

# School Gardens for Future Citizens

eSchool Garden 

## FROM REAL TO VIRTUAL GARDENS



### WELCOME TO THE INTERNET OF THINGS

### PART I

## **I. ORGANISE YOUR GARDEN**

*The seed nursery*

*Compost*

*A weather station*

*The plots*

## **II. FIRST STEPS IN PRACTICE**

*Choose an educational toolkit*

*Integrated Development Environments (IDE)*

*Get practice with the code*

*Connecting your Arduino with the world*

*Prepare your database*

*One Arduino = One space*

*Define your semantic annotation*

*Data paths in the cloud*

## **III. YOUR ACTIVITY**

*Take a look at our solution*

*Adapt the code to your cloud*



*eSGarden*

# EXERCISE 1

BY SARA BLANC  
sablac@upv.es

## ORGANISE YOUR GARDEN

*A garden is not just a mishmash of vegetables. A garden has several components that play a part in the daily gardening processes. A school garden has different spaces, which must be planned. To organise our garden, we take four spaces into account: a nursery area for seedbeds, a compost area, a weather station, and the cultivation plots. Each of these spaces will also be organised in the same way in our virtual garden: the app for your students' tablets or phones.*



Do you know what the **Internet of Things (IoT)** is? IoT connects things around the world. The “things” are smart devices such as phones, tablets or watches, or small sensors hidden in our cities, houses, green spaces, or... in your school garden.

To understand how to install IoT in our school garden, first we need to understand the advantages and disadvantages of our small sensors.

Sensor devices are lightweight and handy, so they can be easily embedded in our environment. Moreover, we can build low-cost devices, and by building these devices with our students we also promote many STEAM competences to work with.

On the other hand, as a disadvantage, sensor devices are “low intelligence”, meaning they are unable to store much data and can only carry out processes that are not very complex.

The solution is to connect the “things” with the internet network. Data collected by the sensors is uploaded to the network and travels, either through cables, through the air, or with the support of satellites, to a bigger brain, the “cloud”, which connects all your things.

The IoT cloud receives DATA. It is able to store thousands of pieces of DATA and send information to your smart device: phone, tablet, etc.



## THE SEED NURSERY

*You should consider preparing a suitable space for seeds and seedbeds. Look for a space in a classroom near a window, or in a greenhouse. Students can observe and monitor the process from seeds to plants in addition to maintaining adequate temperature, humidity and lighting conditions. Use recycled materials such as bricks, yoghurt pots, or paper cups.*



## PROPOSALS

*Introduce compost as part of school's recycling actions to care your environment. It is easy to build a compost box.*



*Some crops need a nursery period and the temperature and humidity requirements for seeding differ from their growth cycle after transplanting.*



## COMPOST

*Another space you should take into account will house the compost, which has a twofold objective. First, it will provide the plants with nutrients through an organic process. And second, the composting process is related with physical and chemical principles that connect with curricular learning.*





## A WEATHER STATION

*Physics and mathematics competencies can be practised and learned by gathering data from your garden. There are many ambient parameters that you can measure with a weather station. These parameters affect both the growth cycle of crops and human health. Consider building a weather station. Moreover, this activity can also relate to technology and informatics by introducing your students to a DIY project*

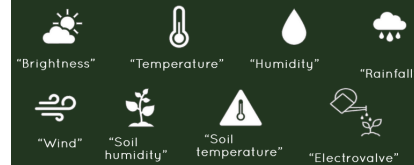
## THE PLOTS

*Finally, the most important part of your garden are the plots. You must divide your garden into plots depending on the area of land dedicated to each crop. Bear in mind that although some crops can share water requirements not all of them can, so divide the garden into plots. We assume that each plot has different water requirements.*



## SENSORS

*Sensors are useful to observe our garden and make important decisions. Considering the school garden structures, we could add sensors such as:*



- **In the seed nursery:** ambient temperature, ambient humidity and brightness.
- **In the weather station:** ambient outdoor temperature, ambient humidity, air quality (e.g. CO<sub>2</sub> or NO), rainfall, noise, brightness, etc.
- **In the compost machine:** temperature and humidity.
- **In the plots:** soil humidity and soil temperature.

*Additionally, some actions could be added to the garden such as making an electro-valve open or close.*



## FIRST STEPS IN PRACTICE:

### CHOOSE AND EDUCATION TOOLKIT

There are a variety of very interesting educational programming kits on the market. In this manual we focus on ARDUINO (<https://www.arduino.cc/>) for several reasons:

- There are many boards that integrate different microcontrollers based on ARM architecture (<https://www.arm.com/>).
- Many wireless shields can be adapted by keeping the digital and analogue pins accessible in order to connect sensors. Some boards have low power consumption, such as the NANO board.
- Some boards have Wi-Fi transceivers like the UNO Wi-Fi or the IoT 33 Nano. Codes can be compiled in any version of board.
- Additional educational resources are available. We will not use them in this manual, but they are an interesting resource.
- Finally, we also value the cost-performance ratio. There are inexpensive solutions that offer good resources for IoT projects.

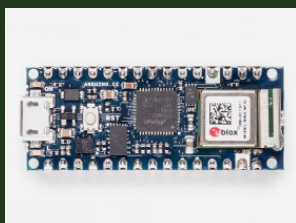
The issue you should consider before deciding on your boards is the compatibility between products. If you decide to build Wi-Fi-based sensor devices by buying boards supplied with a Wi-Fi transceiver, your proposal is simple in “technological” terms. Why? Because you will only work with one communication technology: Wi-Fi.

However, this is not always possible. For example, suppose your garden is quite or very far from the nearest Wi-Fi Access Point to connect your sensors to. In this case you need an alternative to support wireless connection. Currently, there are many IoT communication technologies. Moreover, some technologies are “low power consumption”. Power consumption is a relevant factor, especially if your sensors are fitted with batteries, supplied by power banks, or harvest solar energy.

Among the possible communication options, some technologies will allow you to connect your sensors directly to the internet network. For example, mobile, LTE-M, NB-IoT, satellite, etc. Alternatively, you can install a gateway to reach your IoT cloud, or you can build a bridge (for example, LoRa RF is discussed in the next exercise).

In the exercise we use ARDUINO IoT 33 Nano with Wi-Fi chip

<https://www.arduino.cc/>





## INTEGRATED DEVELOPMENT ENVIRONMENTS (IDE)

Coding and programming require an IDE platform. Vendors offer online resources or desktop packages to install on your own PC. To help you install and configure the ARDUINO IDE, Video 1 and Video 2 present both desktop (offline) procedure and online procedure as well as plugin installation.

[Video 1 – Desktop or Offline ARDUINO IDE Installation \(click\);](#)

[Video 2 – Desktop or Online ARDUINO IDE Installation \(click\);](#)

[Video 3 – Plugin \(click\);](#)

Before writing and compiling any code, you need to “explain” several parameters to your IDE platform:

- The microcontroller of your board (remember that ARDUINO offers a wide range of microcontrollers. Look at your board DATASHEET for the microcontroller within your board).
- Serial port connection. To download new code onto your kit, you must plug your board into your computer with a cable. Configure in the IDE the COM number assigned to the USB cable.
- For ARDUINO, choose the corresponding board in the IDE menu.

Video 4 shows the process with two examples: ARDUINO NANO and ARDUINO UNO Wi-Fi. Video 5 shows the serial port selection option in the IDE menu.

[Video 4 – IDE Configuration \(click\);](#)

It may happen that your board is not installed in your Arduino desktop IDE. How can you update to add a new board?

1. Restart the Arduino IDE.
2. Tools > Board > Board Manager.
3. Find your new board (for example, for Arduino UNO Wi-Fi Rev.2 you need “megaARV Boards” or for Arduino NANO IoT 33 you need "Arduino SAMD 32-bits ARM Cortex M0 +").
4. Install.

[Video 5 – Port selection \(click\).](#)



## CONNECT YOUR ARDUINO WITH THE WORLD

Download the code template available in [esgardenArduino/Arduinolot33\\_Nano\\_WiFi \(github.com\)](https://github.com/esgardenArduino/Arduinolot33_Nano_WiFi). There are four tags:

1. \*.ino includes functions to read your sensors and build a JSON structure.
2. send\_firebase is a template that uses WiFinina and Firebase\_Arduino\_WiFinina libraries.
3. send\_params describes your garden.
4. sensor\_functions need to be changed according to your reading and writing functions.



### ADDING LIBRARIES TO ARDUINO IDE

Wireless mainly refers to packet transmission and reception. Your vendor can supply you with both the hardware and the software libraries for transmission and reception functions. You need the following libraries:

1. Firebase\_Arduino\_WiFinina: to connect Arduino developments with Firebase database supported by Google. If using an alternative IoT Cloud, check for the required communication functions and libraries. Our library is "Firebase Arduino based on WiFinina" based on NINA firmware for ARDUINO Wi-Fi boards.
2. WiFinina: to configure the Wi-Fi transceiver.
3. ArduinoJson: to use the .JSON standard.
4. MemoryFree: to manage microcontroller memory.

```
{
  "DATASLOT_0"
  : { "Type": 5, "Value": [ 74 ] },
  "DATASLOT_1"
  : { "Type": 1, "Value": [ 24 ] },
  "DATASLOT_2"
  : { "Type": 2, "Value": [ 0 ] },
  "DATASLOT_3"
  : { "Type": 6, "Value": [ 250 ] },
  "ID_Node": 1,
  "timestamp": 1605662192815
}
```

JSON is a syntax standard for wrapping our sensor information. We defined a structure according to the garden spaces: General (a weather station), Nursery (seedbed), Compost, and multiple Plots. Each space has a numerical identifier (ID\_Node) Inside each space, the reading of each sensor is a DATASLOT object that identifies the type of sensor and the value read. JSON is necessary in Firebase messages, so adapt the function "MakeJSON" to the syntax required by your Cloud





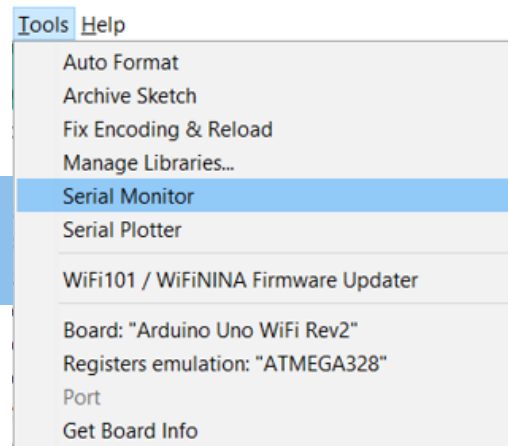
## GET PRACTICE WITH THE CODE

Open the code template available at [esgardenArduino/ArduinoloT33\\_Nano\\_WiFi](https://github.com/esgarden/ArduinoIoT33_Nano_WiFi) ([github.com](https://github.com)).

1. Analyse the code structure of "sensor\_functions"
2. Include libraries, if needed, to support your sensors and actuators. To install new libraries, [click here](#) to see a video tutorial.
3. Implement your functions.

A simple code ReadSensors is also available in [esgardenArduino/ReadSensors](#) repository. Use it for testing purposes.

*Use the "Serial Monitor" to view the sensor data and carry out calibration if needed.*



## SOME SENSORS NEED TO BE CALIBRATED. THIS MEANS TRANSFORMING THE READ DATA INTO USEFUL INFORMATION.



... or use a professional probe ([click here](#))

*To obtain the minimum threshold (maximum humidity), take a full glass of water. Introduce the sensor to the glass and read the value using the Serial Monitor. The value corresponds to the maximum humidity (100%).*

*Now dry the sensor completely and read the data again. The value corresponds to the minimum humidity (0%).*

*Use functions such as the Arduino map function to transform the read value.*



## PREPARING YOUR DATABASE

Create a [Google Firebase](#) real-time data base or alternative Cloud service:

1. Access your [console](#).
2. Create a new project and add a Real-Time Database. You can start in testing mode and change it later.
3. Go to Project Overview > Project Settings > Service Account. There is a default "secret" already created. Add more if needed.

### CONNECTING ARDUINO WITH FIREBASE (SEND\_FIREBASE)

Fill the Wi-Fi SSID and Firebase connection parameters in our Arduino project (send\_firebase):

1. Video 6: [Learn how establish the WiFi connection and Firebase parametres.](#)
2. Video 7: [Find your firebase "secret".](#)
3. Video 8: [Write the secret in Arduino template.](#)

To connect your Arduino to your local network, complete these two simple configuration lines;

```
//CONFIGURATION 1: NEEDED TO WORK WITH WI-FI
#define WIFI_SSID "SSID_school" //Write your school network name or SSID
#define WIFI_PASSWORD "password" //Write the network password
//END CONFIGURATION 1
```

//CONFIGURATION 2: NEEDED TO WORK WITH FIREBASE

```
#define FIREBASE_HOST "databasename.firebaseio.com" //Database name
#define FIREBASE_AUTH "MySecretNumber00000000000000000000000000000000"
#define DELAYFIREBASE 1000
//END CONFIGURATION 2
```

Complete Google firebase connection parameters;

Change your GARDEN name and the ID\_Node to identify your space in the garden.

```
//CONFIGURATION 3: NAME OF YOUR GARDEN; same name as in your APP
String GARDEN = "MySchoolGardenName";
//END CONFIGURATION 3
```

WiFi connection

```
void StartFirebase() {
  while (status != WL_CONNECTED) {
    status = WiFi.begin(ssid, pass);
    delay(10000);
  }
}
```

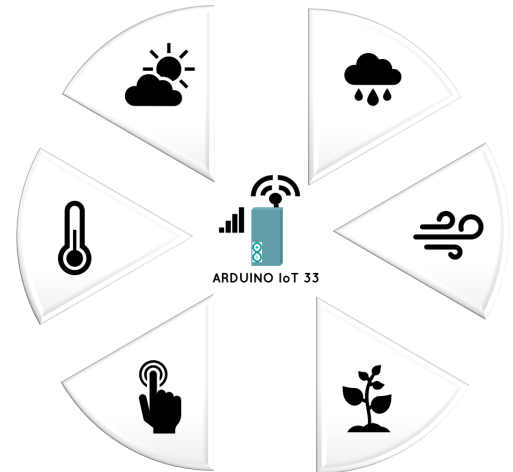
```
String SendFirebase(String Path, String JsonSerialized) {
  Firebase.reconnectWiFi (true);
  if (!firebaseData.bufferOverflow()) {
    if (!Firebase.pushJSON(firebaseData, Path, JsonSerialized)) {
```

Push function to firebase



## ONE ARDUINO = ONE SPACE

A single microcontroller can handle multiple sensors. We propose one microcontroller or ARDUINO for space. Choose different sensors according to the space you want to observe:



## WEATHER STATION

*Include a rich variety of sensors: a weather station is an ideal activity to introduce students to the world of sensors. You will find very different types, with different measurements and magnitudes, such as temperature, humidity, air quality, noise, rainfall, wind, brightness, etc. Take advantage of this variety to connect the garden with subjects such as maths and science.*

*In addition to the electronics, you can build the box and an enclosure to protect it.*

## SEEDBEDS

*One of the most important aspects when establishing a school garden is to choose crops properly. Ambient temperature and humidity requirements differ between the seedbed phase and growing phase. Therefore, we should look for a suitable seedbed space or nursery to maintain optimal germination conditions.*





## COMPOST SPACE

*You can use manual thermometers or make a module, similar to that of a weather station, with temperature and humidity sensors. Build a box and use dust filters to prevent damage to the sensors. Sensors should be embedded in the compost mix, taking temperature parameters as deep down as possible*



## PLOTS AND VEGETABLES

*There are specific sensors for soil humidity and professional probes to measure in different soil profiles. A profile refers to a vertical depth. Imagine a stake driven into the ground that measures a value every 10 centimetres deep. We can differentiate between top-soil humidity, indicating a depth of about 20 centimetres, and bottom-soil humidity, indicating a depth of 50 centimetres. Make fun rules with your student.*

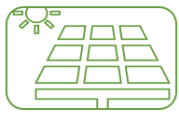




## DEFINE YOUR SEMANTIC ANNOTATION

Spaces, sensors, electro valves and other components must be identified in your project. All students must use the same annotation. IoT projects are collaborative and several students program their modules together by pushing data in the same database.

For example, spaces must be identified to control and manage them individually. We would like daily graphs for each sensor, making clear where the sensor is installed.



ID\_NODE\_01:  
Weather station



ID\_NODE\_02:  
Seed nursery



ID\_NODE\_03:  
Compost machin



ID\_NODE\_04:  
Bean plot



ID\_NODE\_05:  
Strawberry plot



ID\_NODE\_06:  
Aubergine plot

### spaces

Open the `send_params.h` file and identify the following items:

1. Garden name
2. ID\_Node: identifies the space (one Arduino - one space)
3. PLOTLIST: the list of garden spaces

Use the same type to identify a temperature sensor whether it is installed in a weather station or in a seed nursery. Thus, prioritise the order by both space + sensor to differentiate the measuring point.

### sensor types

In our Arduino example, sensors are identified as follows:

- TYPE 00: Brightness - 1 value
- TYPE 01: Temperature - 1 value
- TYPE 02: Humidity - 1 value
- TYPE 03: PH - 1 value
- TYPE 04: Relative noise - 1 value
- TYPE 05: Air Quality - 1 value
- TYPE 06: Rainfall - 1 value
- TYPE 07: Wind - 2 values
- TYPE 08: Soil temperature - 4 values
- TYPE 09: Soil humidity - 4 values



ID\_SENSOR\_00:  
"Brightness"



ID\_SENSOR\_01:  
"Temperature"



ID\_SENSOR\_02:  
"Humidity"



ID\_SENSOR\_03:  
"Rainfall"



ID\_SENSOR\_04:  
"Wind"



ID\_SENSOR\_05:  
"Soil humidity"



ID\_SENSOR\_06:  
"Soil temperature"



ID\_SENSOR\_07:  
"Electrovalve"



## DATA PATHS IN THE CLOUD

You can send the data from your sensors to different IoT platforms that allow you to visualise the data in real time. In this exercise we have set ourselves a major challenge: using a database as a cloud service.

If you continue down this path, you will have to learn how to perform actions such as adding security rules and managing users and permissions in your cloud. In our Arduino project, database paths are defined in Arduino code following a specific structure as follows:

### Data path

Open the `send_firebase.h` file. It contains the following call:

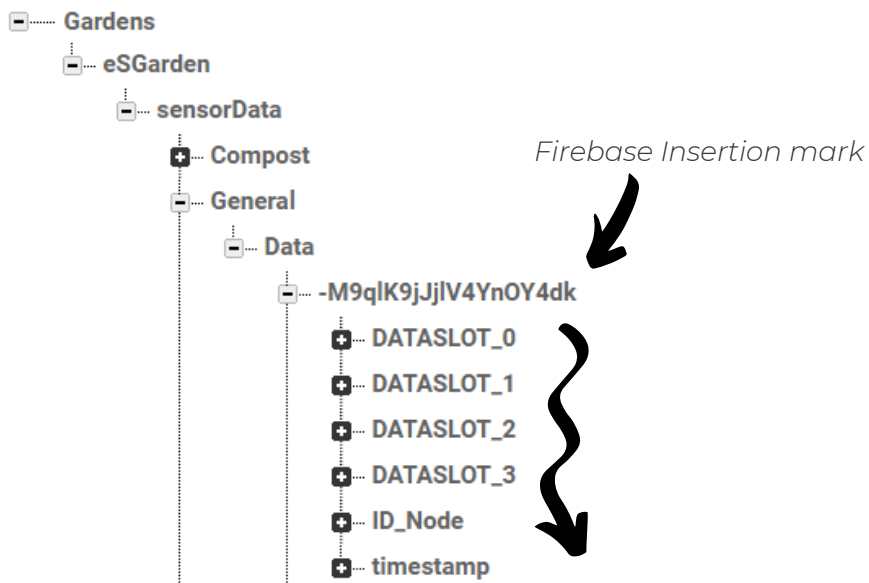
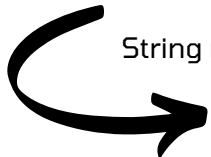
```
Firebase.pushJSON (firebaseData, Path, JsonSerialized)
```

```
int id_node = GetInteger (ID_NODE);
```

```
String route = "/Gardens/"+ GARDEN+ "/sensorData/"+ PLOTLIST [id_node]+ "/Data/";
```



*The path represents the structure of stored data in Firebase. This example push data from node 01: the weather station (general parameters)*



*Firebase Insertion mark*

*JSON wrapper created in MakeJSON function*

*Time mark inserted by firebase API; UNIX format*



## YOUR ACTIVITY:

## TAKE A LOOK AT YOUR SOLUTION

We made an example to explain the “organisation” concept. The app is available for Android on Google Play:



Download our example app. You can enter without logging in, just click on “Sign in” and leave the login fields blank. Your user level is that of a student, so you can navigate but the left upper corner menus are disabled.

This video will help you to understand our app.

When you open the app, you have a student’s view. Consider how our garden is organised and the sensors distributed according to the spaces.

1. What is your level of agreement on the use of apps and web applications in the educational framework?
2. Do you think that our app can motivate students to use technology in their education?
3. Do you think that our app can influence the environmental awareness of students and teachers?
4. Do you think our app is inclusive, making it accessible to students with learning difficulties?
5. Do you think our app is visually attractive and intuitive?

Please fill in the following questionnaire:





## ADAPT THE CODE TO YOUR CLOUD

You are ready to implement code with your own toolkits. Remember these ideas:

1. Organise your gardens, both the real and the virtual.
2. Use microcontroller-base toolkits with Wi-Fi chip or alternative solutions.
3. Propose collaborative projects; use common semantic annotations.
4. Take care of the data and how it is organised in your cloud.

### ***Next exercise ...***

In the next exercise we offer you a practical activity to implement a weather station and a soil moisture probe for your garden.



Project name: eSGarden – School Gardens for Future Citizens  
Number: 2018-ES01-KA201-050599



*This leaflet reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*