

IoT-Leb Hackathon

- Curious to discover the emerging LoRaWAN technology and work on the award-winning script IoT platform?
- Ready to build your first end-to-end IoT chain using cutting-edge technologies?
- Up to test your innovative ideas on a real-world IoT platform?

We invite you on March 12, 2019 to join the first IoT-Leb Hackathon along with Lebanon's most talented hackers, programmers, makers, and entrepreneurs. IoT-Leb Hackathon is part of IoT-Leb 19 conference that will take place at ESIB-USJ, CST Campus, Mar Roukos.

In the evolution towards the Internet of Things, an increasing number of devices equipped with sensors and communication interfaces will interact and collect data, communicate over the internet, and provide valuable monitoring and automation services. IoT-Leb Hackathon is a unique opportunity to get acquainted with the technologies that are set to play an essential role in the IoT landscape. During the Hackathon, you will get exclusive access to the latest IoT technologies.

-. Platform

During this lab, you will benefit from the first experimental platform implementing an end-to-end LoRaWAN solution in Lebanon. The platform consists of the following elements:

- Devices that communicate to one or more gateways via a wireless interface using single hop LoRa and implementing the LoRaWAN protocol. These devices are physically connected to sensors that generate data.
- Gateways or base stations that forward frames between the devices and the network server. Gateways are connected to the network server via IP interfaces.
- A LoRaWAN backend that implements the network server functions and provides frame control and security.
- Applications that enable to visualize and store the sensor data obtained from the devices.

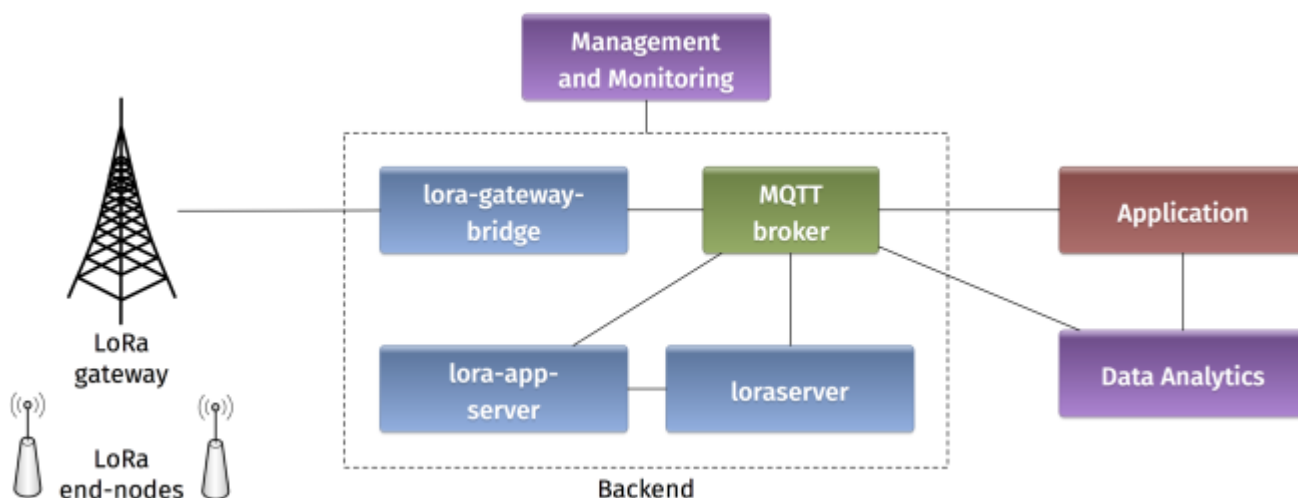


Figure 1. Architecture of the LoRaWAN Platform



Figure 2. Outdoor antenna at ESIB-USJ

- . Backend

In a LoRaWAN network, the devices communicate with a Network Server through the gateway. The backend installed in the platform is based on an open-source LoRaWAN network-server <https://www.loraserver.io>. A web interface is available for configuring the applications and devices on the platform (<https://212.98.XX.XX:8080>).

Start by choosing the application named hackathon to create a new device.

You should provide the following information for creating your device:

- A unique device name: IoTX (where X is your group number)
- The device description
- A unique device EUI on 64 bits: Random identifiers can be generated on <https://www.random.org/bytes/>
- A unique application key on 128 bits also obtained by random generation.

Make sure to choose lora-profile as a Device-profile in order to enable OTAA join method.

- . Devices

Devices in the LoRaWAN platform are implemented on <https://bit.ly/2HvgAa9> with ESP32 system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth, and an additional LoRa transceiver. The combined module as well as the basic configuration steps are presented in [Exploring LoRa lab](#).

Start by verifying the installation on your PC of the latest Arduino IDE and the presence of the LoRaWAN

LMIC library

. Then, download and open the example sketch

example-code-idoe-iot-lab.ino

with Arduino IDE.

Now you should configure your device with the same identifiers DEVEUI and APPKEY as in the backend:

```
// Change this!
// This should also be in little endian format, see below.
static const u1_t PROGMEM DEVEUI[8]= { };
void os_getDevEui (u1_t* buf) { memcpy_P(buf, DEVEUI, 8);}

// Change this!
static const u1_t PROGMEM APPKEY[16] = { };
void os_getDevKey (u1_t* buf) { memcpy_P(buf, APPKEY, 16);}
```



Note that the device EUI should be in little endian format. Hence, the key, 0badde1cafe2deca should be written as 0xCA, 0xDE, 0xE2, 0xAF, 0x1C, 0xDE, 0xAD, 0x0B in the Arduino sketch.

Let us analyze the radio parameters in the sketch by answering the following questions.



- In the setup function, which channels are activated on the device?
- What are the different spreading factors on each channel?
- What is the regulation on the radio channels in LoRa?

The LMIC library (see

documentation

) defines a set of events corresponding to the protocol machine state. These events appear in the onEvent() function.



- What is the difference between the JOINING and the JOINED events?
- When is the EV_TXCOMPLETE event called?

Finally let us look at the message sending on the device.



- What is the function for sending messages on the device? How it is called?
- What is the period of message sending? Explain the implementation choice.
- Is this period guaranteed according to the LoRaWAN specification?

Now you are ready to compile the sketch and upload it to the LoRaWAN device. Connect the device to a USB port on your PC, choose the board type as presented in [Exploring LoRa lab](#) and select the

corresponding port. Compile and upload!

Open the serial monitor in the Arduino IDE at 115200 baud and analyse the debug messages.

Getting back to the backend, you can monitor some important information related to your device such as the activation status, frame counters, and live data.

-. Applications

[mqtt-spy](#) is an open source utility intended to help you with monitoring activity on MQTT topics. It has been designed to deal with high volumes of messages, as well as occasional publications. [mqtt-spy](#) is a JavaFX application, so it should work on any operating system with an appropriate version of **Java 8 installed**. A very useful tutorial is available on <https://github.com/eclipse/paho.mqtt-spy/wiki>. The tool can be directly downloaded from this

[link](#)

You can use [mqtt-spy](#) to debug the messages received from the LoRaWAN devices. After starting the application, configure a new connection to the MQTT broker by simply adding the IP address of the broker in the Server URI field. Now you can subscribe to any MQTT topic. If you want to receive all messages arriving at the backend, you can use the generic topic #. You can also limit to the topic including the messages of any specific device:

application/APPLICATION_ID/node/DEVICE_EUI/rx.



- Summarize the concepts and functionalities of the MQTT protocol.
- What are the possible strengths and weaknesses in terms of security of MQTT?
- What are the different types of topics used by the backend? Explain.
- Explain the different fields in a captured MQTT message received from you device.



The payload received by the MQTT client is decrypted but encoded in Base64. You should decode it to get the original message (using for instance <https://www.base64encode.org>).

If you need to send data to your device, you should publish an encoded message in the corresponding topic `application/APPLICATION_ID/node/DEVICE_EUI/tx` as follows:

```
{
  "confirmed": false,           // whether the payload must
be sent as confirmed data down or not
  "fPort": 10,                 // FPort to use (must be > 0)
  "data": "...."              // base64 encoded data
(plaintext, will be encrypted by LoRa Server)
}
```



The payload sent by the MQTT client must be encoded in Base64.

You can also download a

[python script](#)
that automates the data retrieval from the MQTT broker .

- . LoRaWAN Challenges

Implement and provide technical documentation for each of the following challenges.

- . The End-to-End Challenge

I can send data from the device to the application.



Note that the `String` function can be used to cast the message you want to send in a string format.

- . The Downlink Challenge

I can send data from the application to the device.

- . The Radio Challenge

I can tune the LoRa radio parameters.

These two commands can be helpful when used after the join event:

```
LMIC_disableChannel(N);  
LMIC_setDrTxpow(DR_SF12, 14);
```

- . The Sensor Challenge

I can use different sensors to send data from the device: PIR, moisture, temperature, light, etc.

From:

<http://wiki.lahoud.fr/> - **wikiroute**

Permanent link:

http://wiki.lahoud.fr/doku.php?id=iot_leb_hackathon&rev=1551196181

Last update: **2019/02/26 16:49**

