno official webpage, www.arduino.cc (last access on 22 July 2020).

[24] Repository webpage for LCSS USB adapter, https://github.com/domenico-suriano/SentinAir/tree/master/hardware/lcss (last access on 22 July 2020).

[25] Example of SentinAir output data files, https://github.com/domenico-suriano/SentinAir/tree/master/data_examples (last access on 2 September 2020).

[26] Scikit-learn libraries webpage, https://scikit-learn.org/stable/ (last access on 22 July 2020).

[27] Raab S, Chandra MW. Mobile Ip technology and applications. Cisco Press; 2005.

[28] Shneyderman A, Casati A. Mobile VPN, delivering advanced services in next generation wireless systems. WileyPublishing Inc; 2003.

[29] Dataplicity website, https://www.dataplicity.com (last access on 25 May 2020).

[30] Pitunnel website, https://www.pitunnel.com (last access on 25 May 2020).

[31] Remote.it website, https://remote.it (last access on 25 May 2020).

[32] Lighttpd website, https://www.lighttpd.net/ (last access on 25 May 2020).

[33] Robinson D, Coar K. Rfc3875: the common gateway interface, version 1.1. 2004, https://tools.ietf.org/html/rfc3875 (last access on 25 May 2020).

[34] Connoly D, Quin L. CGI: Common gateway interface. 2011, https://www.w3.org/CGI/ (last access on 25 May 2020).

[35] Sp61 sensor datasheet on the Nissha-Fis website, http://www.fisinc.co.jp/en/common/pdf/A1320301-SP61%20seriesE_P.pdf (last access on 2 September 2020).

[36] Alphasense ISB webpage, http://www.alphasense.com/index.php/products/support-circuits-air/ (last access on 22 July 2020).

[37] POREM project website, http://www.lifeporem.it/index.php/en (last access on 22 July 2020).