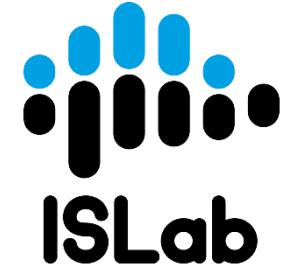


University of Minho
School of Engineering



Ambient Intelligence

Autonomous Systems

Perfil Sistemas Inteligentes @ MEI/MiEI 1º/4º - 2º semestre

Bruno Fernandes, Cesar Analide, Fábio Silva

11/03/2019

Contents

2

Concepts

Sensorization

Hands On

- Concepts
 - Ambient Intelligence
 - Internet of Things
 - Internet of People
 - Smart Cities
- Sensors
 - Beacons
 - ESP8266, ESP32 and Arduino Mega
 - Phidgets
 - Virtual Sensors
- Hands On

Ambient Intelligence (Aml)

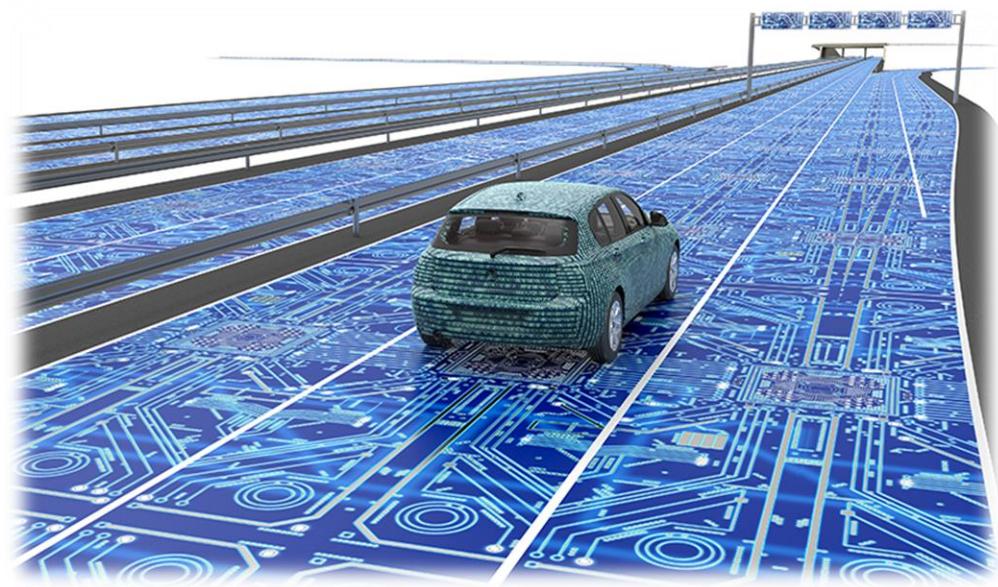
3

CONCEPTS

Sensorization

Hands On

Ambient Intelligence refers to environments that are **sensitive** and **responsive** to the **presence of people** in a non-intrusive manner! As devices grow smaller, connected and more integrated with the environment, the technology disappears into our surroundings!



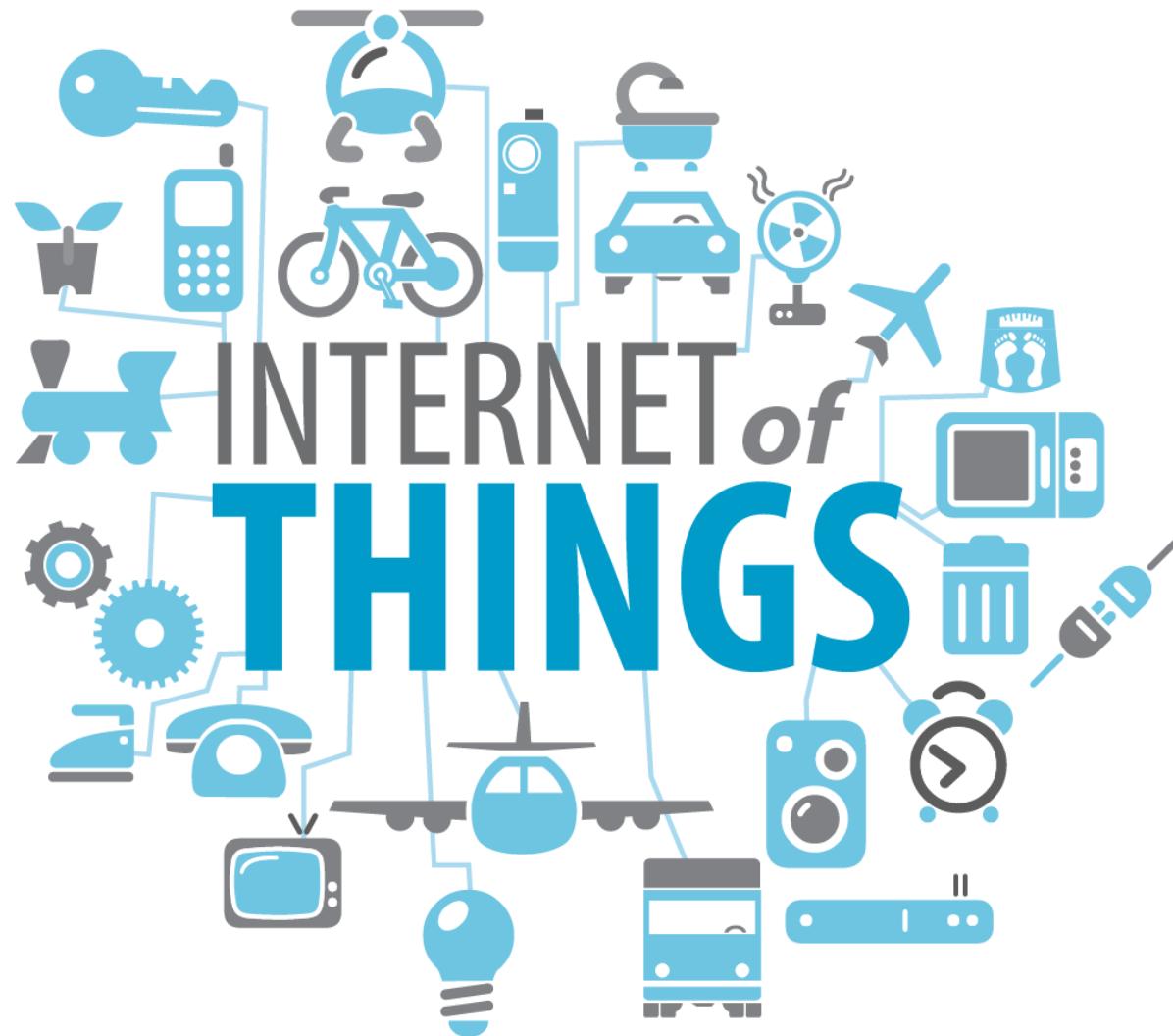
Internet of Things (IoT)

4

CONCEPTS

Sensorization

Hands On



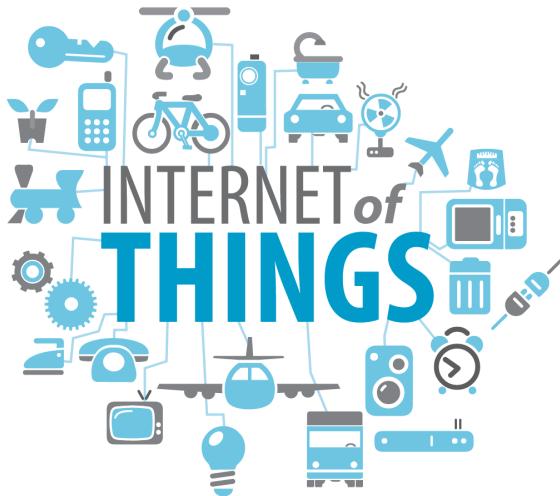
Internet of Things (IoT)

5

CONCEPTS

Sensorization

Hands On



An open and comprehensive **network of intelligent objects** that have the capacity to auto-organize, share information, data and resources, **reacting** and **acting** in face of situations and changes in the environment.

Internet of People (IoP)

6

CONCEPTS

Sensorization

Hands On



Internet of People (IoP)

7

CONCEPTS

Sensorization

Hands On



A dynamic global **network** where **things** and **people** communicate and understand each other; where everyone and everything can sense the other and the world, and act on such knowledge and information, aiming to enhance **people's quality of life**.

Smart Cities

8

CONCEPTS

Sensorization

Hands On



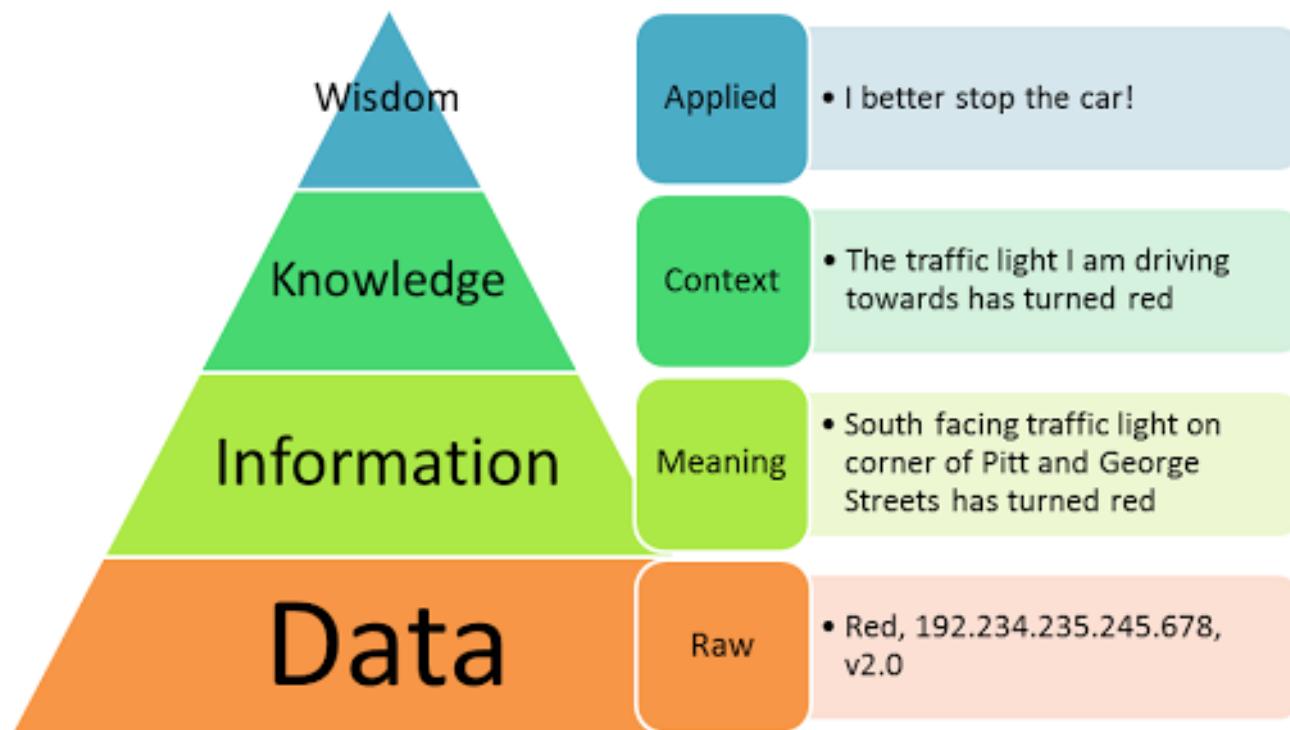
Data Pyramid (aka DIKW pyramid)

9

CONCEPTS

Sensorization

Hands On



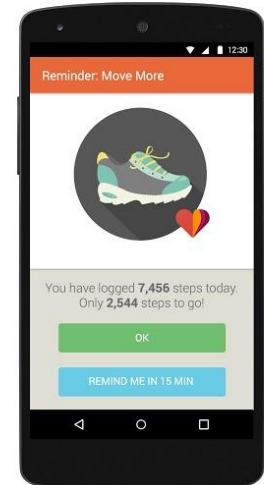
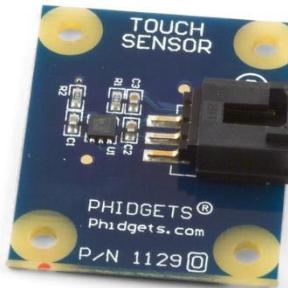
Sensorization

10

Concepts

SENSORIZATION

Hands On



Sensorization

11

Concepts

SENSORIZATION

Hands On

A **sensor** is something that is able to **percept phenomena that is being observed** and **translate its state**

Traditionally sensors were **physical** and observed physical phenomena but sensors may also be **virtual**:

- Access to web API
- Mathematical formulae

Currently, data fusion can also infer virtual assets such as:

- Emotions
- Well Being
- Sustainability
- Happiness

Ambient Intelligence

12

Concepts

SENSORIZATION

Hands On

1. Data Acquisition

- Sensors
- Services
- Data processing

2. Reasoning

- Data Modeling
- Machine Learning
- Decision Models

3. Actuation

- Notifications
- Interactions
- Actions

Beacons

13

Concepts

SENSORIZATION

Hands On

A **bluetooth-based sensor** with low-cost, low-power transmitters (a Bluetooth Smart/LE signal), which **notify bluetooth devices** of one's presence.

This signal makes it **possible to identify the beacon** as well as other telemetry information about the receiving device. It has **no user interface or GPS** capabilities.

The beacon works as such:

- periodically wakes up
- transmits a BLE signal
- returns to a low-power state



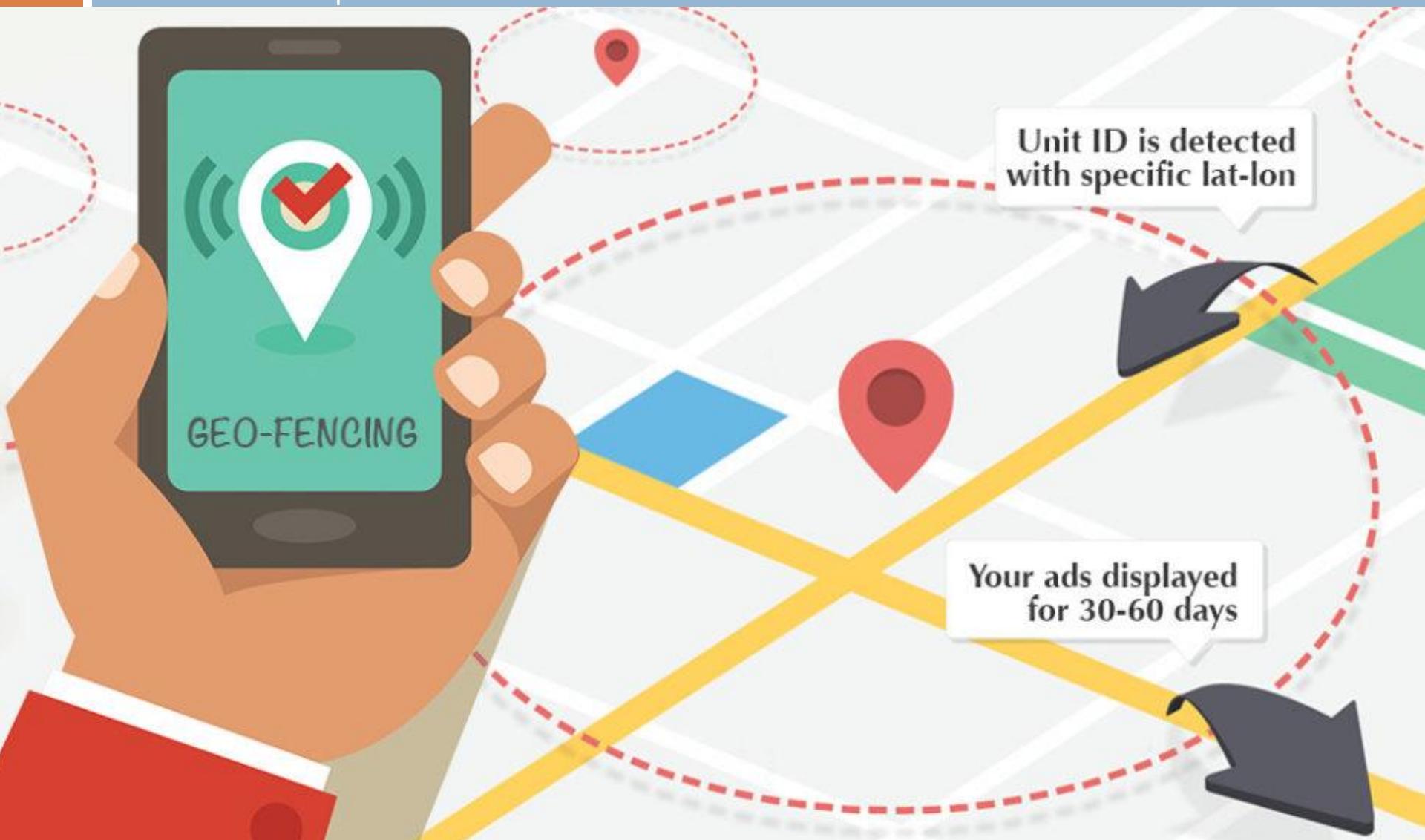
Beacons - Case Studies

14

Concepts

SENSORIZATION

Hands On



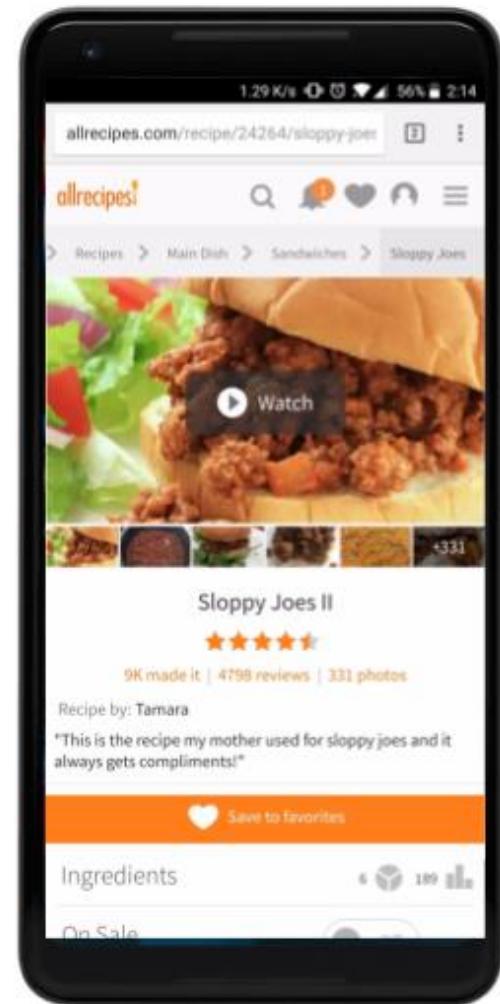
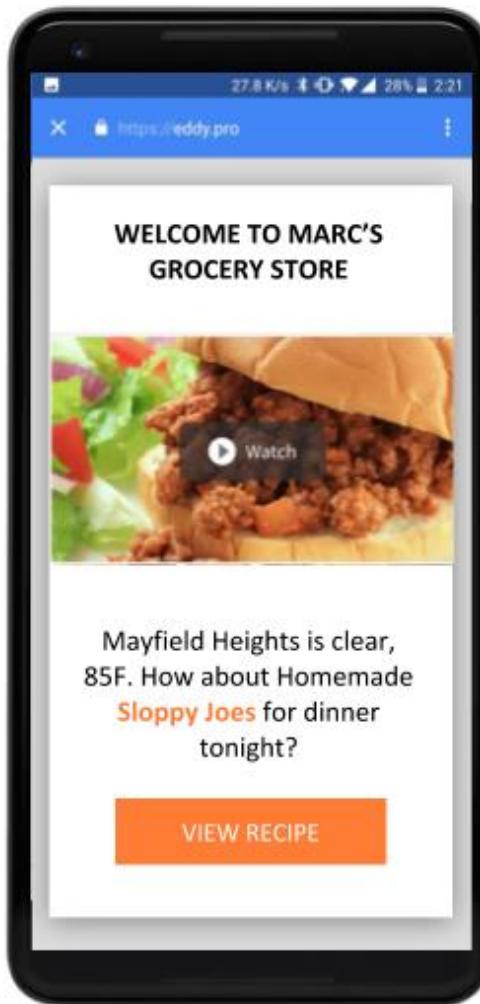
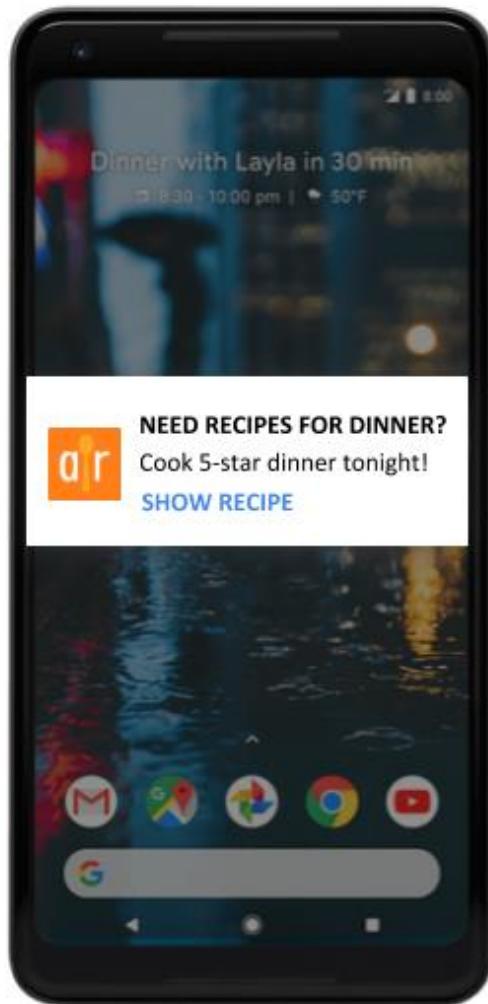
Beacons - Case Studies

15

Concepts

SENSORIZATION

Hands On



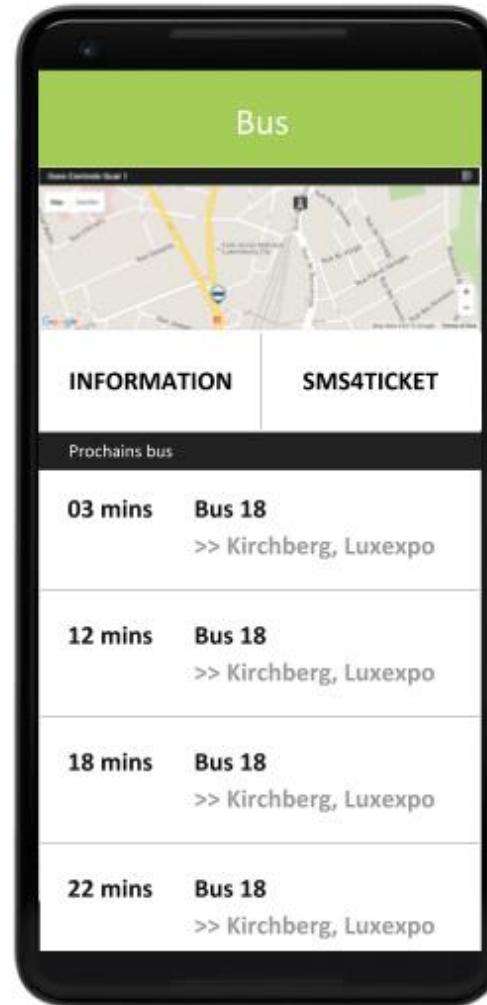
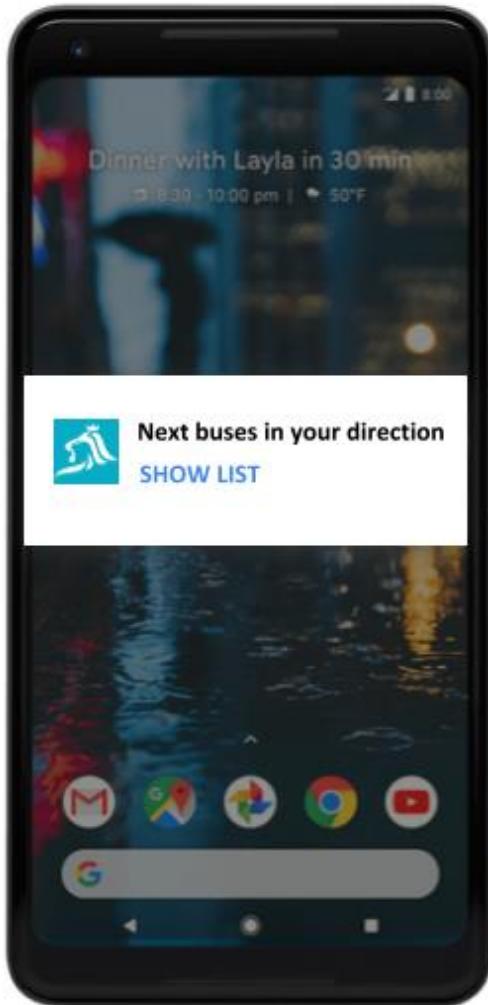
Beacons - Case Studies

16

Concepts

SENSORIZATION

Hands On



Beacons - How To

17

Concepts

SENSORIZATION

Hands On

The screenshot shows the GIMBAL web application interface. The left sidebar has a dark theme with white text and icons. It includes links for Home, Applications, Places, Beacons (selected), Beacon Management, Beacon Configurations, Beacon Sharing, and Server Rules. The main content area has a light background. At the top, there are four tabs: My Beacons (highlighted in red), Public Beacons, Registered Sharing Keys, and Third-Party Beacons. Below the tabs is a toolbar with buttons for Activate Beacon, Export Beacons, and Import CSV. A search bar is followed by a dropdown menu set to 50. A table lists three beacons: Pedestrian A, Pedestrian B, and Runner B. Each row includes columns for Icon, Name, Hardware, Factory ID, Firmware, Battery Level, Visibility, Type, Assigned Configuration, Applied Configuration, Tags, and Actions (with edit and delete icons). A note above the table says, "These are all the beacons you have activated, to buy more, visit our [Store](#)".

ICON	NAME	HARDWARE	FACTORY ID	FIRMWARE	BATTERY LEVEL	VISIBILITY	TYPE	ASSIGNED CONFIGURATION	APPLIED CONFIGURATION	TAGS	ACTIONS
	Pedestrian A			Default	Med-High	Private	Gimbal	Series 10 Recommended			
	Pedestrian B			Default	Med-High	Private	Gimbal	Series 10 Recommended			
	Runner B			Default	Med-High	Private	iBeacon	iBeacon			

Beacons - How To

18

Concepts

SENSORIZATION

Hands On

Add **Maven Central** and **Google** to the repositories section of your project's **build.gradle** file:

```
repositories {  
    ...  
    mavenCentral()  
    google()  
    ...  
}
```

Add the following artifacts to the **dependencies section** of your application module's **build.gradle** file:

```
dependencies {  
    ...  
    implementation 'com.gimbal.android.v4:gimbal-sdk:4.0.1'  
    implementation 'com.gimbal.android.v4:gimbal-slf4j-impl:4.0.1'  
    implementation 'com.google.android.gms:play-services-ads-identifier:16.0.0'  
    implementation 'com.google.android.gms:play-services-location:16.0.0'  
    implementation 'com.google.firebaseio:firebase-messaging:17.4.0'  
    ...  
}
```



Beacons - How To

19

Concepts

SENSORIZATION

Hands On

Three main things to do:

1. Set the API key

```
//must be invoked before any other Gimbal calls, from onCreate()

import com.gimbal.android.Gimbal;
...
public class MyApplication extends Application {
    private static final String GIMBAL_APP_API_KEY = "YOUR APP'S API KEY HERE";
    ...
    public void onCreate() {
        ...
        Gimbal.setApiKey(this, GIMBAL_APP_API_KEY);
        ...
    }
}
```



Beacons - How To

20

Concepts

SENSORIZATION

Hands On

Three main things to do:

2. Register Gimbal event listeners

```
//The Gimbal SDK has 3 listener interfaces for Communications, Places and Beacons
import com.gimbal.android.Gimbal;
import com.gimbal.android.Communication;
import com.gimbal.android.CommunicationListener;
import com.gimbal.android.CommunicationManager;
...
public class MyApplication extends Application {
    private CommunicationListener communicationListener;
    ...
    public void onCreate() {
        Gimbal.setApiKey(this, GIMBAL_APP_API_KEY);
        communicationListener = new CommunicationListener() {
            //override methods as required
        };
        CommunicationManager.getInstance().addListener(communicationListener);
    }
}
```



Beacons - How To

21

Concepts

SENSORIZATION

Hands On

Three main things to do:

3. Start Monitoring for Gimbal Events

```
import com.gimbal.android.Gimbal;  
...  
public class PermissionActivity extends AppCompatActivity {  
    public static final int LOCATION_PERMISSION_REQUEST_CODE = 101;  
    ...  
    @Override  
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {  
  
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {  
                Gimbal.start();  
            }  
        }  
    }  
}
```



Beacons - How To

22

Concepts

SENSORIZATION

Hands On

Some key docs:

- https://docs.gimbal.com/proximity_overview.html
- <http://docs.gimbal.com/android/v4/devguide.html>
- <https://github.com/gimbalinc/hello-gimbal-android>



ESP8266

23

Concepts

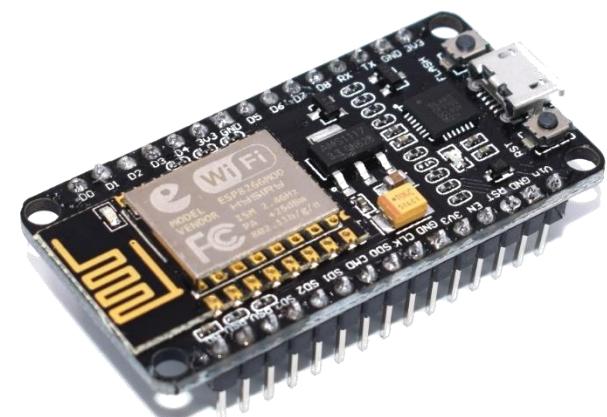
SENSORIZATION

Hands On

A low-power **Arduino type board**, suitable for the IoT, that can facilitate the **bridge towards Smart Cities**, removing the need for wired communication and processing.

A very interesting set of features...

- Wi-Fi capability (2.4 GHz band)
- 4 MB of flash memory
- a micro-USB interface
- a built-in antenna
- open-source
- small dimensions (4.8x2.4x0.5cm)
- low weight (109g)
- Ultra-Low Power Consumption



ESP32

24

Concepts

SENSORIZATION

Hands On

A low-power **Arduino type board**, suitable for the IoT, that can facilitate the **bridge towards Smart Cities**, removing the need for wired communication and processing.

A very interesting set of features...

- The same as the previous slide!
- **BLE connectivity** (Hybrid Wi-Fi & Bluetooth Chip)!
- **Dual-core!**

However, being this a new player the availability (and **documentation**) of libraries for the ESP32 is **significantly lower** when compared to the ESP8266.



Arduino Mega

25

Concepts

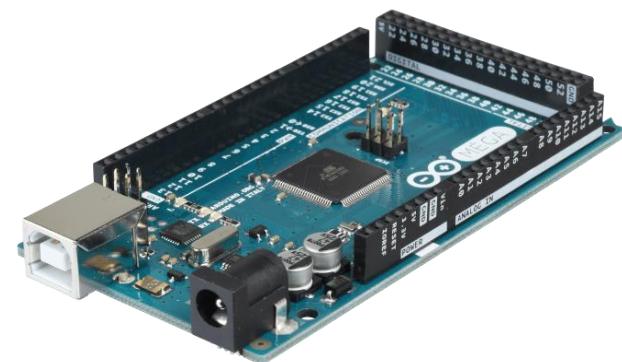
SENSORIZATION

Hands On

A low-power **Arduino type board**, suitable for the IoT, that can facilitate the **bridge towards Smart Cities**, removing the need for wired communication and processing.

Also a very interesting set of features...

- More stable than ESPs...
- However it **does not have neither WiFi or BLE** capabilities!



Arduino(-type) Boards - Case Studies

26

Concepts

SENSORIZATION

Hands On



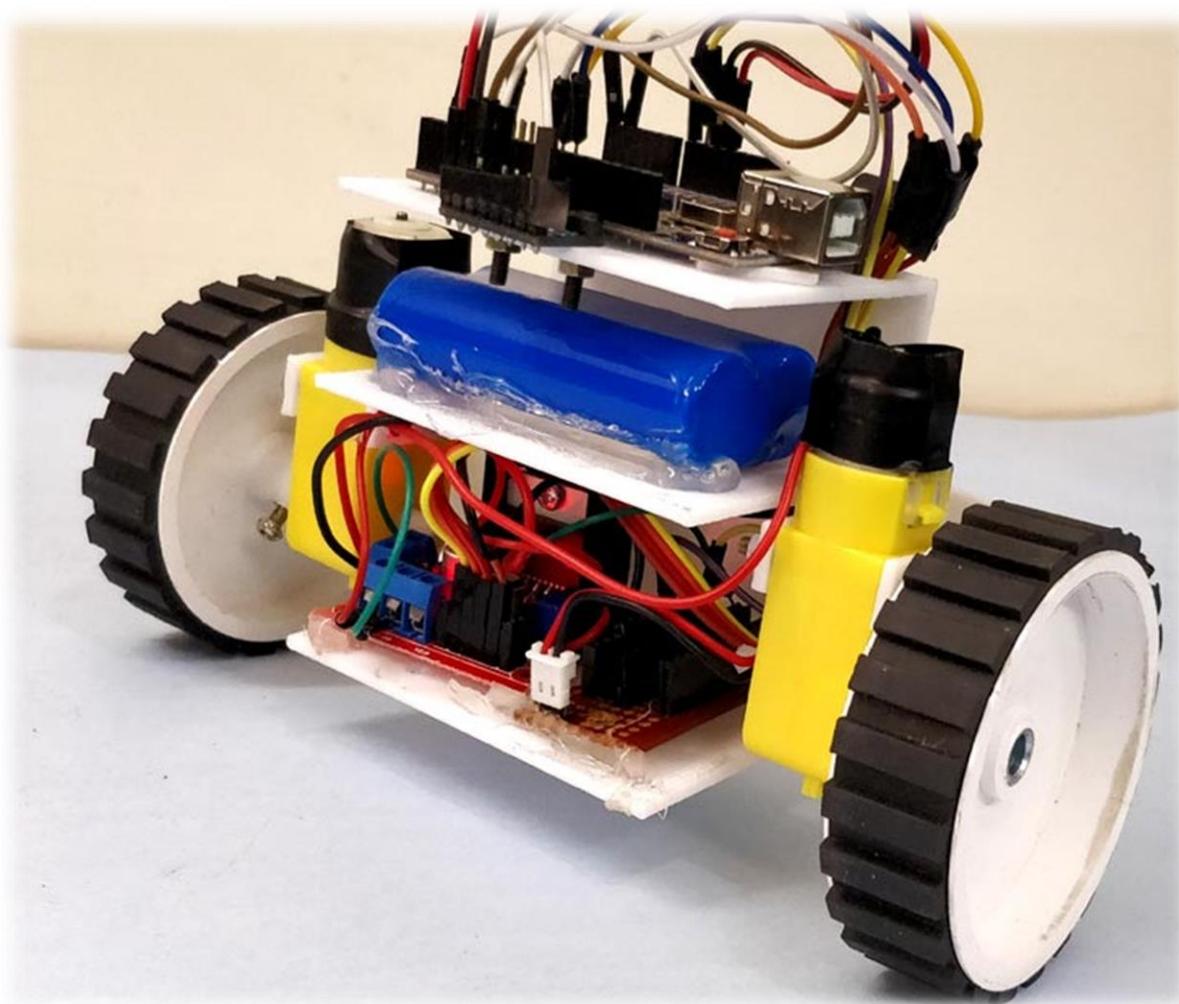
Arduino(-type) Boards - Case Studies

27

Concepts

SENSORIZATION

Hands On



Arduino(-type) Boards - Case Studies

28

Concepts

SENSORIZATION

Hands On



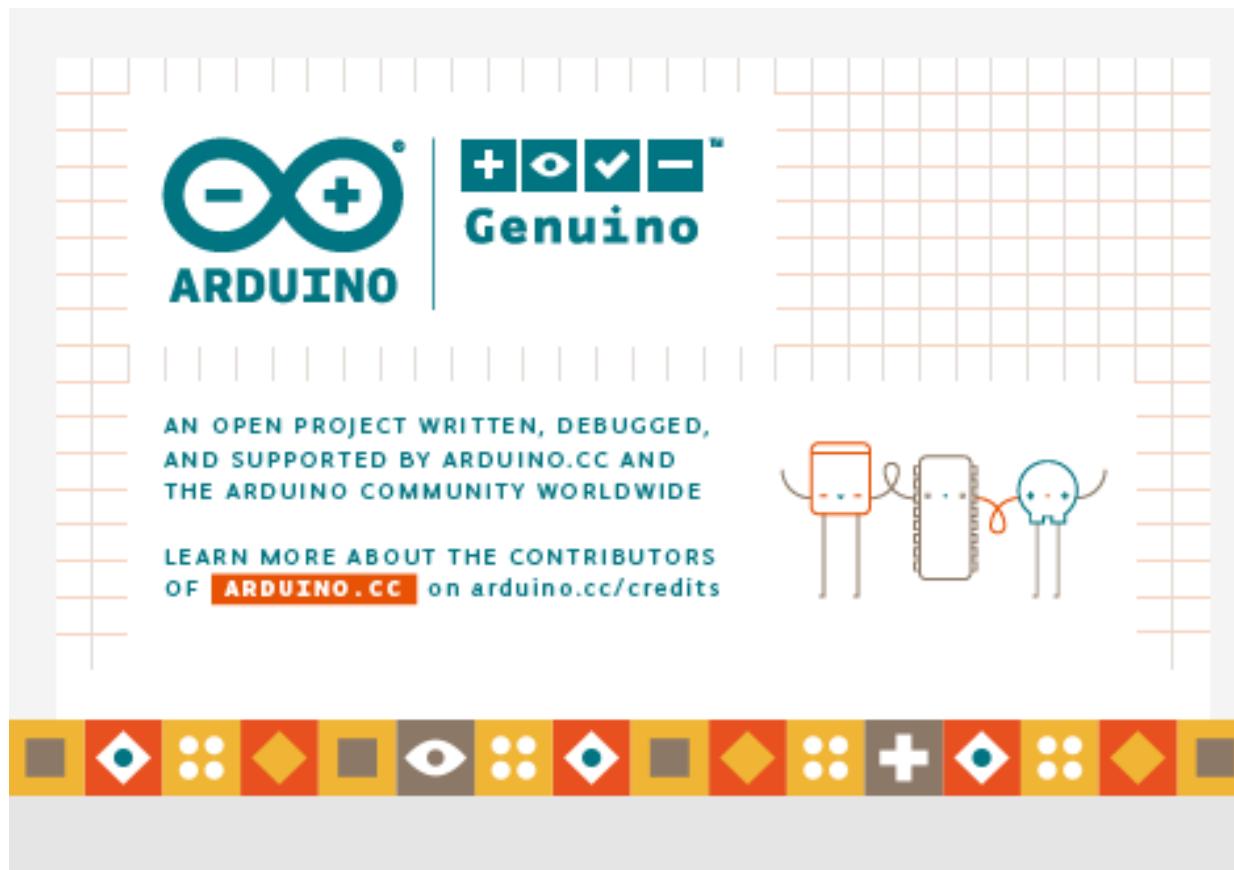
Arduino IDE - How To

29

Concepts

SENSORIZATION

Hands On



Arduino IDE - How To

30

Concepts

SENSORIZATION

Hands On

- Makes it very easy to develop code and upload it to a board!
- Provides basic **one-click mechanisms** to **compile** and **upload sketches**
 - **Sketch** is the name given to a program developed with this IDE (written in C/C++)
- The nature of the Arduino project facilitated the release of many open-source libraries
- Programming in the Arduino IDE requires the developer to define, at least, two functions:
 - **setup()** - called once when a sketch starts after powering up or resetting, being used to initialize variables, input and output pin modes, and other libraries required by the sketch
 - **loop()** - repeatedly executed in the main program until the board is powered off or reset

Arduino IDE - How To

31

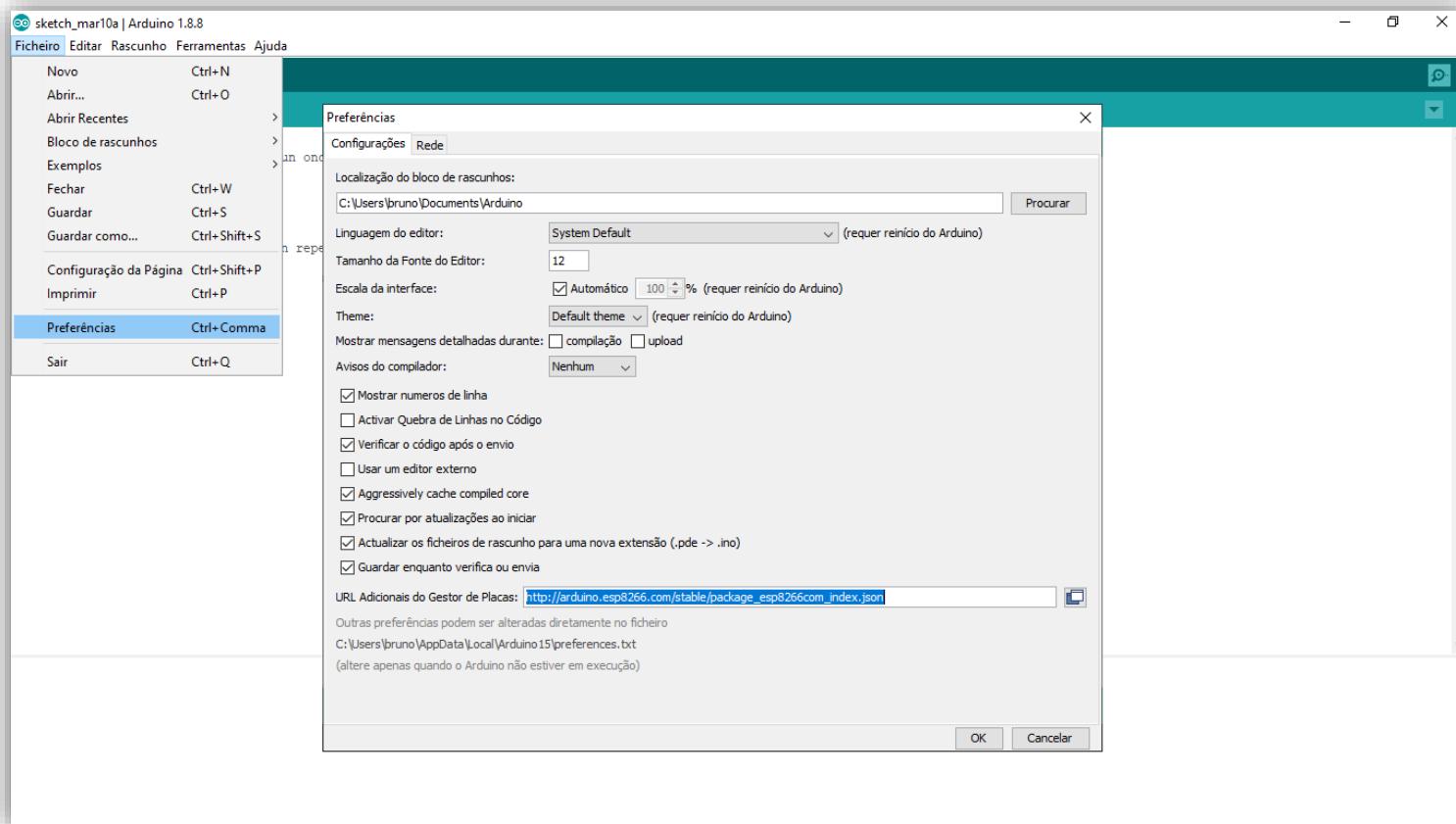
Concepts

SENSORIZATION

Hands On

1. Install the board (ESP8266) in Arduino IDE

- http://arduino.esp8266.com/stable/package_esp8266com_index.json



Arduino IDE - How To

32

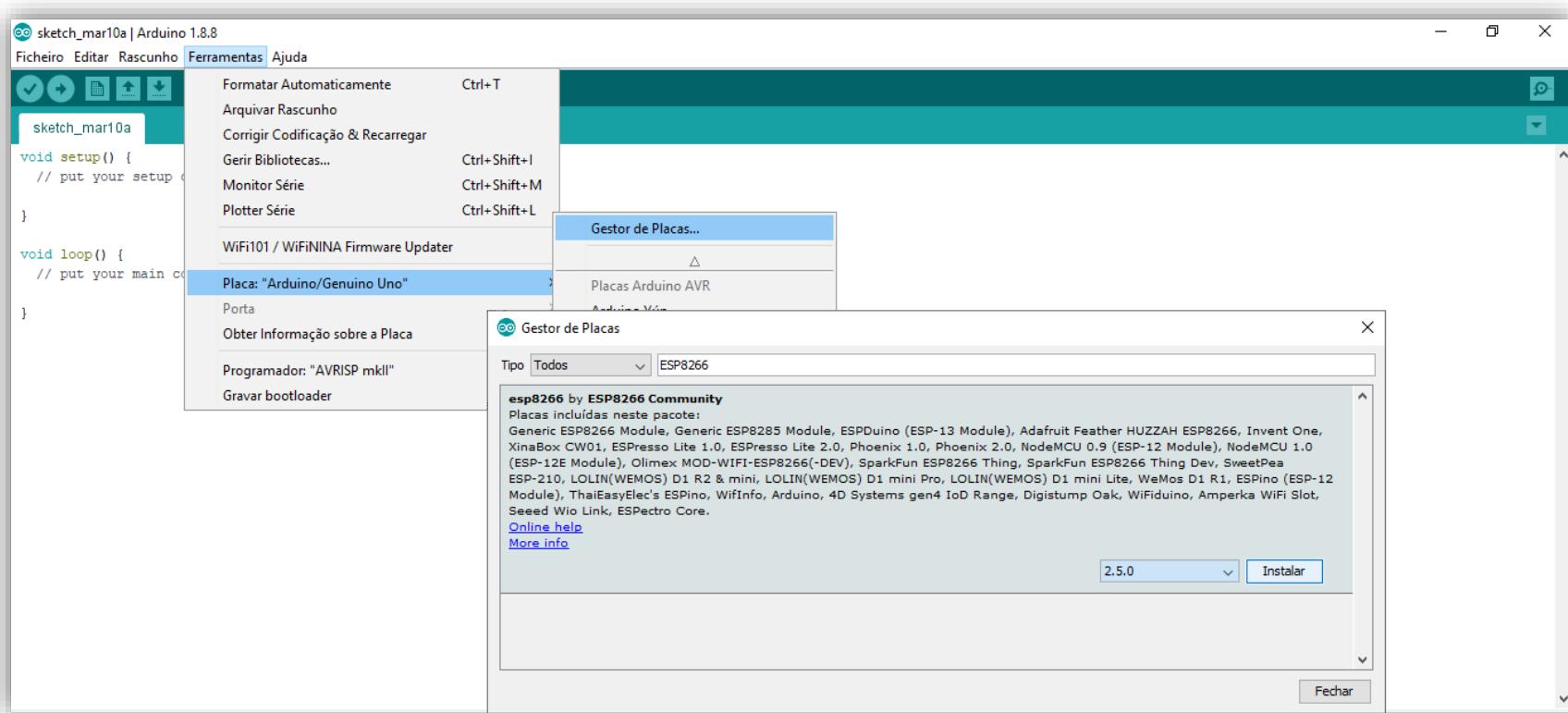
Concepts

SENSORIZATION

Hands On

1. Install the board (ESP8266) in Arduino IDE

- http://arduino.esp8266.com/stable/package_esp8266com_index.json



Arduino IDE - How To

33

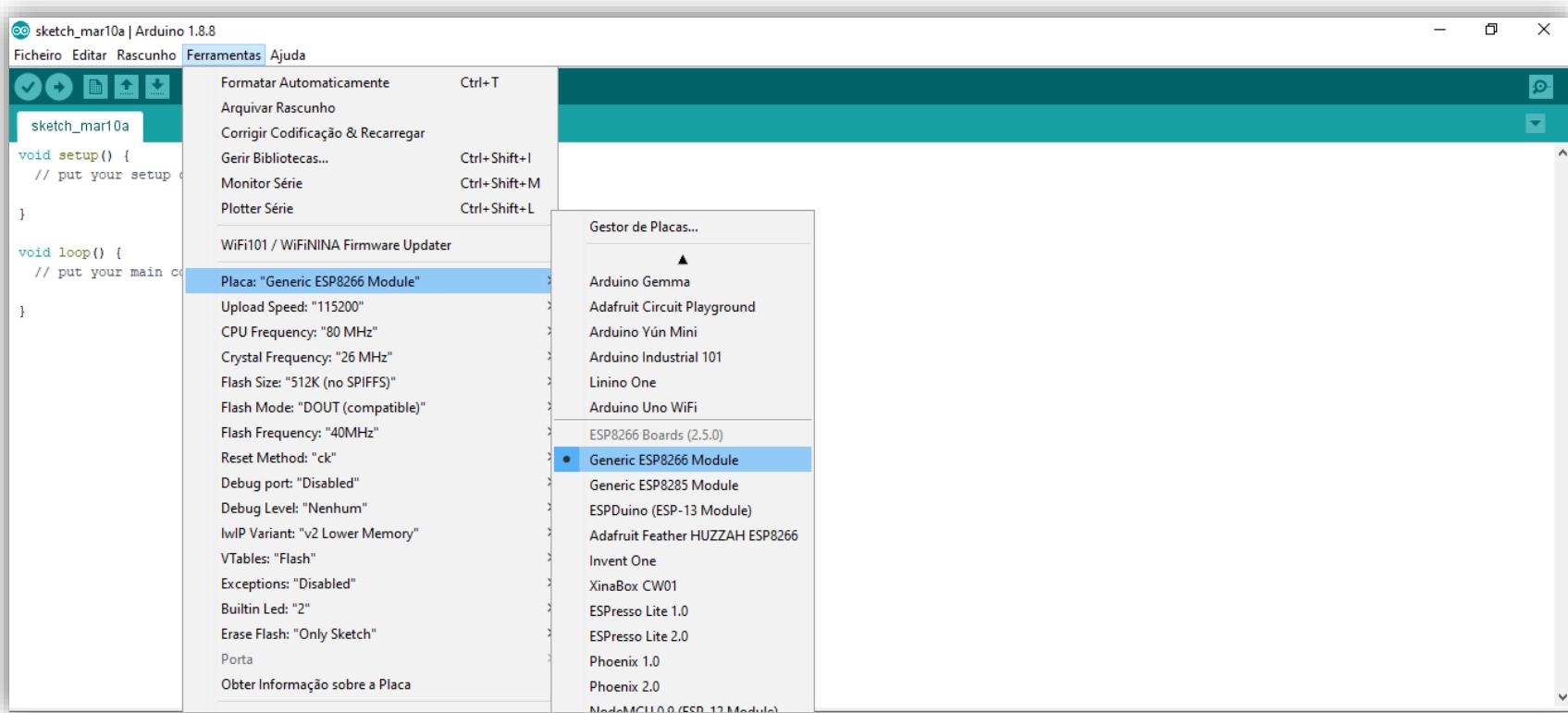
Concepts

SENSORIZATION

Hands On

1. Install the board (ESP8266) in Arduino IDE

- http://arduino.esp8266.com/stable/package_esp8266com_index.json



Arduino IDE - How To

34

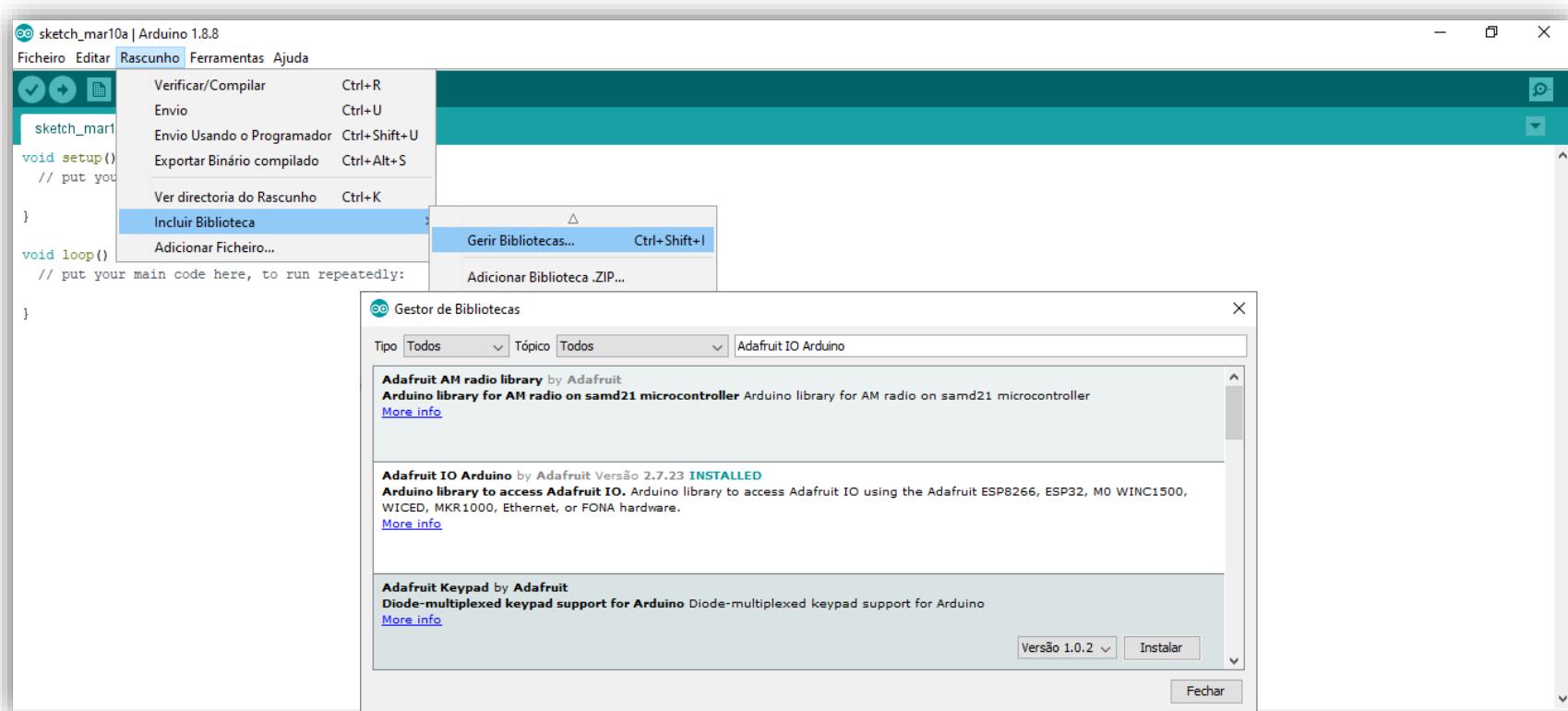
Concepts

SENSORIZATION

Hands On

2. Install some libraries

- ArduinoHttpClient, Adafruit IO Arduino, Adafruit MQTT, PubSubClient, ArduinoJson and FirebaseArduino (<https://github.com/googlesamples/firebase-arduino/archive/master.zip>)



Arduino IDE - How To

35

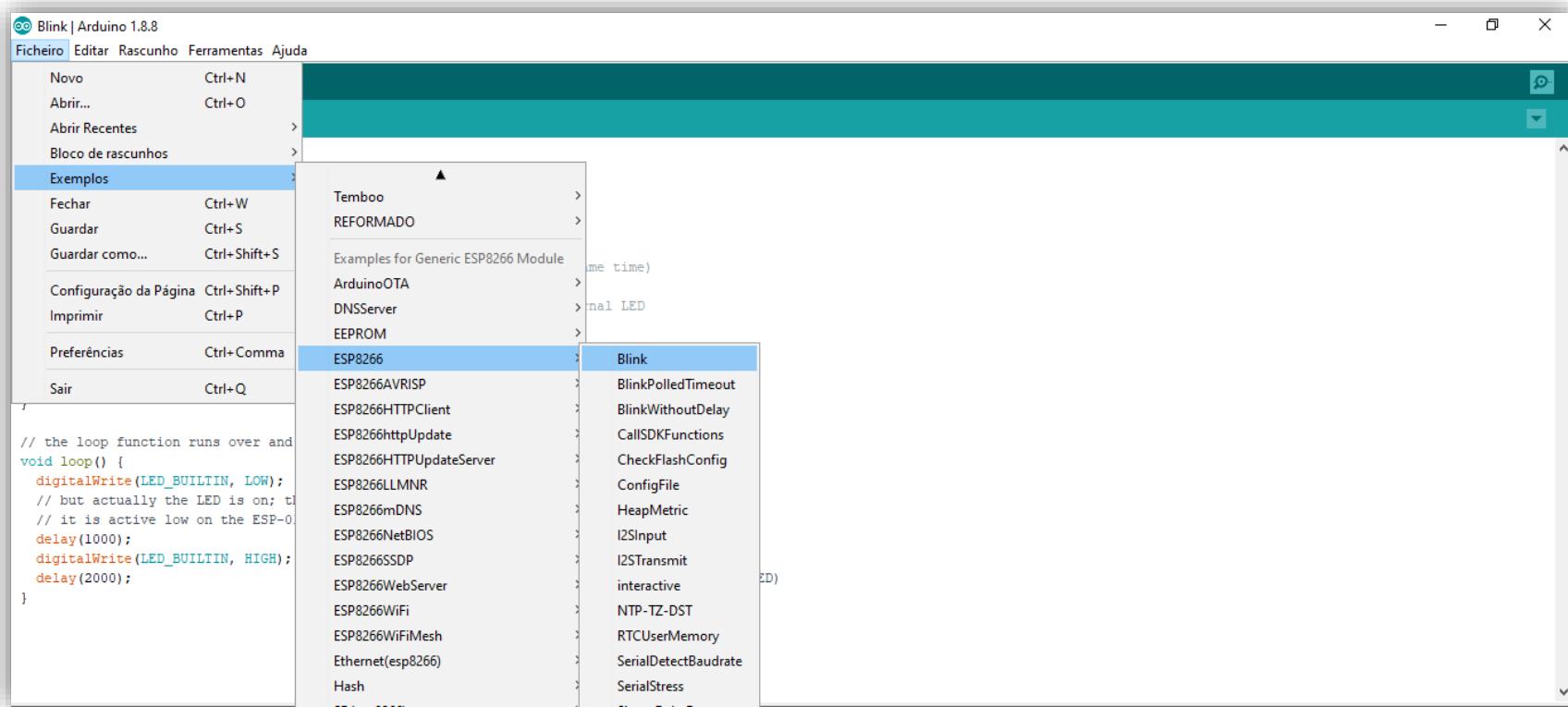
Concepts

SENSORIZATION

Hands On

3. Try an example sketch!

- Blink the Led!



Arduino IDE - How To

36

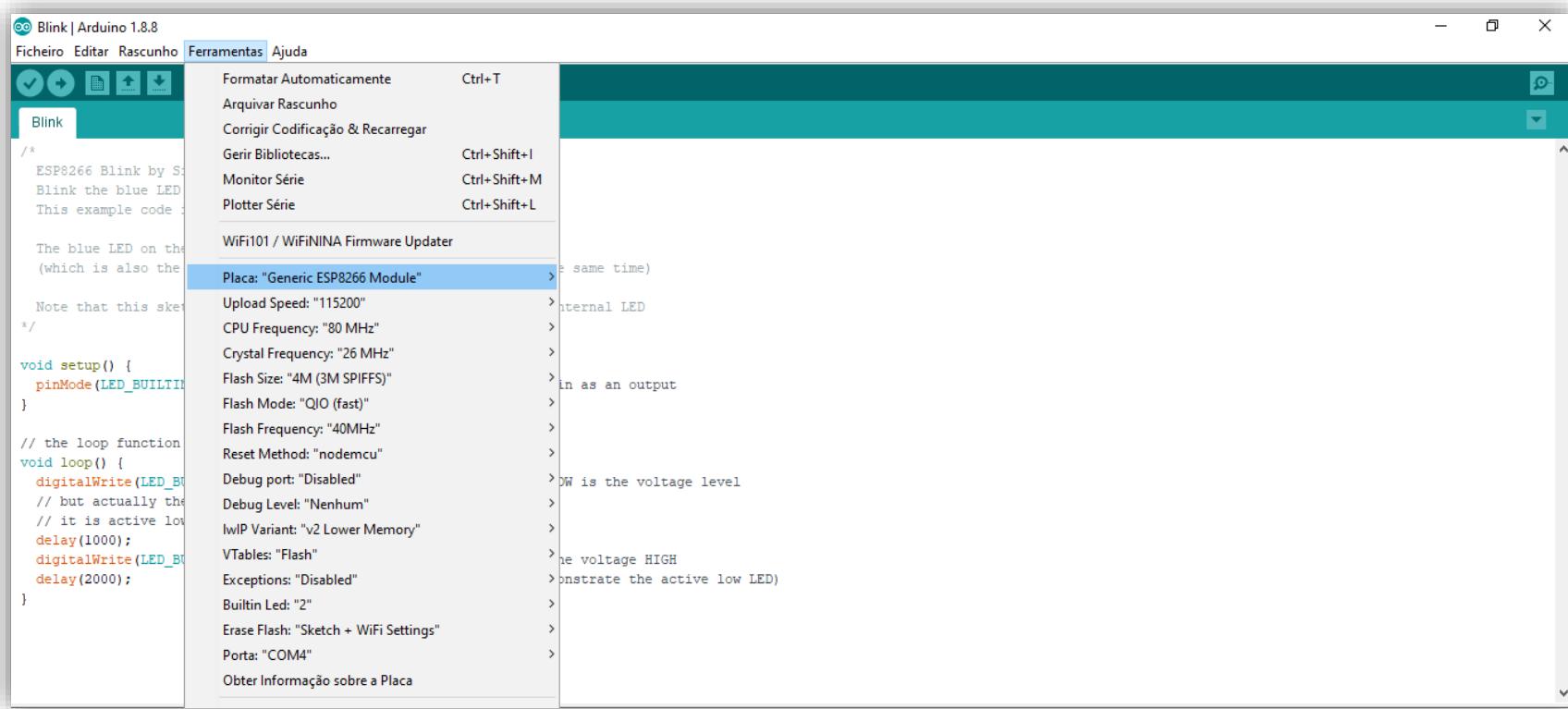
Concepts

SENSORIZATION

Hands On

3. Try an example sketch!

- Connect the board to the PC and set the correct board parameters



Arduino IDE - How To

37

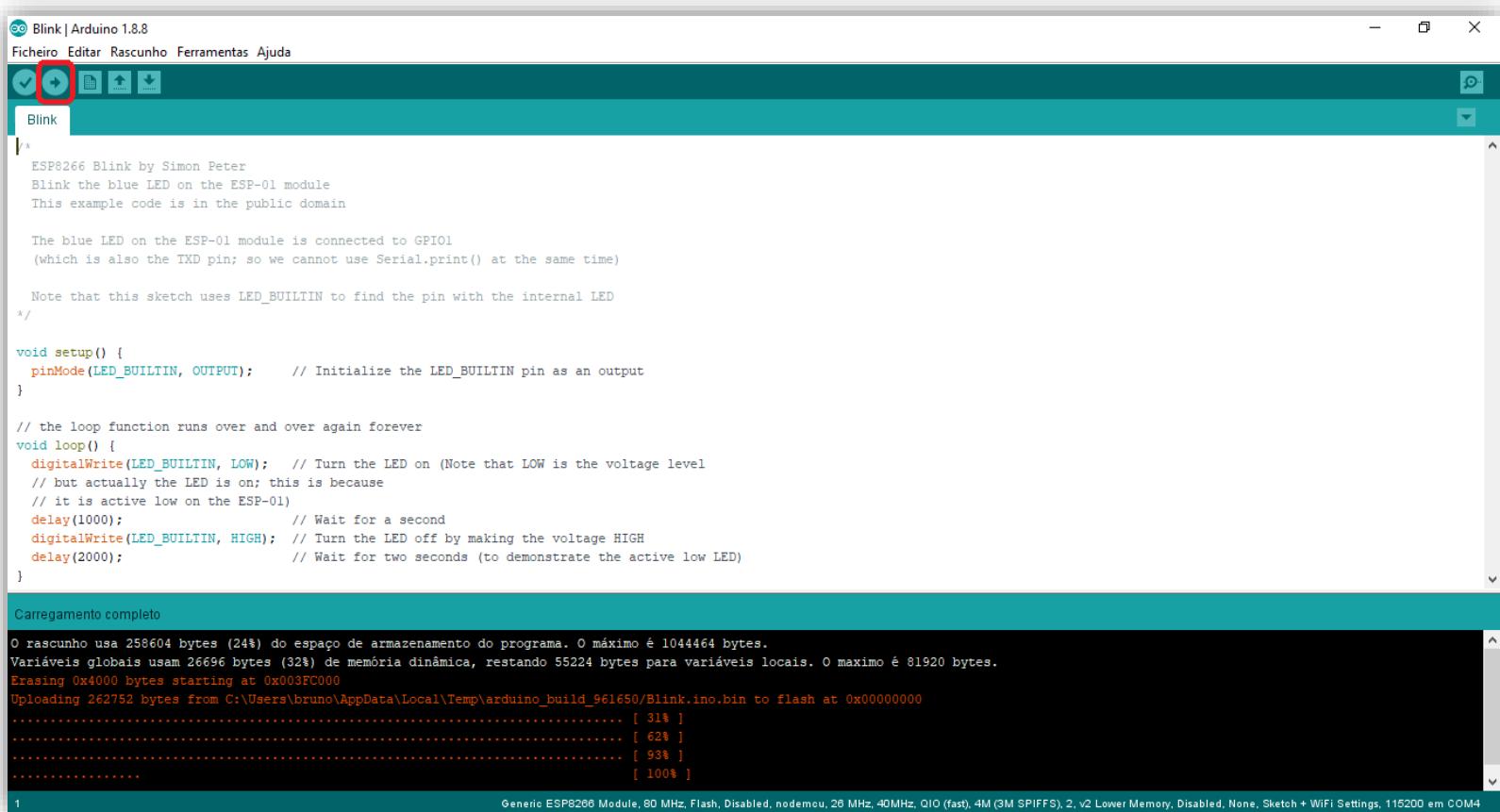
Concepts

SENSORIZATION

Hands On

3. Try an example sketch!

- Compile and Upload it!



EE Blink | Arduino 1.8.8

Ficheiro Editar Rascunho Ferramentas Ajuda

Blink

ESP8266 Blink by Simon Peter
Blink the blue LED on the ESP-01 module
This example code is in the public domain

The blue LED on the ESP-01 module is connected to GPIO1
(which is also the TXD pin; so we cannot use Serial.print() at the same time)

Note that this sketch uses LED_BUILTIN to find the pin with the internal LED

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);      // Initialize the LED_BUILTIN pin as an output  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW);    // Turn the LED on (Note that LOW is the voltage level  
    // but actually the LED is on; this is because  
    // it is active low on the ESP-01)  
    delay(1000);                      // Wait for a second  
    digitalWrite(LED_BUILTIN, HIGH);   // Turn the LED off by making the voltage HIGH  
    delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)  
}
```

Carregamento completo

O rascunho usa 258604 bytes (24%) do espaço de armazenamento do programa. O máximo é 1044464 bytes.
Variáveis globais usam 26696 bytes (32%) de memória dinâmica, restando 55224 bytes para variáveis locais. O maximo é 81920 bytes.
Erasing 0x4000 bytes starting at 0x003FC000
Uploading 262752 bytes from C:\Users\bruno\AppData\Local\Temp\arduino_build_961650\Blink.ino.bin to flash at 0x00000000
..... [31%]
..... [62%]
..... [93%]
..... [100%]

1 Generic ESP8266 Module, 80 MHz, Flash, Disabled, nodemcu, 26 MHz, 40MHz, QIO (fast), 4M (3M SPIFFS), 2, v2 Lower Memory, Disabled, None, Sketch + WiFi Settings, 115200 em COM4

Arduino IDE - How To

38

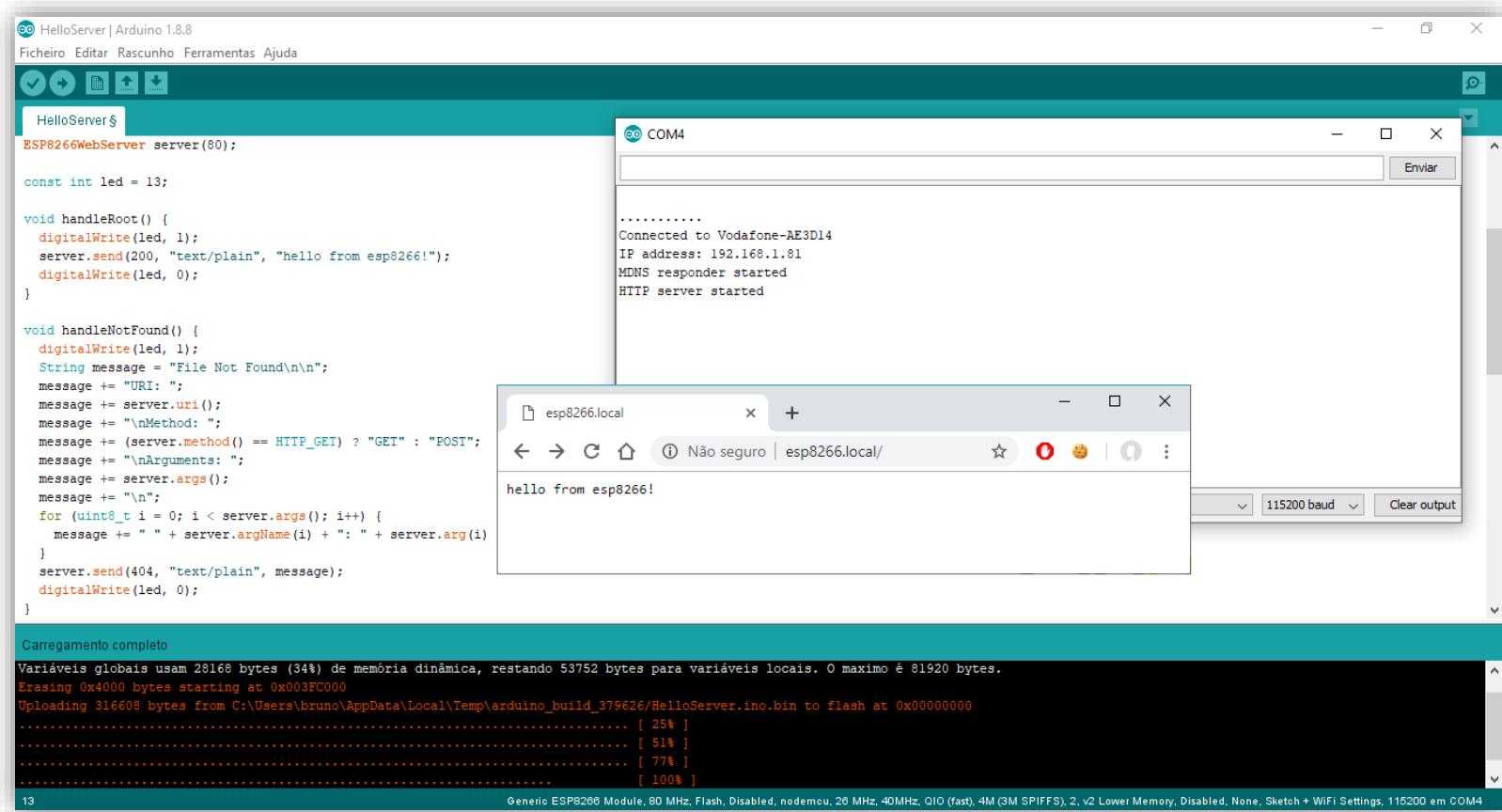
Concepts

SENSORIZATION

Hands On

3. Try an example sketch!

- Another example HelloServer!



Arduino IDE - How To

39

Concepts

SENSORIZATION

Hands On

3. Try the Crowd Sensing sketch (compatible with ArduinoJson v5)!

- <https://goo.gl/RoPJM7>

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** CrowdSensing | Arduino 1.8.8
- File Menu:** Ficheiro, Editar, Rascunho, Ferramentas, Ajuda
- Sketch Area:** The code for the CrowdSensing sketch is displayed. It includes setup and loop sections with various WiFi and probe request handling functions.
- Serial Monitor:** A window titled "COM4" shows the output of the ESP8266 probe requests. The output includes messages like "Connected to Vodafone-AE3D14; IP address: 192.168.1.81" and a list of probe requests with their details (RSSI, Millis Last Detected).
- Bottom Status:** Shows the upload progress: "Carregamento completo" (Upload complete), memory usage: "Variáveis globais usam 32120 bytes (3%) de memória dinâmica, reservando 49600 bytes para variáveis locais. O MÁXIMO é 61920 bytes.", and the upload command: "Erasing 0x4000 bytes starting at 0x003FC000".
- Page Footer:** Includes the page number "47" and the footer text "Generic ESP8266 Module, 80 MHz, Flash, Disabled, nodemcu, 26 MHz, 40MHz, QIO (fast), 4M (3M SPIFFS), 2, v2 Lower Memory, Disabled, None, Sketch + WiFi Settings, 115200 em COM4".

Phidgets

40

Concepts

SENSORIZATION

Hands On

Phidgets are building-blocks for **sensing and control** using a computer, tablet or a phone. With Phidgets, we just need **focus on programming** and the details of the final product.

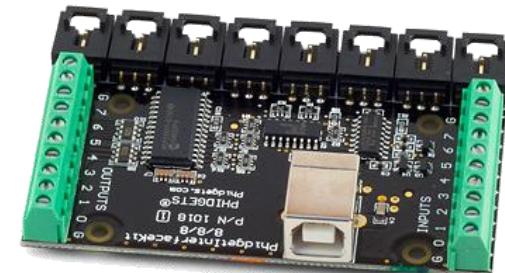
Install the drivers

- https://www.phidgets.com/docs/Phidgets_Drivers
- https://www.phidgets.com/docs/Language_-_Java



Phidget Programming Basics:

- https://www.phidgets.com/docs/Phidget_Programming_Basics



Virtual Sensors

41

Concepts

SENSORIZATION

Hands On



<https://openweathermap.org/api>



<https://docs.openaq.org/>



FOR DEVELOPERS

<https://developer.tomtom.com/>



<https://www.openuv.io/uvindex>



<https://pro.whitepages.com/apis/>



<https://developers.google.com/>



<https://developers.coinbase.com/>

Virtual Sensors

42

Concepts

SENSORIZATION

Hands On

- Essentially, you need to make **API calls** through **HTTP Get Requests**
- Parse the received JSON (you could go for XML if you really want to...)
- You may, or may not, need an **API key**
- You will definitely have a limited amount of requests to make

Open Weather Map

43

Concepts

SENSORIZATION

Hands On

The screenshot shows the OpenWeatherMap website's main navigation bar and the 'Weather API' page. The top navigation bar includes links for Support Center, Sign In, Sign Up, and temperature units (°C, °F). Below the bar, there are links for Weather, Maps, Guide, API, Price, Partners, Stations, Widgets, and Blog. The 'Weather API' page features a main message encouraging users to sign up for their APIs. It then details three types of forecasts: Current weather data, 5 day / 3 hour forecast, and 16 day / daily forecast, each with its own API documentation and subscription links.

Support Center Q Weather in your city Sign In Sign Up °C °F

OpenWeatherMap Weather Maps ▾ Guide API Price Partners Stations Widgets Blog

Weather API

Home / Weather API

Please [sign up](#) and use our fast and easy-to-work weather APIs for free. Look at our [monthly subscriptions](#) for more options than Free account can provide you. Read [How to start](#) first and enjoy using our powerful weather APIs.

Current weather data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid accounts

5 day / 3 hour forecast

[API doc](#) [Subscribe](#)

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

16 day / daily forecast

[API doc](#) [Subscribe](#)

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for all paid accounts

Open Weather Map

44

Concepts

SENSORIZATION

Hands On

The screenshot shows the OpenWeatherMap website with the 'Weather API' section selected. The browser window displays a JSON response from the API endpoint `api.openweathermap.org/data/2.5/weather`.

```
{"coord":{"lon":-8.43,"lat":41.54}, "weather":[{"id":802,"main":"Clouds","description":"nuvens dispersas","icon":"03n"}], "base":"stations", "main": {"temp":14.56,"pressure":1032,"humidity":93,"temp_min":13.33,"temp_max":16.11}, "visibility":10000, "wind": {"speed":4.6,"deg":320}, "clouds":{"all":40}, "dt":1552243889, "sys": {"type":1,"id":6900,"message":0.0036}, "country":"PT", "sunrise":1552200793, "sunset":1552242906}, "id":8010448, "name":"Braga", "cod":200}
```

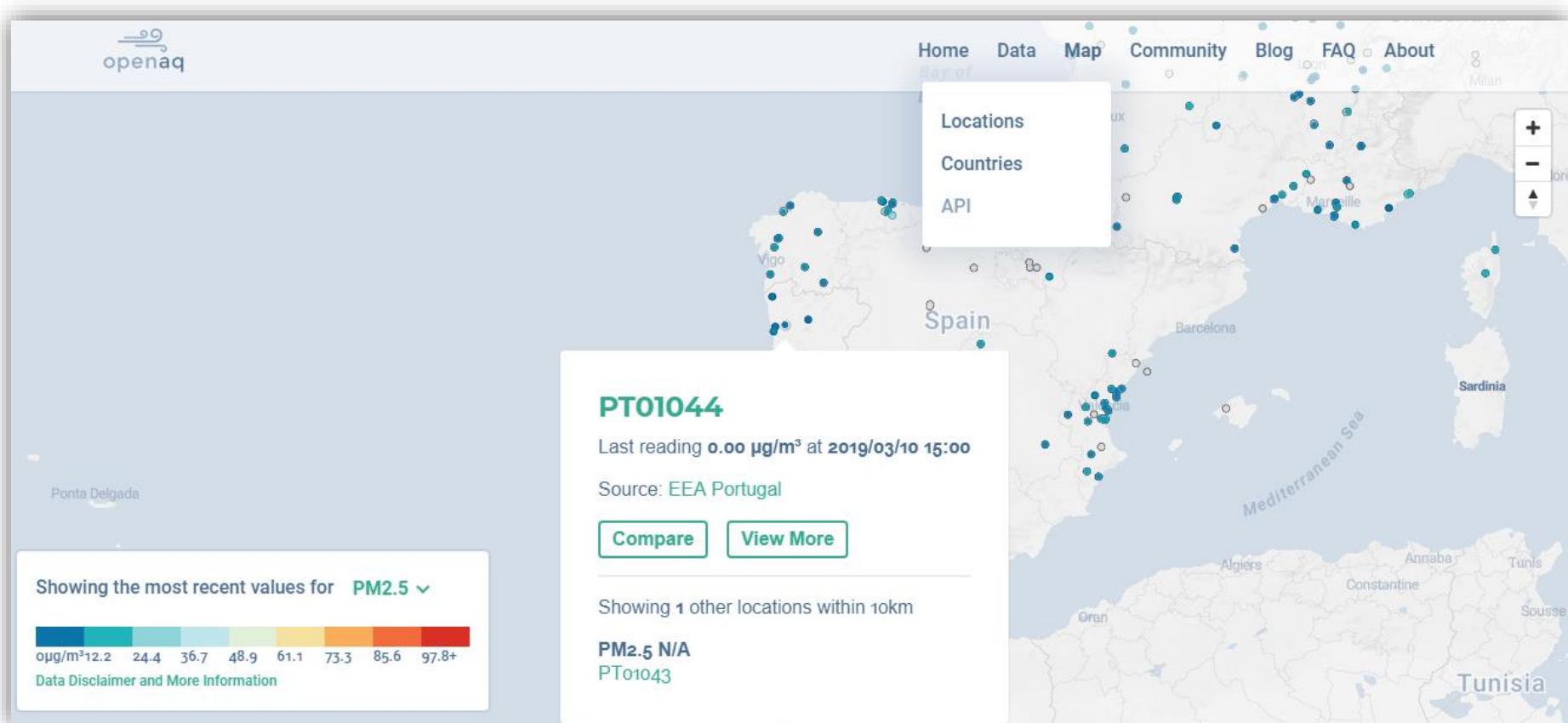
Open Air Quality

45

Concepts

SENSORIZATION

Hands On



Open Air Quality

46

Concepts

SENSORIZATION

Hands On

The screenshot shows the OpenAQ website interface. At the top, there's a navigation bar with links for Home, Data, Map, Community, Blog, FAQ, and About. Below the navigation is a map of Europe with several data points plotted. A sidebar on the left shows a color scale for PM2.5 levels from 0 to 85.5, with specific values for Braga, Portugal: 74.4, 36.7, 61.1, 75.5, and 85.5. The main content area displays the JSON API response for the city of Braga:

```
{"meta": {"name": "openaq-api", "license": "CC BY 4.0", "website": "https://docs.openaq.org/", "page": 1, "limit": 100, "found": 3}, "results": [{"location": "PT01041", "city": "Braga", "country": "PT", "distance": 1397113.1649692839, "measurements": [{"parameter": "no2", "value": 13.7, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}, {"parameter": "pm10", "value": 0, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}], "coordinates": {"latitude": 41.54972, "longitude": -8.405833}}, {"location": "PT01042", "city": "Braga", "country": "PT", "distance": 1396491.5102984651, "measurements": [{"parameter": "pm10", "value": 10, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}, {"parameter": "no2", "value": 0, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}, {"parameter": "o3", "value": 29, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}], "coordinates": {"latitude": 41.569443, "longitude": -8.456944}}, {"location": "PT01046", "city": "Braga", "country": "PT", "distance": 1404548.6633526601, "measurements": [{"parameter": "no2", "value": 11.6, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}, {"parameter": "pm10", "value": 22, "lastUpdated": "2019-03-10T15:00:00.000Z", "unit": "µg/m³", "sourceName": "EEA Portugal", "averagingPeriod": {"value": 1, "unit": "hours"}}], "coordinates": {"latitude": 41.449722, "longitude": -8.296389}}]
```

Open UV

47

Concepts

SENSORIZATION

Hands On

OpenUV Dev Console Sign In

OpenUV Playground

- Real-time UV Index
- UV Index Forecast
- Protection Time

Run in Postman Change Log

OpenUV API Docs

Authorisation

To authorise your client just add your API Key `YOUR_API_KEY` to `"x-access-token"` header for each request.

 Get My API Key * By signing in you agree with [Terms of Service](#)

Real-time UV Index

Get real-time UV Index by location. Optional altitude, ozone level and datetime could be provided as query parameters.

`GET https://api.openuv.io/api/v1/uv`

Request

Curl JavaScript Node.js Ruby Java Swift

```
curl -X GET \
'https://api.openuv.io/api/v1/uv?lat=-31.45&lng=115.67&dt=2018-01-24T10:56:00Z' \
-H 'x-access-token: YOUR_API_KEY'
```

JSON Response

[SIGN IN FOR REAL DATA](#)

```
{
  - result: {
    uv: 5.9,
    uv_time: "2018-02-22T11:20:51.810Z",
    uv_max: 9.51,
    uv_max_time: "2018-02-22T04:31:33.481Z",
    ozone: 267.11,
    ozone_time: "2018-02-22T09:06:10.889Z",
    - safe_exposure_time: {
      st1: 154,
      st2: 185,
      st3: 247,
      st4: 309,
      st5: 494,
      st6: 926
    },
    - sun_info: {
      - sun_times: {
        solarNoon: "2018-02-24T04:31:17.645Z",
        nadir: "2018-02-23T16:31:17.645Z",
        sunrise: "2018-02-23T06:45:17.645Z",
        sunset: "2018-02-23T17:45:17.645Z"
      }
    }
  }
}
```

Hands On

48

Concepts

Sensorization

HANDS ON

The screenshot shows a software application window titled "Commsensing | Arduino 1.8.2". The main area displays sensor data from a COM port, with a large watermark "HANDSON" overlaid across the center. The data shows multiple "Probe request" entries, each with a timestamp and a "Sant" value. Below this is a terminal window showing serial communication logs.

```
//Setup times
on_start_timeWithTime, startCallback, 1000L
on_start_timeWithTime, endTimes, 1000L

// Register event handlers. Callback functions will be called as long as the
// Call "onStationConnected" each time a station connects
stationConnectedHandler = WiFi.onStationConnected(onStationConnected)
// Call "onStationDisconnected" each time a station disconnects
stationDisconnectedHandler = WiFi.onStationDisconnected(onStationDisconnected)
// Call "onProbeRequestCaptureData" and "onProbeRequestFrom" each time a probe
probeRequestHandler = WiFi.onDataFromProbeRequest(onProbeRequestHandler)
probeRequestCaptureDataHandler = WiFi.onDataFromProbeRequest(onProbeRequestCaptureDataHandler)

Serial.println("All setup has been made! ");
if (isConnected) {
    Serial.print("Times is enabled and publish of data will be done every ");
    Serial.print(endTimes);
    Serial.println(" milliseconds ");
    startTimes();
} else {
    Serial.println("Times is DISABLED! ");
}

//Commsensing example
Serial.println("Starting 10000 bytes streaming at 1000000000");
Streaming 100000 bytes from C:\Users\bruno\AppData\Local\Temp\runfile_0010_040204\Commsensing.ino.ino to disk at 1000000000
..... [ 0% ]
..... [ 25% ]
..... [ 50% ]
..... [ 75% ]
..... [ 100% ]
```

Serial output:

```
Serial.println("Starting 10000 bytes streaming at 1000000000");
Streaming 100000 bytes from C:\Users\bruno\AppData\Local\Temp\runfile_0010_040204\Commsensing.ino.ino to disk at 1000000000
..... [ 0% ]
..... [ 25% ]
..... [ 50% ]
..... [ 75% ]
..... [ 100% ]
```