Making Sense

Advances and experiments
in participatory sensing

SMART CITIZEN TOOLKIT UPDATES

D2.4

DELIVERABLE

PROJECT ACRONYM	GRANT AGREEMENT #	PROJECT TITLE
Making Sense	688620	Making Sense

DELIVERABLE REFERENCE NUMBER AND TITLE

D2.4
Documentation on toolkit add-ons

Revision: v7.0

AUTHORS

Guillem Camprodon (IAAC)

Victor Barberán (IAAC)

Maximo Perez (IAAC)

Project co-funded by the European Commision within the Call H2020 ICT2015 Research and Innovation action

DISSEMINATION LEVEL

✓	**P**	**Public**
	C	Confidential, only for members of the consortium and the Commission Services

REVISION HISTORY

REVISION	DATE	AUTHOR	ORG...	DESCRIPTION
v1.0	10-03-2017	Guillem Camprodon	IAAC	First Draft
v2.0	30-03-2017	Victor Barberán and Maximo Perez	IAAC	Technical reviewing
v3.0	17-04-2017	Emma Pareschi	WAAG	Content review
v4.0	23-04-2017	Guillem Camprodon	IAAC	Integration of contributions
v6.0	26-04-2017	Gui Seiz	IAAC	Content editing

STATEMENT OF ORIGINALITY

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

INDEX

INTRODUCTION

Since 2012 the Smart Citizen project has aimed to develop tools to support participatory sensing. The two main outcomes of the project are the Smart Citizen Platform and the Smart Citizen Kit, both designed following an open source approach.

The following document describes the effort carried out on Making Sense to develop and test the Smart Citizen Kit platform on real scenarios during the different pilots developed. This effort was carried out with the primary objective of creating a set of tools and resources around the SCK 1.5 that together with the previous *D2.3 Documentation on firmware to integrate sensors* allows communities to develop their own sensing frameworks and strategies.

The deliverable structures in two main sections: The Smart Citizen Kit 1.5 and Beyond the SCK 1.5. The first one covers the whole Smart Citizen Kit 1.5 ecosystem including hardware and software but also the enclosures, onboarding tools and other developments carried out as part of the project. This is built on the Open Source nature of the project and it is aimed at supporting the future development of new applications for the Kit, from new sensors to new deployment strategies. The second section exemplifies how the presented ecosystem is used to integrate new features that go beyond the Smart Citizen Kit and is aimed to support and inspire others towards new contributions.

Note: This deliverable has its own dedicated Github repository[1] containing all the tools and examples explained below.

1 Smart Citizen Toolkit repository https://github.com/fablabbcn/smartcitizen-toolkit

1

THE SMART CITIZEN KIT 1.5

The Smart Citizen Kit 1.5 is the conclusion of more than 2 years of development and the experience acquired after more than 1000 people who used the device previously all over the world. This new design had been partially supported by the Making Sense project, specially on the validation and integration phase.

In early 2013 the first version of the Smart Citizen Kit was released, the SCK 1.0. The design set some of the core aspects of the hardware design (Data Board + Sensor Board, WiFi enabled, Arduino compatible) but was few more than an early prototype. By the end of the same year we released the SCK 1.1 with improved sensors and reliability. This had been the reference design for the past 3 years till the SCK 1.5 we present here.

This primary design goals of this new Kit is to maintain the core open source values of the previous generations while increasing the extensibility and reducing the entry barrier. This had been achieved by conceiving the new SCK as a whole where the hardware, the firmware but also the onboarding process for users was designed as a whole.

The following sections describe the core features of the new Kit focusing on those related to the extensibility and modularity of the platform.

Fig 1a, 1b. Making Sense Barcelona pilot participants assembling a Smart Citizen Kit 1.5

1.1 Hardware Design

The SCK 1.5 as on previous versions combines a Data Board with a Sensor Board. This design ensures modularity while maintains costs down by keeping all the sensor on a single board.

However based in our previous experience while running workshops with previous versions of the Kit we know we needed to provide an easy way to expand the Kit quickly with off the shelf sensors or customized Arduino boards. We also know this was not limited to sensors but also actuators from motors to displays. This had been achieved by providing a I2C bus with a plug-n-play connector already found in existing products from Seeedstudio.

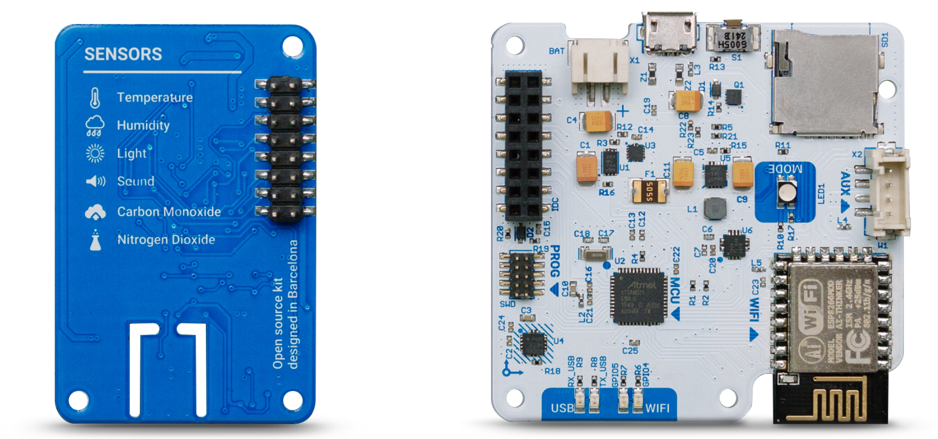


Fig 2. The SCK 1.5 Data Board and Sensor Board

1.1.1 The Data Board

The Data Board is a data logger in asteroids allowing data to be published online via Wi-Fi and/or stored locally on the SD card.

It also takes care of the power management supporting a Li-Po battery and USB or Solar Panel charging. All the different modes and functionalities can be controlled via a single push button and an RGB LED for status feedback. The board is powered by an ARM M0+ 32-bits, the Atmel SAM-D21, combining the low power consumption of the ARM M0 family with the power of a 32-bits processor with 32KB of RAM and 256KB FLASH memory. Compared to previous versions of the Kit this offers up to 8 times more memory allowing custom firmware for add ons and new sensor board to be integrated without constraints.

The Data Board connects to the sensor board via the same pinout standard found in previous versions of the Kit providing power, analog and digital communications (ADC, GPIO, I2C, VCC). Further information about the design of custom sensor boards can be found in MS D2.3 Documentation on Firmware to Integrate Sensors.

The Wi-Fi connectivity is provided by an Espressif ESP8266. Compared to the modules used on previous versions this runs custom firmware as the core processor. This allows for further customization of the whole communication stack easily.

1.1.2 The Urban Sensor Board

The Urban Sensor Board in the SCK 1.5 is an upgraded version of the original SCK sensor board aimed at measuring environmental conditions in cities.

The board features a digital temperature sensor, a lux ambient light sensor, a dB C sensor based on a MEMS microphone (the final commercial release of the board will also offer dBA) and a Carbon Dioxide and Nitrogen Dioxide sensor based on a robust MEMS sensor for the detection of pollution from automobile exhausts.

Compared to previous version of the board the main work has been focused on reengineering the noise and air pollution sensors. Regarding noise monitoring we fully reengineered the noise sensor to use a MEMS microphone reducing the power consumption and increasing the sensitivity. The design has been tested on two of the Barcelona Making Sense pilots and currently is on its final engineering phase towards the final commercial production of the board. This final step involves a new fully digital I2S MEMS microphone capable of providing sound pressure in dB A scale rather than just dB C. In terms of air pollution we follow the same approach as on the previous design by using the SGX MEMS MIC-4514 in order to maintain a low cost design. In this direction the work has been focused on the firmware side controlling the sensor based on the latest work by Spinelle et al (2017). The design is currently under scientific field validation as part of the iSCAPE project under the European Community's H2020 Programme Grant Agreement No. 689954.

As already described on the previous D2.3 Documentation on firmware to integrate sensors we took seriously the idea other sensor metrics might be required on future sensor campaigns and uses. In this direction we built the Add-on connector on the Data Board and we document on this deliverable how new custom and commercial sensors can be used. We also provide the standard sensor board pinout and firmware for those cases when a full new Sensor Board might need to be developed for a full new set of sensors.

1.1.3 Hardware details

DATA BOARD	
MCU	**SAMD21 (ARM M0+ 32-bits)**
CLOCK	32Mhz
FLASH	256KB
RAM	32KB
WIFI	Espressif ESP8266-12E (FCC)
OTHER BUILT-IN PERIPHERALS	Micro SD CARD, RTC (Real TIme Clock), PV + USB LiPo Battery Charger, wCompass+Accelerometer, Groove Add On connector, RGB LED
FIRMWARE	Repository
DESIGN FILES	v1.5
AMBIENT BOARD	
LIGHT	**BH1730FVC**
Type	*Digital Ambient Light Sensor*
Units	*Lux*
Datasheet	*BH-1730FCV.pdf*
TEMP	**SHT21**
Type	*Digital Temperature and Relative Humidity Sensor*
Units	°C
Datasheet	SHT21.pdf
HUMIDITY	**SHT21**
Type	*Digital Temperature and Relative Humidity Sensor*
Units	*% Rel*
Datasheet	SHT21.pdf
NOISE	**SPU0414HR5H**
Type	New MEMS microphone with envelop follower sound pressure sensor Currently moving to a fully digital I2S version for the final release offering dB A
Units	*dB C (dB A)*
Datasheet	SPU0414HR5H.pdf
CO	**MiCS-4514**
Type	*MOS CO and NO² gas sensor*
Units	*kOhm (ppm)*
Datasheet	*MiCS-4514_CO_NO2.pdf*
NO²	**MiCS-4514**
Type	*MOS CO and NO² gas sensor*
Units	*kOhm (ppm)*
Datasheet	*MiCS-4514_CO_NO2.pdf*

For more information visit http://docs.smartcitizen.me

1.2 Firmware Design

The Smart Citizen Firmware is entirely developed in C++ and is built on top of the Arduino Zero Core for the Atmel SAMD21 MCU being fully compatible with existing Arduino Libraries supporting the Arduino Zero family. All the firmware is open and available under a GPL v3 license in Github[1].

The firmware is developed using the Platformio IDE simplifying the management of dependencies and environments. This IDE was chosen in favour of the Arduino IDE for being a much more robust tool while still being easy to install and fully open-source. However we plan to offer support to the Arduino IDE in the future to help new users with experience with this tool.

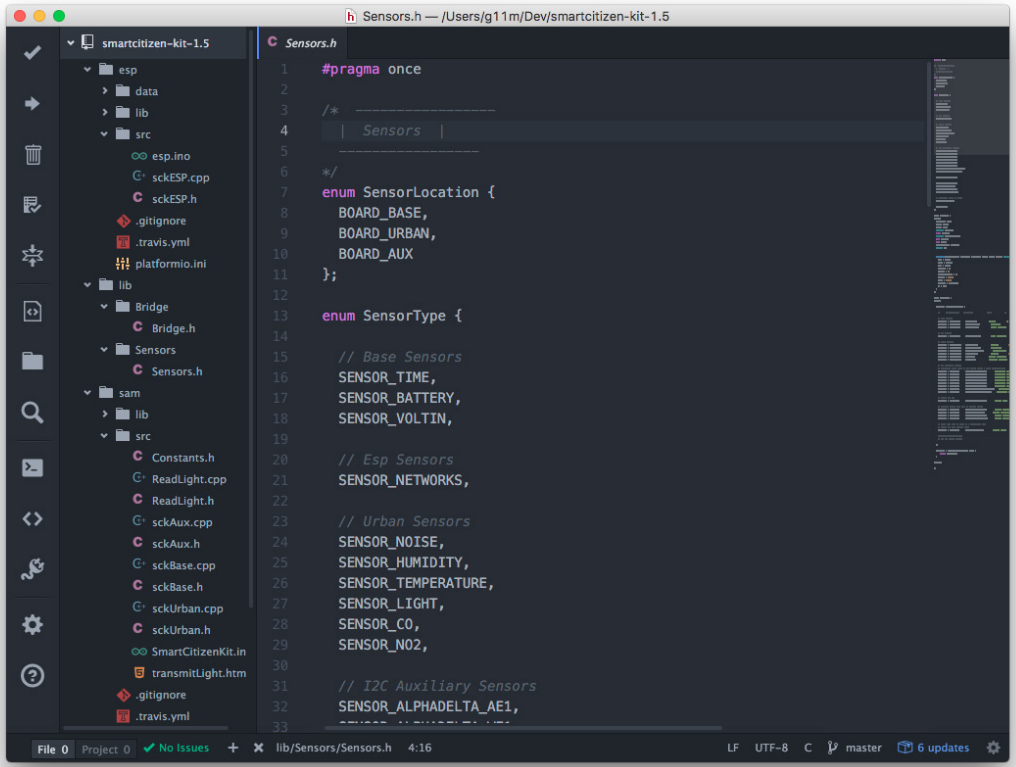


Fig 3. The SCK 1.5 firmware on the Platformio IDE

The firmware supports both processors: the core SAM D21 and the ESP8266. The

1 Check the Hardware specifications on the section above on the Ambient board sensors specifications

development for both is unified by using a set of bridge libraries that exposde a common set of structures and calls in both sides. The following table describes all the main classes and their functionality:

LIBRARY	USED	FUNCTIONALITY
SCKbase	**Core board - Arduino Zero Core (SAM D21)**	*Manages the SCK Data Board core functions: operation modes, communications, time and power management.*
SCKurban		*Manages the SCK Urban Sensor Board functions: reading sensors and data post processing*
SCKaux		*Manages any addons we connect using the SCK Aux Port. Here as where we can include any functions we want to read any custom addon we develop. If the addon is a sensor we will need to define it too on the SCKsensors library.*
SCKsensors	**Shared**	*This is a cross shared library that provides the configuration for all the sensors on the kit for both processors.*
SCKbridge		*Provides a common set of functions for both processors to communicate one with the other efficiently.*
SCKesp	**WiFi module - Arduino ESP Core (ESP 8266)**	*Manages all the network related functions: Wi-Fi connectivity, Captive Portal configurations, MQTT communications.*

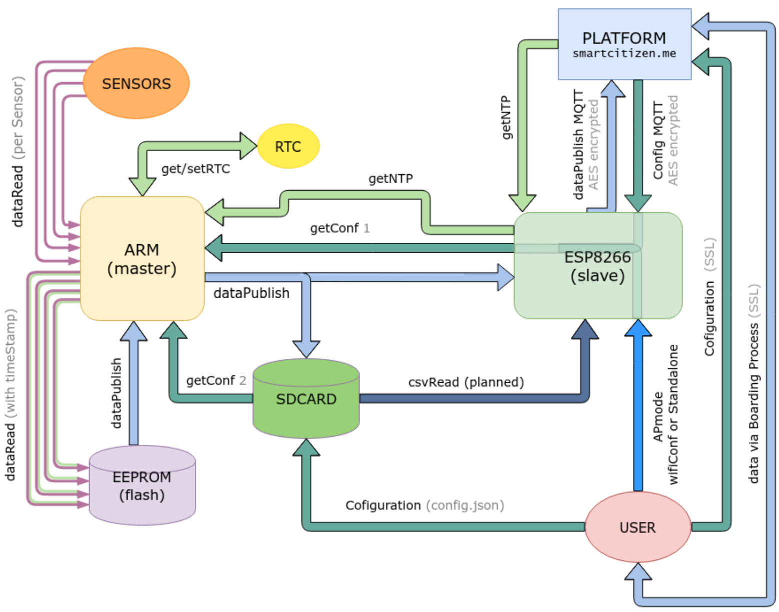


Fig 4. The SCK 1.5 internal firmware structure

1.2.1 SCK Shell

The SCK 1.5 provides a comprehensive command shell over USB allowing people with a minimal experience on Arduino to manage all the kits functionalities. This functionalities are also currently being implemented on the Smart Citizen Platform to support the same actions to be performed remotely on an easy way. The shell also has complete functionalities to avoid spell mistakes while typing the commands.

INSTRUCTIONS

How to use the serial Shell to configure the firmware

1. Use any Serial console as the serial monitor found on the Arduino IDE
2. Connect the SCK 1.5 over USB
3. Select the right Serial Port and open the Serial Monitor
4. Use *help* to list all the commands available.
5. Run any command as *list sensor* or *disable humidity*

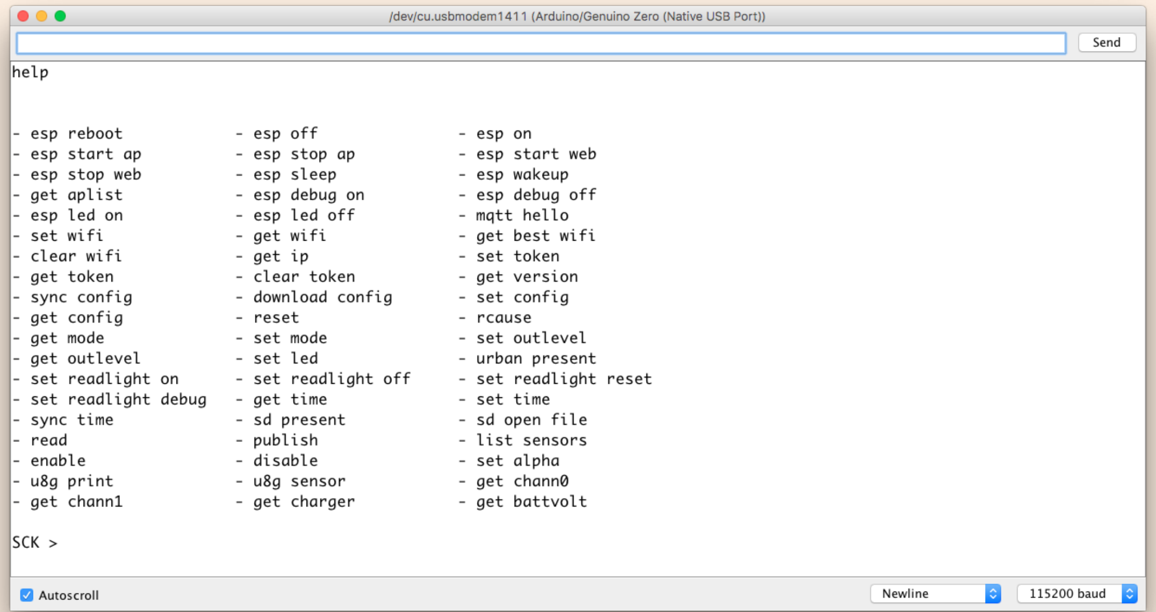


Fig 5. The SCK 1.5 Shell list of commands is shown by the 'help' command

1.2.2 Toolchain

The firmware on the SCK 1.5, both for the ARM and the ESP, can be easily upload via USB. The ARM processor uses the Arduino Zero bootloader and also acts as a bridge to support programming of the ESP module. The whole toolchain involving compilation and uploading of both firmwares is managed by an interactive shell utility based on Platformio. The current instructions support Ubuntu Linux. On the next months we will also release instructions for compiling and updating the firmware using the standard Arduino IDE.

INSTRUCTIONS

Compile and upload the firmware in Ubuntu Linux and Mac OS

Install Platformio[2] and the ESPtool[3]

On Ubuntu Linux use built in Advanced Packaging Tool

```
$ apt install platformio esptool
```

On Mac OS you will need the Apple Developer tools and Hombrew to be installed

```
$ brew install platformio
$ git clone https://github.com/themadinventor/esptool.git
$ cd esptool
$ sudo python ./setup.py install
```

Clone the repository using Git

```
$ git clone https://github.com/fablabbcn/Smart-Citizen-Kit-15
```

Connect an SCK 1.5 over USB and run the compilation and upload script

```
$ cd Smart-Citizen-Kit-15
$ sh ./build.sh
```

2 Platformio website http://platformio.org/

3 ESP tool website https://github.com/igrr/esptool-ck

1.2.3 Advanced features

Compared to previous version the SCK 1.5 does not offer an ISCP interface for programming loading a using a the standard AVR ISP.

Instead it uses a SWD interface as found in many ARM boards. This allows to upload the bootloader but also to use Open OCD to debug the firmware in real time. In order to avoid being tied to any vendor specific SWD hardware we developed low cost programmer using a Raspberry Pi. This means a programmer to flash the SCK 1.5 bootloader is available to anyone for less than 20€.

INSTRUCTIONS

Compile and upload the firmware in Ubuntu Linux and Mac OS

On a Raspberry Pi running Raspbian run the following commands to install OpenOCD

```
$ git clone git://git.code.sf.net/p/openocd/code openocd-code
$ cd openocd-code
$ ./bootstrap
$ ./configure --enable-sysfsgpio --enable-bcm2835gpio
$ make
$ sudo make install
```

Now with OpenOCD install you can reinstall the SCK bootloader by using the configuration you can download from our Github. The connections with the Raspberry Pi are detailed on the figure below.

```
$ sudo openocd -f openocd_unbrick_raspi.cfg
```

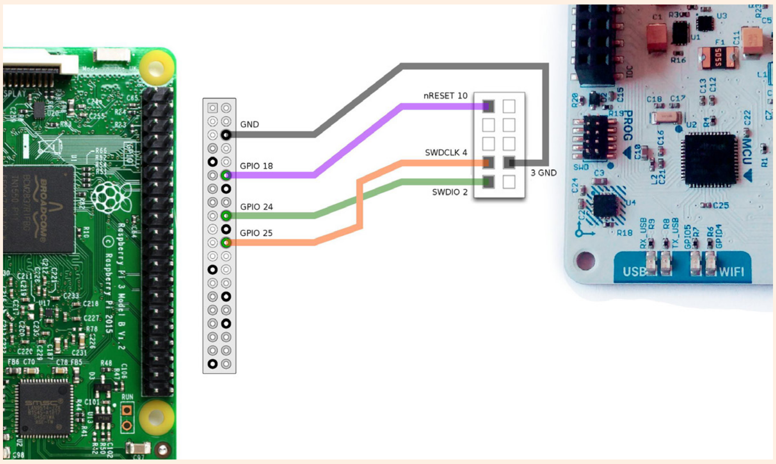


Fig 6. The required wiring between the SCK 1.5 Prog port (JTAG SWD) and a Raspberry Pi pinout

1.3 Packaging design

The SCK 1.5 is not simply a PCB board. Instead is a set of tools to support users on capturing data. The following Kit has been created for the Making Sense project and used in two of the pilots in Barcelona.

Fig 7. The Smart Citizen Making Sense Pack

The following section documents all the carried work while all the fabrication files and technical drawings open and available under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License here

https://github.com/fablabbcn/smartcitizen-enclosures

1.3.1 Smart Citizen Making Sense Kit

The Making Sense Kit has been designed to support users on deploying the SCK 1.5 as this was found to be a critical part of the overall engadging strategy. The following table contains all the parts included on the pack:

PARTS	DESCRIPTION
Smart Citizen Kit 1.5 Data Board	*The SCK 1.5 Core PCB*
Smart Citizen Kit 1.5 Urban Sensor Board	*The SCK 1.5 Urban Sensors PCB*
Smart Citizen Kit Li-Po 2Ah battery	*The standard SCK 2Ah Lithium Polymer battery.*
Smart Citizen Shell Enclosure	*Outdoor case for the SCK 1.5 3D printed on ABS with a laser cut acrylic cover. It also contains 4 plastic commercial joins by Essentra. This components can be easily replaced by standard screws.*
Micro USB Cable	*Micro USB to USB cable to load new firmware and to charge the Kit batteries*
Velcro reusable tie	*Velcro tie to attach the Kit to posts and rails.*
Booklet	*A printed A5 book containing basic information on how to deploy and take care of the Kit*
Clothbag	*A silkscreened bag where to keep all the parts when they are not used.*

The parts found inside the Smart Citizen Making Sense Pack

Fig 8. Some of the parts inside the Making Sense Kit

1.3.2 Smart Citizen Shell Enclosure

The Smart Citizen Shell Enclosure is the natural evolution of the Smart Citizen casing with an improved airflow and an easier to assembly design.

While previous versions of the case required the use of screws this can be assembled using just press fittings. The design is also fully open source and can be fully made on a Fab Lab using 3D printing and laser cutting. This is aimed local communities to improve it and adapted to support new sensors and features.

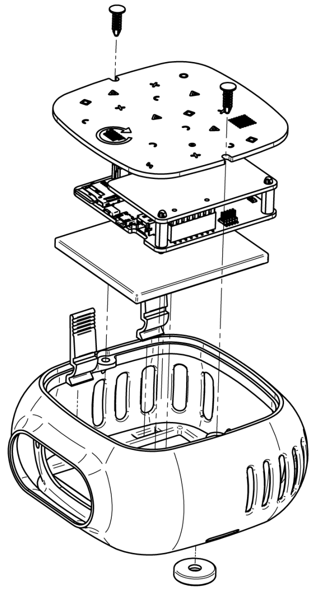


Fig 9. The SCK 1.5 3D printed shell enclosure with the acrylic cover and th plastic fittings
Fig 10. The SCK 1.5 shell enclosure assembly process

1.3.3 Addons support: Stevenson screen

While thinking on future add ons to support better air quality measurements with the SCK we found the need to rethink the Smart Citizen Shell Enclosure to improve the internal airflow, waterproofing and modularity.

Even this is not yet a fully finished design we document it here to support others while developing new add ons for the Kit.

Fig 11. A 3D render of the Stevenson design

The chosen design is based on the well know Stevenson screen design used to protect weather instruments since the XIX century. The enclosure is built out of multiple louver frames that can be stacked one on top of the other. This allows the enclosure to grow in one axis to accommodate different sensor configurations. The louvered design protects the Kit inside against precipitation and direct heat radiation from outside sources, while still allowing air to circulate freely around it.

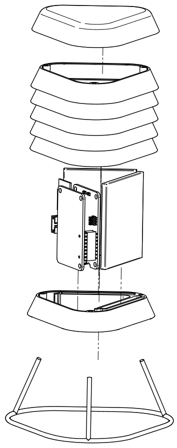


Fig 12. The Stevenson enclosure supporting sensor add ons and two batteries for extended battery life

1.4 User Experience

A critical issue addressed while building the Smart Citizen Kit 1.5 has been to look at the use of the technology as a whole and not simply as a printed circuit board with sensors.

We think this is critical towards infrastructuring sensing communities and much of the work described on the sections above had this aspect as a top priority.

One of the particular difficulties while designing the different users processes was keeping things simple and understandable for citizens at the same time than providing a flexible framework that could deal with the open nature of the project. This is especially critical in order to support add-ons by the community to address local needs and sensing requirements.

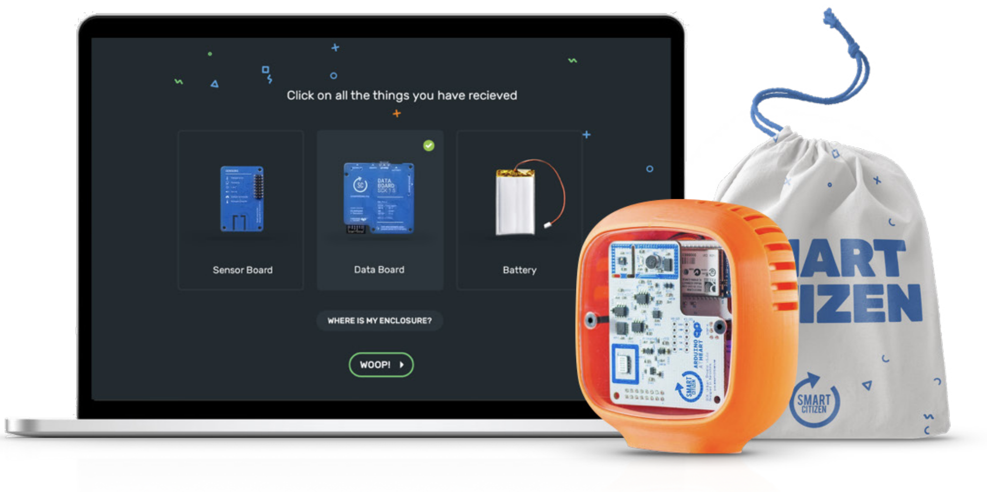


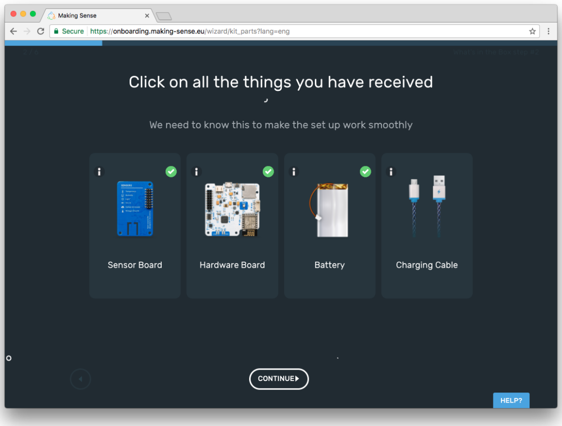
Fig 13. The Smart Citizen onboarding developed for Making Sense

1.4.1 Onboarding Users

The first problem to tackle when involving citizens with sensors is how to support them technically on the physical setup and software configuration of the Smart Citizen Kit.

In order to address this problem the Onboarding Framework was developed as part of the Making Sense Online Toolkit as it was already described on D2.1 Online Technical Toolkit.

The Onboarding Framework is an website for desktop and mobile built using Angular JS. It provides a robust design and technical framework where information is presented as an step by step guide support static content but also dynamic forms to dynamically interact with the users.



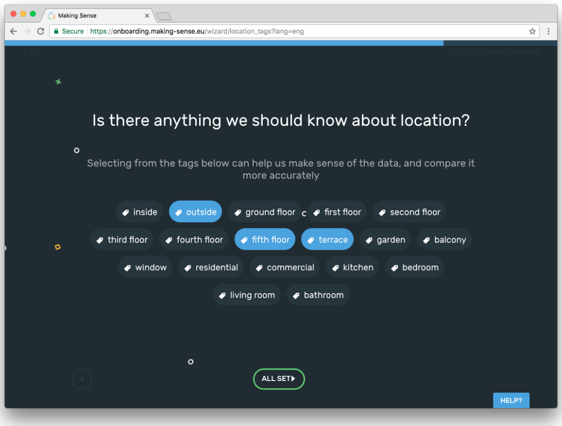
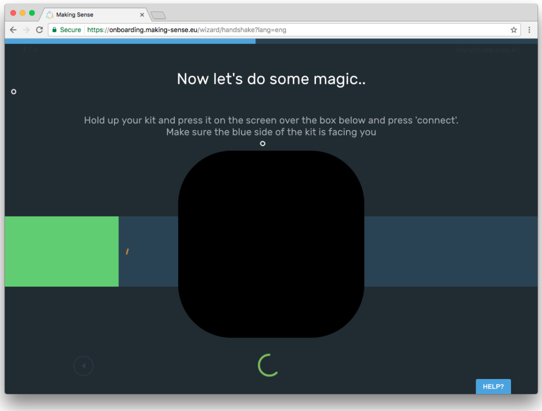


Fig 14. The different sections of the onboarding web page developed and tested on the project

One of the most critical features supported is to allow the user to send his Wi-Fi credentials to the Smart Citizen Kit along with pairing it with his account. The flexibility of the design allows to support multiple processes depending on the user constraints. Currently two processes are supported:

> The light process uses light patterns generated by the user computer or mobile screen to send the information to the SCK avoiding any need for a physical connection or special configuration on the user computer.



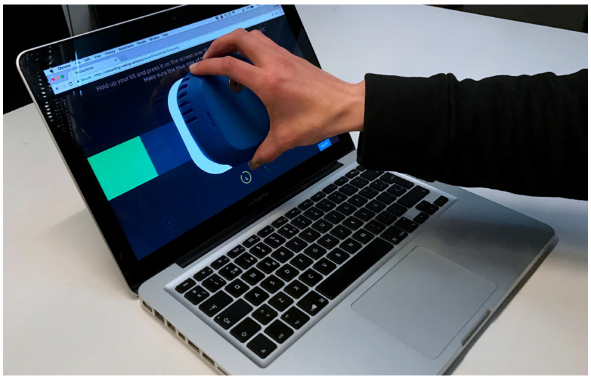
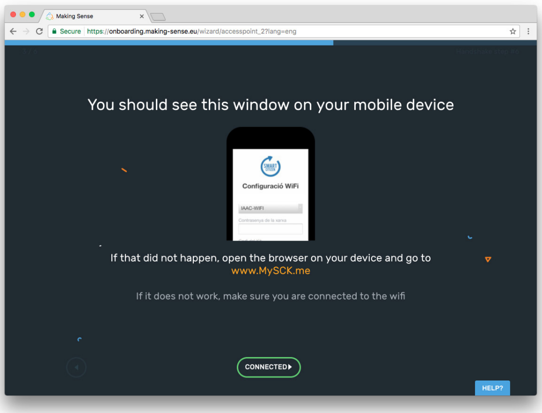


Fig 15-16. The Light setup process

The access point process is similar to the one found on other IOT devices and requires the user to join a Wi-Fi network created by the SCK where the user is asked to provide his credentials on a captive portal.



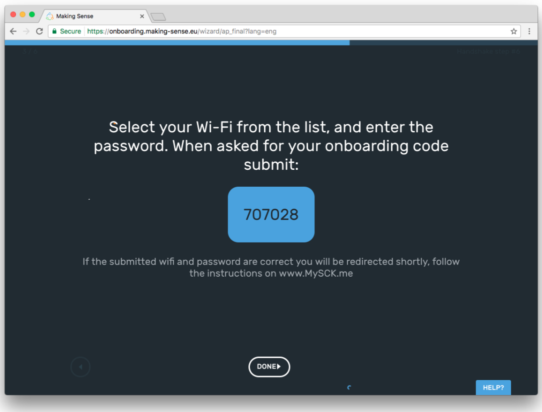


Fig 17-18. The AP mode process

On the same direction the framework can be used to support any custom process that future add-ons might require. This can be to complex interactive functionalities that might require

experience on HTML and Javascript but also new static pages with assembly instructions that can be simply added by editing a plain text file. On the same direction multi language support is built-in from the ground up. At the time English, Spanish and Catalan are available.

Below you can see an example of the how the different pages and languages are defined on the configuration JSON file:

```
{
  "index": 1,
  "part": 2,
  "template": "collaborators",
  "url": "making_sense",
  "companyLogo": "app/images/MS LOGO.png",
  "h2": "Making Sense is a project to ...",
  "h4": "We want to help you deploy sensors to help...",
  "segueButton": "SOUNDS GOOD!"
}
```

The onboarding web application is open and available under a MIT license on Github[4] also a live version can be found online[5].

4 https://github.com/fablabbcn/onboarding-app
5 https://onboarding.making-sense.eu/

1.4.2 User Modes

On previous versions of the Smart Citizen Kit the limited memory size constrained the firmware complexity requiring users to upload a different firmware to switch from the different operation modes.

This included for example switching from publishing data online to storing data locally on SD. Users can select the different modes by simply pressing the button (2) while looking at the mode RGB led (1).

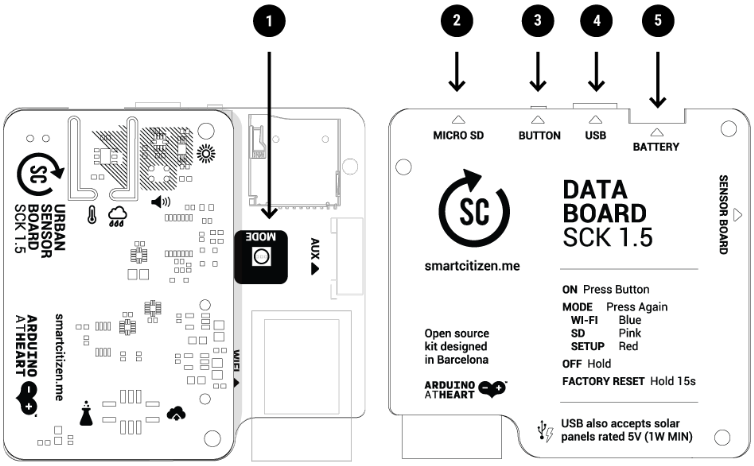


Fig 19. The SCK 1.5 ports and interfaces

The SCK 1.5 overcomes this issue and allows users to select with a simply button the operation mode of the device. At the moment 3 modes are supported:

- **Setup Mode** This mode allows the user to setup the Kit via the light or access point process.
- **Online Mode** This is the normal operation mode of the device where data is recorded and published on-line over Wi-Fi.
- **SD Mode** This mode is designed to work on places where no internet connectivity is available and stores data locally on a Micro SD card.

Further modes can be easily added by expanding the firmware. For example we can create a mode to enable or disable sensors we add over the Aux connection.

1.4.3 Seamless Customization

One of the key features introduced to the Smart Citizen Platform as part of the Making Sense project is the support for multiple sensor types including customize metrics and calibration algorithms per sensors. The feature is called Device Blueprints and was already described on the previous D2.3 Documentation on firmware to integrate sensors. During the past year this feature was successfully tested during the second Making Sense pilot in Amsterdam by WAAG Society.

The following JSON show the minimum information a blueprint needs. (For more information see D2.3):

```
{
   “name”: “The Frog”,
“description”: “Custom Arduino Humidity Sensor”,
   “slug”: “ms:0,5”,
   “components”: [{
           “map”: “hum”,
           “equation”: “(125.0 / 65536.0  * x) + 7”,
           “sensor”: {
                   “name”: “HPP828E031”,
                   “description”: “Humidity”,
                   “unit”: “%”,
                   “measurement”: {
                           “name”: “relative humidity”,
                           “description”: “Relative humidity is a measure...”
                   }
           }
   }]
}
```

This feature is critical towards supporting new sensors users might like to include via the Aux port or even by designing new sensor boards. However in order to simplify this process we made the firmware already able to detect different sensor configurations as for example an Seedstudio Grove Analog sensor connected over the Aux port.

1.5 Data tools

A collection of small tools for visualizing data and engaging communities during sensing campaigns has also been developed as part of the Smart Citizen Toolkit.

This includes dashboards for visualizing the status of the kits on a pilot or tools for printing plotter sized charts for participants to explore the data easily. The tools add on the ones described previously on D2.3 and they are aimed at local community technical leaders to customize them to match their needs and requirements.

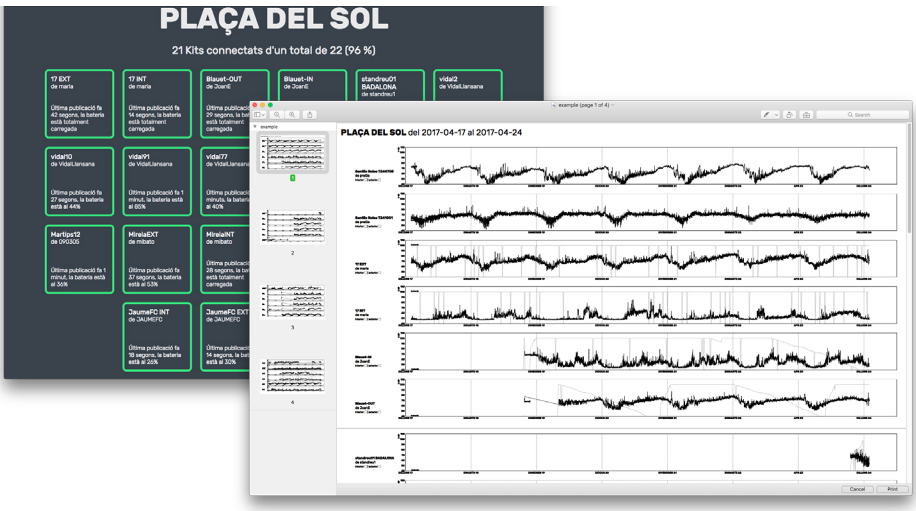


Fig 20. Screenshots of some of the available resources

All of them are also available as part of the Smart Citizen Toolkit repository on Github[6]

6 https://github.com/fablabbcn/smartcitizen-toolkit

2
BEYOND THE SCK 1.5: ADD-ONS

As described on the previously the Smart Citizen Kit 1.5 is everything but a black box.

We hope that even if certain customizations or add ons might require advanced technical skills the documentation available makes it easy for communities to grow and built upon the technology. In this sense on the following section we provide examples of customizations and add-ons developed to the SCK 1.5 towards supporting others on the same task.

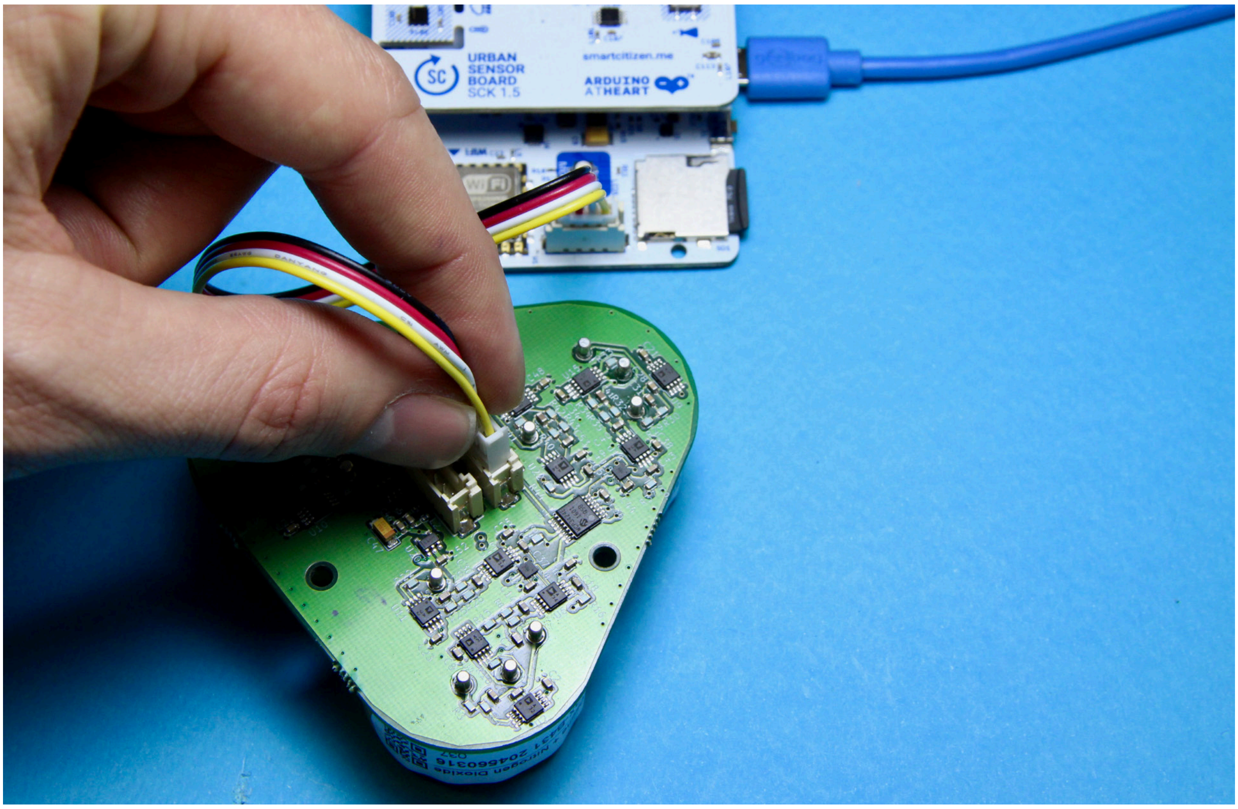


Fig 21. A custom add-on being plug to the SCK 1.5 using the Aux connector

2.1 Third party addons

A critical feature of the new SCK 1.5 is its compatibility with off the shelf addons produced by well know open hardware companies as Adafruit or Seeedstudio.

The following section describes add-ons already implemented on the latest firmware release. This means this accessories are fully plug'n'play and do not require any more changes than a restart on the Kit to start playing with them. Here we can take full advantage of the kit Shell described in section 2.2. Later user can change and adapt the behaviour by doing minor edits on the firmware.

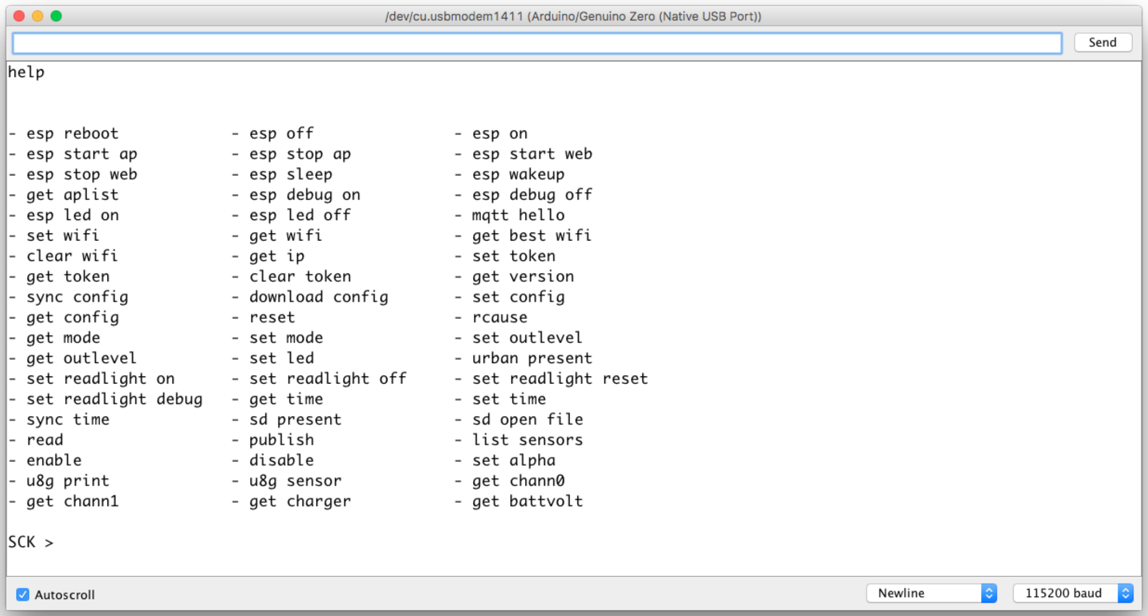


Fig 22. The SCK 1.5 Serial Shell showing the list of sensor supported by the current firmware after typing the list sensors command

EXAMPLE

Adding a Seeedstudio Grove Analog to Digital sensor

The Aux port on the Smart Citizen Kit 1.5 comprises is compatible with the Seeedstudio Grove I2C connector as the standard connector. By using a Grove - I2C ADC module we can then read any of the Seeedstudio Grove Analog sensors.

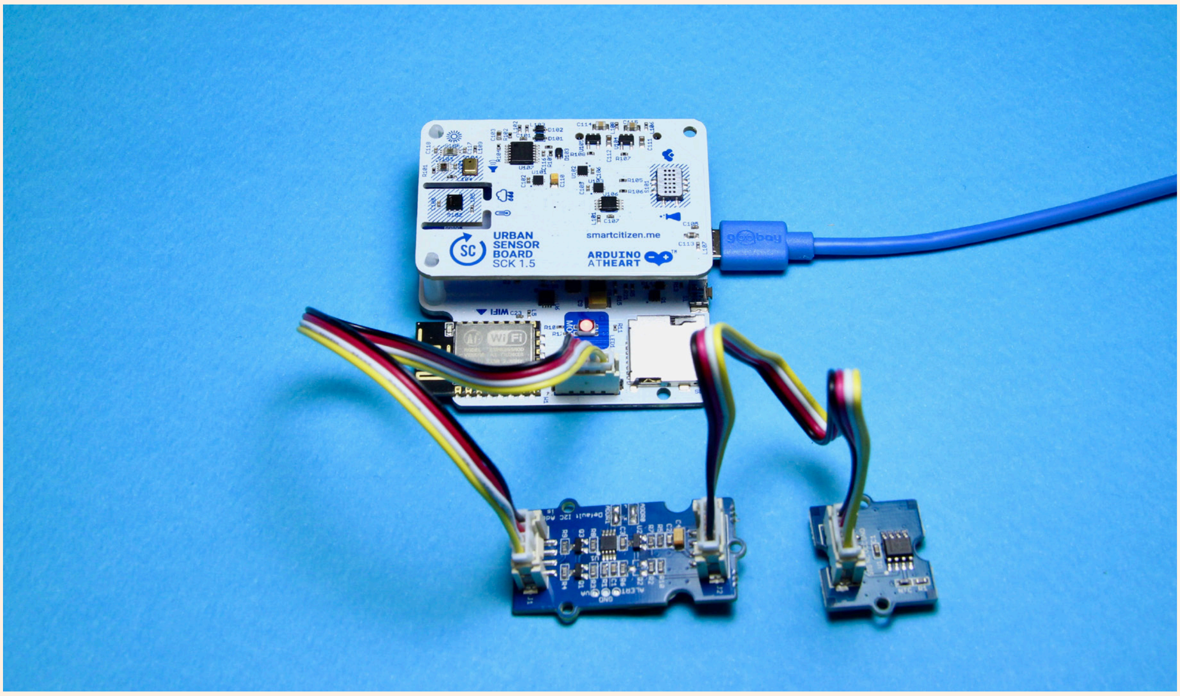


Fig 23. The Seeedstudio ADC with an analog Temperature sensor connected to the SCK 1.5 ready measure indoor and outdoor temperature.

Once the Grove - I2C ADC module is connected we can connect any of the available modules.
The Analog to Digital converter will give us back the connected voltage of the sensor with a resolution of more than 1mV (12-bits ADC). On the following example we use a simple button that can be use by the user to record events together with all the other sensors. The module is automatically detected once the kit is restarted and the data will start to be read and publish online as if it was any other sensor available on the kit. We can start reading the consumption over the serial port by using the Shell as described in section 2.2 using the command read Groove ADC and enable Groove ADC.

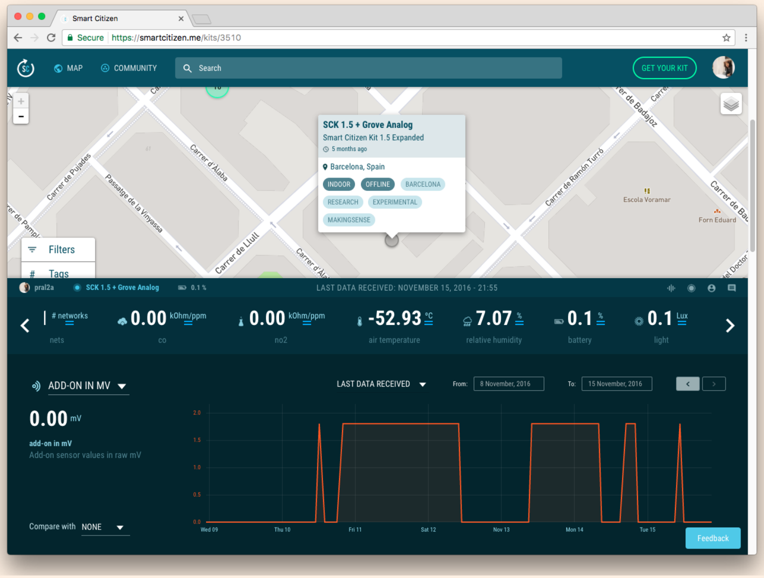


Fig 24. The Smart Citizen Platform showing the Seeedstudio ADC as an extra sensor on a SCK 1.5

By using the Groover - I2C ADC we use should be able to use sensors from the following list of sensors.

Grove - Water Sensor
Grove - Magnetic Switch
Grove - Alcohol Sensor
Grove - Grove - PH Sensor
Grove - Differential Amplifier
Grove - Electricity Sensor
Grove - Sound Sensor
Grove - IR Distance Interrupt
Grove - Tilt Switch*
Grove - Moisture Sensor
Grove - PIR Motion Sensor

Grove - Infrared Reflective Sensor
Grove - Light Sensor
Grove - Dust Sensor
Grove - Air quality
Grove - Gas Sensor
Grove - Temperature Sensor*
Grove - Air Quality Sensor
Grove - Gas Sensor(O_2)
Grove - HCHO Sensor
Grove - Collision_Sensor

*Indicates the sensor has been successfully tested using the described approach.

Also by doing a bit of soldering we can change the patch Grove I2C ADC to change the address. This combined with the Grove - I2C Hub will allow you to use up to 9 Grove I2C ADC. The detailed documentation can be found on the Smart Citizen Toolkit repository.

EXAMPLE

Adding a Seeedstudio Grove Display

Even the Kit is designed to publish data on-line and the Smart Citizen API can be used to create custom web visualizations sometimes we might find the need to have display on the same Kit to display realtime data on-site. This can be simply achieved by connecting an standard Grove - OLED Display 0.96". The display is part of Seeedstudio Grove I2C modules. This modules are fully compatible with the Kit and just require a minimal customization of the firmware. Currently the OLED Display is already supported by the firmware and more add ons as the Grove - Barometer Sensor or Grove - Temperature and Humidity Pro Sensor will be supported soon.

To use the display we just need to connect it and restart the Kit. We can start the display by connecting over the serial port and using the Shell as described in section 2.2. Be using the command u8g sensor "sensor name" as an example u8g sensor noise will display the decibel value from the microphone on the screen. You can also use sensors list to see all the sensors available

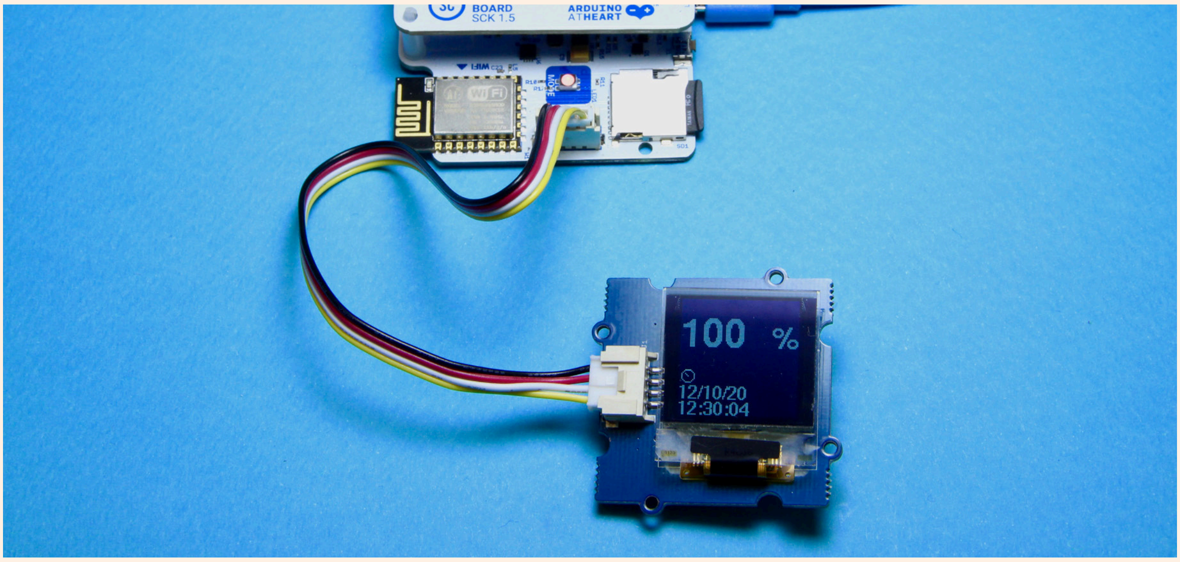


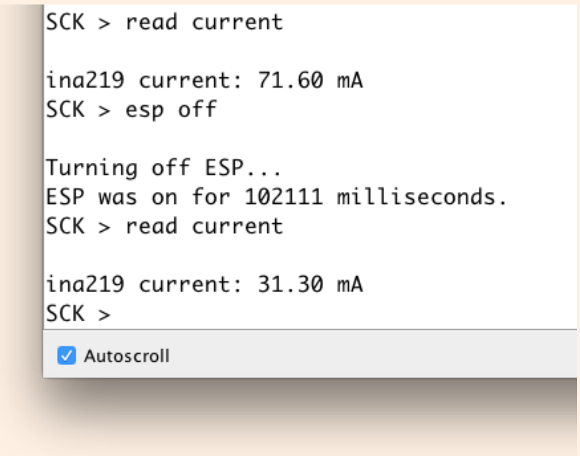
Fig 25. The OLED display showing the dBC recorded by the SCK 1.5 in real time

The data displayed can be easily customized by editing the *Groove_OLED::displayReading* function on the *SCK_Aux.cpp* library.

EXAMPLE

Adding a Current Monitor during firmware development

While implementing new functionalities on the Kit firmware as adding new add ons or simply adjusting the number of post per minute we might like to know in real time what is the power consumption of the Kit to know how much the battery will last.



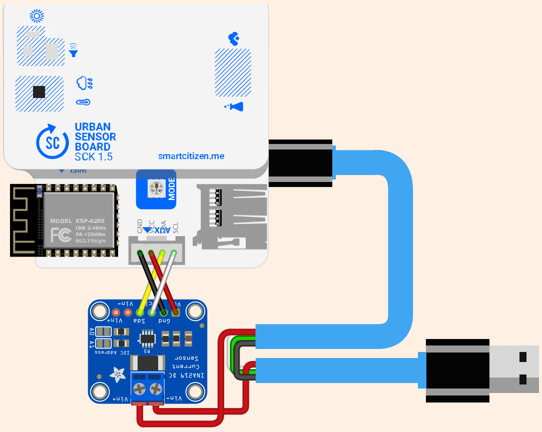


Fig 26. The SCK 1.5 Serial Shell showing the current consumption from the external sensor with the ESP on and off.

Fig 27. The Fritzing schematic of the connections required.

This can be easily done by using the Adafruit INA219 High Side DC Current Sensor (ID: 904). This sensor will require some minimal soldering as can be seen on the image below. The sensor will be detected once the Kit is restarted. We can start reading the consumption over the serial port by using the Shell as described in section 2.2 using the command read ina219 current . You can also use sensors list to see all the other metrics the sensors offer.

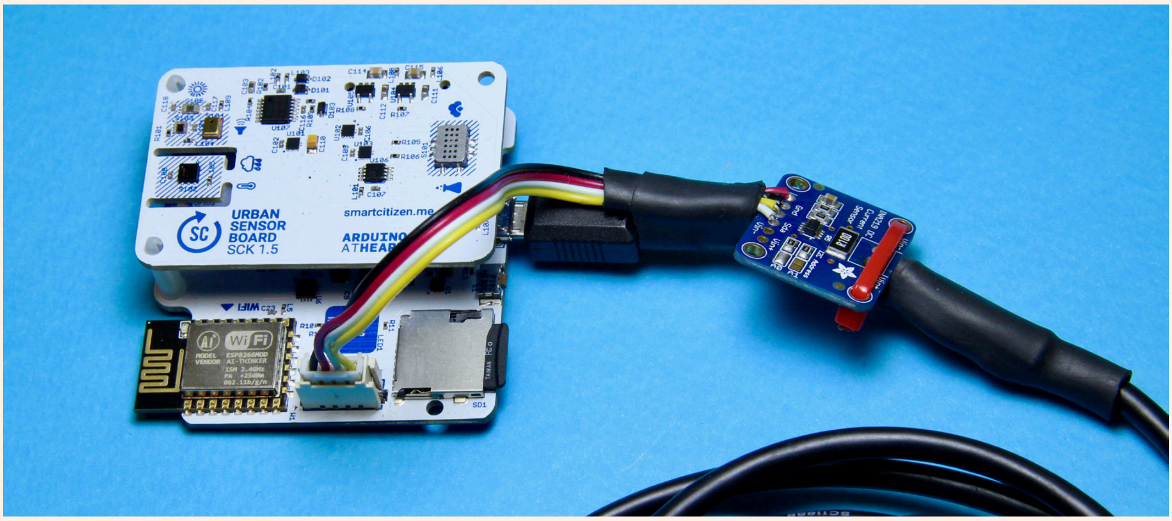


Fig 28. The finished accessory ready to measure the current while developing add-ons for the SCK 1.5

2.2 SCK AQM Addons: Alphasense Delta

The Alphasense Delta Addon has been developed by IAAC as part of the iSCAPE project under the European Community's H2020 Programme Grant Agreement No. 689954. However on the Making Sense project we took the work of integrating the sensor within the SCK1.5 making easier for users to use as part of the whole addons ecosystem.

The board contains 3 Alphasense series B Electrochemical sensors for Carbon Dioxide, Dioxide Nitrogen and Ozone. The board is currently under tests and it will be released open-source as part of the Smart Citizen Toolkit on the next months.

MODEL	GAS
OX-B431	Ozone
NO2-B43F	Nitrogen Dioxide
CO-B4	Carbon Monoxide

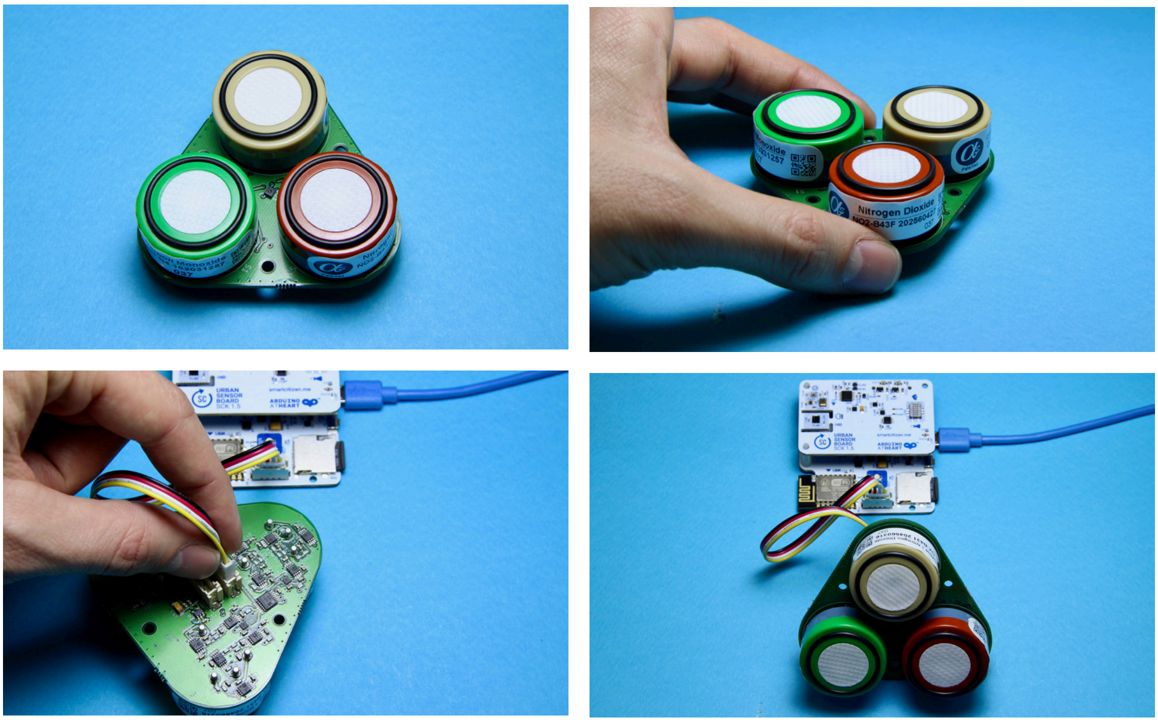


Fig 29. The gas sensor board being connected to the SCK 1.5

2.3 Public Displays: Sensor Box

The Sensor Box is display installation aimed at engaging citizens to discuss about data on the public space.

The installation was built by the Making Sense Barcelona community champions to talk about noise problems affecting neighbours around the Plaça del Sol area in Barcelona. However the installation was built from the ground up to be replicable ad easy to built in oder Fab Labs worldwide. This aims at creating a tool communities can built to engage citizens to discuss about issues by using the data provided by the Smart Citizen Kit.

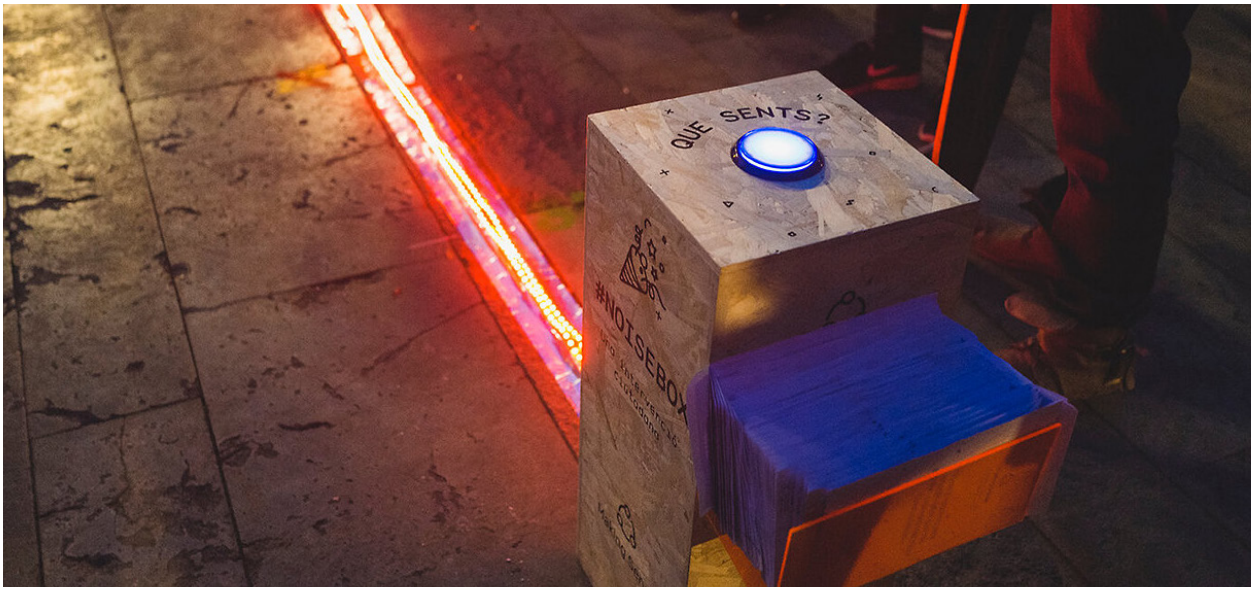


Fig 30. The noise box in action at Plaça del Sol in Barcelona

The device comprises a wooden box equipped with a Smart Citizen Kit to which was a 5 meter long LED strip has been attached. Participants can press a button on the box to trigger the noise sensor. The original installation was battery powered but it can also be plugged to simplify the design and costs.

INSTRUCTIONS

How to create your own Sensor Box

The full list of materials and instructions is available at the Smart Citizen Toolkit repository.

Fig 31. The final assembly process of the box. The transparent cover allows anyone to see what is going on inside.

The */built* folder contains the files for building the installation: NoiseBox.blend the whole installation design in blender, CableClip.stl and Hinge.stl 3D printed parts for the cable clips and hinges used, Acrylic.dxf the acrylic cover lasercut file and noiseBoxSchematic.fzz the wiring diagram for the installation.

The */code* folder contains the Arduino files to drive the installation. The Arduino sketch reads sensor data from an SCK 1.5 over the I2C bus when a user presses the button display the result on a WS2811 addressable LED strip. This code was originally created to display reading from the noise sensor in dB but it can quickly be changed to support any other sensor. It runs on an Arduino UNO but any compatible board can be used.

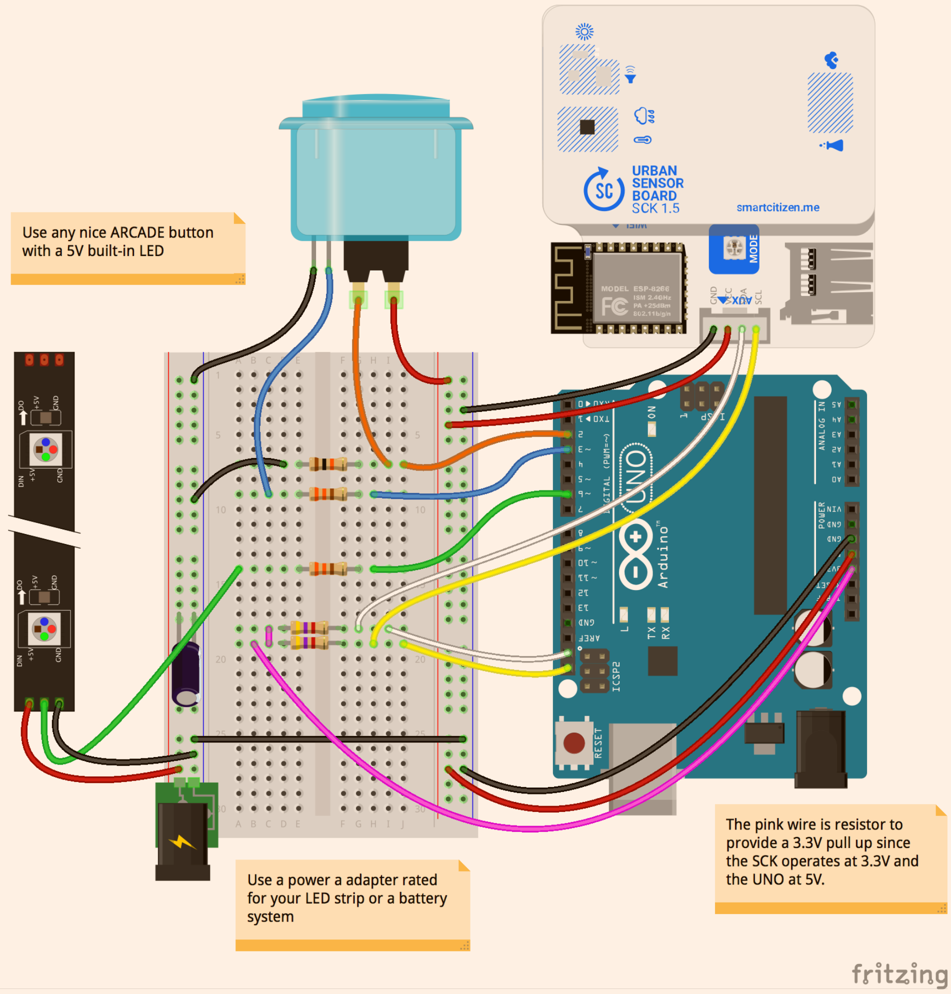


Fig 32. The Fritzing schematic of the connections required. This is also a good reference towards any application requiring an Arduino to talk with an SCK 1.5

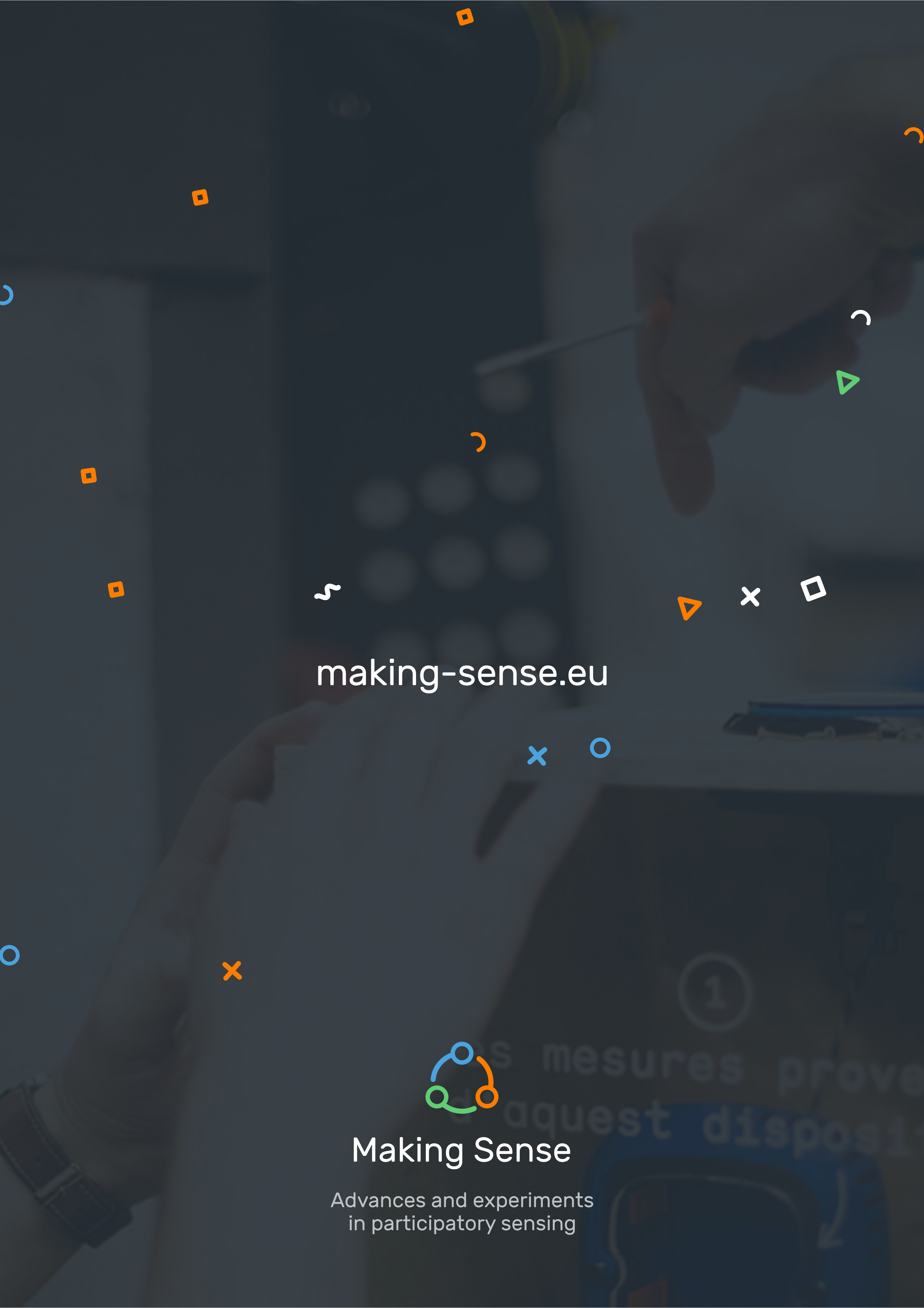
From a technical point of view this is also a good example to learn how to connect a standard Arduino to the SCK 1.5 to expand its functionalities with any Arduino compatible actuator: lights, motors, etc.

3
CONCLUSIONS

We hope the following document shows the enormous effort carried out during the project to develop a technical platform that supports communities on the development of participatory sensing initiatives.

Making Sense has been a really important opportunity to validate the Smart Citizen Kit 1.5 with real users during the Barcelona Pilots prior to move to the final industrialization and commercial exploitation of the platform. However it also allowed us to explore how open source technologies like this can be appropriated by the communities as on the building of the Noise Box installation during the Barcelona Community Champions Pilot.

The current results of this work are also going to be exploded on the next months with the commercial launching of the Smart Citizen Kit 1.5 and on future research projects where the resources documented will be used. We aim that that modularity, standards and open source contribute to create a robust ecosystem where companies and communities can work together co-creating new strategies for sensing and citizen participation.



making-sense.eu

Making Sense

Advances and experiments
in participatory sensing